LAPORAN PENELITIAN

ANALISIS ASPEK-ASPEK KUALITAS
SCHEMA DATABASE
(Studi Kasus Pada Database Akademik ISTA Yogyakarta)

Dibayai Dari Dana Lembaga Penelitian ISTA
Tahun Anggaran 2006/2007
Berdasarkan Surat Perjanjian Pelaksanaan Penelitian
No. 19/SPPP/Lemlit/II/2007

Oleh:
Suwanto Raharjo, S.Si., M.Kom.
Edhy Sutanta, S.T., M.Kom.

INSTITUT SAINS & TEKNOLOGI AKPRIND
YOGYAKARTA
2007
HALAMAN PENGESAHAN

1. Judul Penelitian:
   ANALISIS ASPEK-ASPEK KUALITAS SCHEMA DATABASE
   (Studi Kasus Pada Database Akademik ISTA Yogyakarta)

2. Bidang Penelitian:
   Teknologi

3. Ketua Peneliti:
   a. Nama Lengkap: Suwanto Raharjo, S.Si., M.Kom
   b. Jenis Kelamin: Laki-laki
   c. NIK: 03.0775.579.E
   d. Disiplin Ilmu: Teknik Informatika
   e. Jabatan: Asisten Ahli - 150
   f. Fakultas/Jurusan: Teknologi Industri/Teknik Informatika
   g. Alamat:
      Jl. Kalsiatalk No. 28 Komplek Balapan
      Yogyakarta
   h. Telepon/Faks/Email: (0274) 563029 ext.132 / (0274) 563847
   i. Alamat Rumah: Mulungan Wetan 7/17, Matali Sleman,
      Yogyakarta, 55285
   j. Telepon/Faks/Email: 081 5656-4133 / -- / wa@n@staff.aksindy.ac.id

4. Jumlah Anggota Peneliti:
   1) Nama Anggota I: Edhy Sutanta, S.T., M.Kom.
   2) Nama Anggota II: --

5. Lokasi Penelitian:
   Institut Sains & Teknologi AKPRIND
   Jl. Kalsiatalk No. 28 Yogyakarta
   --
   9 bulan
   06 Februari 2007
   06 Juli 2007
   Rp. 3.500.000,-
   (Tiga Juta Lima Ratus Ribu Rupiah)

Mengetahui:
Ketua Jurusan Teknik Informatika

Edhy Sutanta, S.T., M.Kom.
NIK. 96.0222.515.E

Yogyakarta, 05 Juli 2007
Ketua Peneliti,

Menyetujui,
Ketua Jabatan Penelitian

[Signature]
NIK. 03.0775.579.E

[Signature]
NIK. 96.0222.515.E
SURAT KETERANGAN KARYA ILMIAH

Dengan ini kami:
Nama lengkap: Ir. Amir Hamzah, M.T.
NIP: 87.0563.319.E
Jabatan: Lektor Kepala
Bidang Ilmu: Teknik Informatika
Lembaga: Jurusan Teknik Informatika, FTI, ISTA Yogyakarta

Memberikan rekomendasi untuk karya ilmiah dengan judul:
ANALISIS ASPEK-ASPEK KUALITAS SCHEMA DATABASE
(Studi Kasus Pada Database Akademik ISTA Yogyakarta)

Atas nama Saudara tersebut di bawah ini:

b. NIK: 03.0775.579.E
c. Jabatan: Asisten Ahli - AK 150
d. Bidang Ilmu: Teknik Informatika
e. Lembaga: Jurusan Teknik Informatika,
FTI, ISTA Yogyakarta

Isi rekomendasi karya ilmiah itu sebagai berikut:

a. Mutu: Amat Baik / Baik / Cukup / Kurang
b. Sofistifikasi: Amat Baik / Baik / Cukup / Kurang
c. Kemutakhiran: Amat Baik / Baik / Cukup / Kurang

Yogyakarta, 05 Juli 2007
Yang memberi rekomendasi

Ir. Amir Hamzah, M.T.
SURAT KETERANGAN KARYA ILMIAH

Dengan ini kami:
Nama lengkap : Ir. Amir Hamzah, M.T.
NIP : 87.0563.319.E
Jabatan : Lektor Kepala
Bidang Ilmu : Teknik Informatika
Lembaga : Jurusan Teknik Informatika, FTI, ISTA Yogyakarta

Memberikan rekomendasi untuk karya ilmiah dengan judul:
ANALISIS ASPEK-ASPEK KUALITAS SCHEMA DATABASE
(Studi Kasus Pada Database Akademik ISTA Yogyakarta)

Atas nama Saudara tersebut di bawah ini:

b. NIK / NIDN : 96.0372.515.E / 0508037201
c. Jabatan : Lektor Kepala- AK. 550
d. Bidang Ilmu : Teknik Informatika
e. Lembaga : Jurusan Teknik Informatika,
               FTI, ISTA Yogyakarta

Isi rekomendasi karya ilmiah itu sebagai berikut:

a. Mutu : Amat Baik / Baik / Cukup / Kurang
b. Sofistifikasi : Amat Baik / Baik / Cukup / Kurang
c. Kemutakhiran : Amat Baik / Baik / Cukup / Kurang

Yogyakarta, 05 Juli 2007
Yang memberi rekomendasi

Ir. Amir Hamzah, M.T.
SURAT KETERANGAN KARYA ILMIAH

Dengan ini kami:
Nama lengkap : Muhammad Sholeh, S.T., M.T.
NIK : 94.1269.498.E
Jabatan : Lektor Kepala
Bidang Ilmu : Teknik Informatika
Lembaga : Jurusan Teknik Informatika, FTI, ISTA Yogyakarta

Memberikan rekomendasi untuk karya ilmiah dengan judul:
ANALISIS ASPEK-ASPEK KUALITAS SCHEMA DATABASE
(Studi Kasus Pada Database Akademik ISTA Yogyakarta)

Atas nama Saudara tersebut di bawah ini:
   b. NIK : 03.0775.579.E
   c. Jabatan : Asisten Ahli - AK 150
   d. Bidang Ilmu : Teknik Informatika
   e. Lembaga : Jurusan Teknik Informatika,
   FTI, ISTA Yogyakarta

Isi rekomendasi karya ilmiah itu sebagai berikut:
   a. Mutu : Amat Baik / Baik / Cukup / Kurang
   b. Sofistifikasi : Amat Baik / Baik / Cukup / Kurang
   c. Kemutakhiran : Amat Baik / Baik / Cukup / Kurang

Yogyakarta, 05 Juli 2007
Yang memberi rekomendasi

Muhammad Sholeh, S.T., M.T.
SURAT KETERANGAN KARYA ILMIAH

Dengan ini kami :
Nama lengkap : Muhammad Sholeh, S.T., M.T.
NIK : 94.1269.498.E
Jabatan : Lektor Kepala
Bidang ilmu : Teknik Informatika
Lembaga : Jurusan Teknik Informatika, FTI, ISTA Yogyakarta

Memberikan rekomendasi untuk karya ilmiah dengan judul :
ANALISIS ASPEK-ASPEK KUALITAS SCHEMA DATABASE
(Studi Kasus Pada Database Akademik ISTA Yogyakarta)

Atas nama Saudara tersebut di bawah ini :
  b. NIK / NICN : 96.0372.515.E / 0508037201
  c. Jabatan : Lektor Kepala- AK. 550
  d. Bidang ilmu : Teknik Informatika
  e. Lembaga : Jurusan Teknik Informatika,
     FTI, ISTA Yogyakarta

Isi rekomendasi karya ilmiah itu sebagai berikut :
  a. Mutu : Amat Baik / Baik / Cukup / Kurang
  b. Sofistifisasi : Amat Baik / Baik / Cukup / Kurang
  c. Kemutakhiran : Amat Baik / Baik / Cukup / Kurang

Yogyakarta, 05 Juli 2007
Yang memberi rekomendasi

Muhammad Sholeh, S.T., M.T.
BAGIAN A.
LAPORAN HASIL PENELITIAN
RINGKASAN DAN SUMMARY

Kenyataan di lapangan selama ini menunjukkan bahwa tidak semua organisasi berhasil mengimplementasikan SIM dengan baik. Salah satu hasil audit pada implementasi SIM di Indonesia menunjukkan bahwa yang paling sering dijumpai adalah terjadinya fenomena tambal sulam aplikasi SIM, termasuk di dalamnya adalah kegagalan implementasi dan fenomena “tambal sulam” pada SIAKAD. Hal ini terbukti dengan masih banyaknya PT yang telah melakukan pengembangan dan implementasi SIAKAD, namun hasilnya belum memuaskan hingga saat ini. Salah satu peran database, yaitu sebagai sumber informasi bagi SIM, mengimplikasikan bahwa database yang digunakan harus mampu memenuhi kebutuhan berbagai informasi dan data bagi para penggunanya, baik untuk saat ini maupun mendatang. Rancangan schema database seharusnya dirancang sedemikian rupa sehingga mempunyai kualitas yang tinggi. Terkait dengan hal tersebut, terdapat banyak aspek yang mempengaruhi keberhasilan implementasi SIM dalam organisasi, diantaranya adalah aspek organisasi, manajemen, perilaku, atau teknologi yang meliputi perangkat keras, perangkat lunak, rekayasa sistem, database.

Dengan mengacu pada hasil penelitian Olaf Herden (2001), penelitian ini melakukan analisis tentang aspek-aspek kualitas schema database pada database akademik yang digunakan di ISTA. Analisis dilakukan pada rancangan logikal dengan menggunakan model logika berupa business rule yang digunakan pada ISTA. Aspek-aspek kualitas schema yang dianalisis meliputi 9 (sembilan) kriteria, yaitu correctness, consistency, relevance, scope, level of detail, completeness, minimality, ability of integration, serta readability. Hasil penelitian ini diharapkan dapat memberikan rekomendasi langkah yang mungkin dilakukan oleh pengelola/penanggungjawab database akademik di ISTA. Berdasarkan hasil analisis, secara umum dapat dinyatakan bahwa schema database akademik ISTA memiliki kualitas yang rendah. Kondisi ini akan menjadi potensi bagi timbulnya berbagai permasalahan yang serius di kelak kemudian hari. Dan oleh karenanya, maka perlu segera dilakukan upaya-upaya penyempurnaan untuk menghindari timbulnya permasalahan yang lebih kompleks dan semakin sulit ditangani.
KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Kuasa, karena atas rahmat dan ijin-Nya laporan penelitian ini dapat kami selesaikan.

Pada kesempatan ini kami menyampaikan ucapan terimakasih dan penghargaan kepada berbagai pihak yang telah mendukung kelancaran dan terlaksananya penelitian ini, yaitu:

1. Rektor ISTA Yogyakarta, yang telah memberikan dukungan dalam pelaksanaan penelitian ini
2. Dekan Fakultas Teknologi Industri, ISTA Yogyakarta, yang telah memberikan dukungan dalam pelaksanaan penelitian ini
3. Kepala Lembaga Penelitian ISTA Yogyakarta, yang telah memberikan dukungan dalam pelaksanaan penelitian ini
4. Bapak Ir. Amir Hamzah, M.T., yang telah memberikan rekomendasi, masukan dan saran terkait dengan penelitian ini
5. Bapak Muhammad Sholeh, S.T., M.T., yang telah memberikan rekomendasi, masukan dan saran terkait dengan penelitian ini
6. Rekan-rekan Dosen Jurusan Teknik Informatika, FTI, ISTA Yogyakarta
7. Semua pihak yang telah mendukung pelaksanaan penelitian ini

Kami menyadari bahwa hasil penelitian ini masih mengandung kedangkalan dan kekurangan, oleh karena itu umpan balik, saran dan masukan dari para Pemerhati dan Pembaca sangat kami harapkan untuk melakukan peningkatan kualitas pada masa selanjutnya.

Akhirnya, kami berharap agar hasil penelitian ini dapat bermanfaat dan mencapai sasaran yang diharapkan.

Yogyakarta, 05 Juli 2007

Peneliti
DAFTAR ISI

Halaman:
HALAMAN JUDUL ........................................................................ i
HALAMAN PENGESAHAN ........................................................... ii
SURAT KETERANGAN KARYA ILMIAH ........................................ iii

BAGIAN A. LAPORAN HASIL PENELITIAN ........................................ viii
RINGKASAN DAN SUMMARY ..................................................... viii
KATA PENGANTAR........................................................................ ix
DAFTAR ISI .................................................................................. x

BAB I PENDAHULUAN ............................................................. 1
1.1. Latar Belakang Masalah ...................................................... 1
1.2. Rumusan Masalah ............................................................. 4
1.3. Batasan Masalah ............................................................... 4

BAB II LANDASAN TEORI .......................................................... 5
2.1. Tinjauan Tentang Konsep SIM ............................................. 5
  2.1.1. Definisi SIM .......................................................... 5
  2.1.2. Tujuan SIM ........................................................... 8
  2.1.3. Karakteristik Kebutuhan Informasi dalam SIM ................. 9
  2.1.4. Metodologi Pengembangan SIM .................................... 10
  2.1.5. Perancangan Sistem Dalam SIM .................................. 18
2.2. Tinjauan Tentang Konsep Database ..................................... 20
  2.2.1. Definisi Database ................................................. 20
  2.2.2. Tujuan dan Keuntungan Database ............................. 21
  2.2.3. Kekangan Dalam Database ...................................... 22
  2.2.4. Pandangan Terhadap Database ................................ 23
  2.2.5. Relational Database Model/RDBM.............................. 24
2.3. Database dan SIM ............................................................ 31
2.4. Perancangan Database Untuk SIM .................................... 31
2.5. Verifikasi Struktur Database ............................................. 40
2.6. Schema Database Universal ............................................. 42
2.7. Pendekatan Dalam Pengukuran Kualitas Informasi ..................... 44

BAB III TUJUAN DAN MANFAAT PENELITIAN ......................... 46
3.1. Tujuan Penelitian ........................................................... 46
3.2. Manfaat Penelitian .......................................................... 46

BAB IV METODOLOGI PENELITIAN ........................................ 47
4.1. Metode Penelitian........................................................... 47
4.2. Lingkup Permasalahan...................................................... 49
4.3. Aspek Permasalahan ......................................................... 49
4.4. Jadwal Waktu Penelitian................................................... 50
BAB V HASIL DAN PEMBAHASAN ................................................................. 51
  5.1. Hasil ........................................................................................................ 51
      5.1.1. Sistem Informasi di Perguruan Tinggi (SIPT) ......................... 51
      5.1.2. Sub Sistem Sistem Informasi Akademik (SIAKAD) dalam SIPT ................................................................. 68
      5.1.3. Implementasi Sistem Informasi di ISTA ............................... 81
      5.1.4. Implementasi SIAKAD di ISTA ........................................ 99
  5.2. Pembahasan .......................................................................................... 100
      5.2.1. Instalasi Database Server PostgreSQL ................................. 100
      5.2.2. Pembuatan Duplikasi Database Akademik ............................. 100
      5.2.3. Membuat Database Duplikasi ........................................... 101
      5.2.4. Analisis/Pengujian Aspek-Aspek Kualitas Schema Database ............................................................................... 102
      5.2.5. Rekapitulasi Hasil Analisis/Pengujian Aspek-Aspek Kualitas Schema Database ......................................................... 112

BAB VI KESIMPULAN DAN SARAN ............................................................ 115
  6.1. Kesimpulan .......................................................................................... 115
  6.2. Saran ..................................................................................................... 117

DAFTAR PUSTAKA ......................................................................................... 118
LAMPIRAN: SCHEMA DATABASE AKADEMIK ISTA .......................... 122

BAGIAN B. DRAFT ARTIKEL ILMIAH ......................................................... 227

BAGIAN C. SINOPSIS PENELITIAN LANJUTAN ................................. 242
BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan teknologi informasi-komputer saat ini sudah mencapai pada tahap di mana ukurannya semakin kecil, kecepatannya semakin tinggi, namun harganya semakin murah dibandingkan dengan kemampuan kerjanya. Kondisi ini mendorong masyarakat berlomba-lomba memanfaatkan komputer sebagai alat bantu pengolahan data dengan cara membangun sistem pengolahan data terkomputerisasi untuk penyajian informasi, baik untuk keperluan pribadi maupun organisasinya. Sekalipun kurang tepat, sistem pengolahan data terkomputerisasi untuk penyajian informasi dalam suatu organisasi sering disebut sebagai Sistem Informasi Manajemen (SIM).


Kenyataan di lapangan menunjukkan, bahwa tidak semua organisasi berhasil mengimplementasikan SIM dengan baik. Eko Nugroho (1991) telah meneliti pengaruh struktur organisasi dan sumber daya manusia terhadap keberhasilan implementasi SIM dalam organisasi dalam penelitian tesisnya. Hasil penelitian tersebut membuktikan, bahwa rata-rata persentase tingkat keberhasilan implementasi SIM, khususnya di DIY relatif rendah, hanya sekitar 60%. Penelitian tersebut juga telah membuktikan bahwa sumberdaya manusia (para teknisi sistem informasi, misalnya operator, analis sistem komputer, manajer unit pemrosesan

Organisasi yang mengimplementasikan SIM dapat bergerak dalam bidang apa saja, baik yang bergerak dalam bidang produksi barang maupun jasa, profit maupun non-profit, berskala kecil, menengah, maupun besar. Salah satu contoh organisasi yang telah mengimplementasikan SIM adalah Perguruan Tinggi (PT). Implementasi SIM dalam lingkungan PT utamanya digunakan untuk pengolahan data akademik yang sering dikenal dengan sebutan Sistem Informasi Akademik (SIAKAD). Kegagalan implementasi SIM dan fenomena “tambal sulam” aplikasi SIM ternyata juga dapat terjadi dalam SIAKAD. Hal ini dapat dibuktikan dengan masih adanya PT yang telah melakukan pengembangan dan implementasi SIAKAD lebih dari satu dekade lamanya, namun hasilnya belum memuaskan hingga saat ini.

Dalam suatu implementasi SIM, dapat dipahami bahwa kebutuhan informasi (dan data) bagi para pengguna untuk dapat dipastikan akan mengalami perubahan. Perubahan kebutuhan informasi (dan data) tersebut akan memerlukan penyesuaian pada komponen SIM lainnya,
termasuk database. Perubahan kebutuhan informasi (dan data) tersebut dapat terjadi, baik akibat perubahan yang terencana maupun yang tidak terencana sebelumnya. Perubahan kebutuhan informasi (dan data) akan memerlukan perubahan pada program aplikasi, nilai-nilai rinci data (data items) yang perlu disimpan dalam database atau bahkan perubahan pada struktur tabel database. Pada umumnya perubahan pada program aplikasi diperlukan karena adanya tuntutan kebutuhan tampilan keluaran yang lebih baik dan lebih menarik. Perubahan pada nilai-nilai rinci data (data items) dalam database dapat dilakukan dengan cara menambahkan atau menyesuaikan atribut dan/atau nilai-nilai rinci data (data items) dalam tabel database. Sedangkan perubahan rancangan struktur tabel database akan relatif mahal, lama, dan hanya dapat dilakukan dengan cara membangun kembali definisi struktur tabel database, kerelasan antar data dalam database, batasan-batasan nilai pada atribut dan/atau nilai-nilai rinci data, dan sekaligus perubahan program aplikasi.

Salah satu peran database, yaitu sebagai sumber informasi bagi SIM, mengimplikasikan bahwa database yang digunakan harus mampu memenuhi kebutuhan berbagai informasi (dan data) bagi para penggunanya, baik untuk saat ini maupun mendatang. Tanpa mengabaikan sumber aday informasi yang lainnya, aspek rekayasa sistem, khususnya rancangan schema database akan sangat mempengaruhi keberhasilan implementasi SIM dalam organisasi. Rancangan schema database yang berkualitas akan meningkatkan kualitas SIM, sebaliknya rancangan schema database yang kurang berkualitas akan berpotensi menimbulkan permasalahan-permasalahan dan bahkan keagilan implementasi SIM pada suatu organisasi, termasuk Perguruan Tinggi. Dengan demikian, schema database seharusnya dirancang sedemikian rupa sehingga mempunyai kualitas yang baik/tinggi.

Dengan menggunakan meta model, penelitian Olaf Herden (2001) telah berhasil mendefinisikan dan mengukur aspek-aspek kualitas schema berorientasi obyek (object oriented) suatu database, dan mengusulkan
sebuah proses untuk melakukan tinjauan dan pengukuran aspek kualitas schema suatu database. Aspek-aspek yang relevan untuk pengukuran kualitas schema suatu database yang dimaksud meliputi 9 (sembilan) kriteria, yaitu kebenaran (correctness), konsistensi (consistency), relevansi (relevance), jangkauan (scope), tingkat detail (level of detail), kelengkapan (completeness), minimalitas (minimality), kemampuan untuk diintegrasikan (ability of integration), serta kemampuan untuk dibaca (readability) (Olaf Herden, 2001, http://www.iro.umontreal.ca/~sahraouh/qaoose01/Herden.PDF).

1.2. Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, dapat dipahami bahwa kualitas database merupakan permasalahan penting. Oleh karena itu, maka rumusan permasalahan dalam penelitian ini adalah sebagai berikut:

1. Bagaimanakah melakukan analisis kualitas schema database
2. Bagaimanakah kualitas schema database yang digunakan pada database akademik di ISTA

1.3. Batasan Masalah

Analisis dalam penelitian ini difokuskan pada aspek kualitas schema database dengan batasan sebagai berikut:

1. Analisis dilakukan pada rancangan schema database yang digunakan pada database akademik yang digunakan di ISTA
2. Analisis dilakukan pada rancangan logikal, bukan implementasi pada DBMS, teknologi perangkat keras, maupun program aplikasi yang digunakan
3. Analisis dilakukan dengan menggunakan model logika berupa business rule yang digunakan pada ISTA
BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Tentang Konsep SIM

2.1.1. Definisi SIM

Dalam lingkungan pengolahan data menggunakan komputer, sistem pengolahan data terkomputerisasi untuk penyajian informasi bagi keperluan manajemen dalam organisasi sering disebut sebagai “Sistem Informasi Manajemen/SIM” (Management Informations Systems/MIS), walaupun tidak semua orang menyetujui penggunaan istilah SIM tersebut. Istilah SIM telah didefinisikan oleh para pakar dengan cara yang berbeda-beda. Sebagian orang menyebutnya sebagai “Sistem Informasi” saja, dan sebagian yang lain menyebutnya sebagai “Pengolahan Data Elektronik/PDE” (Electronic Data Processing/EDP). Namun demikian menurut Eko Nugroho (1991), pada umumnya SIM merupakan istilah yang paling banyak dipergunakan.

Gordon B. Davis (1987) mendefinisikan SIM sebagai integrasi sistem manusia-mesin yang menyediakan informasi untuk mendukung fungsi-fungsi operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi. SIM menggunakan hardware dan software komputer; prosedur manual; model untuk analisis, perencanaan, kendali dan pembuatan keputusan; database.


Menurut Efraim Turban dan Jay E. Aronson (2001), SIM mengumpulkan, memproses, menyimpan, menganalisis, dan
menyebarkan informasi untuk kebutuhan khusus. SIM menerima input dan memproses data untuk menyediakan informasi untuk para pengambil keputusan dan membantu para pengambil keputusan dalam mengkomunikasikan hasil yang diperolehnya. Pengambil keputusan dapat bersifat individual atau kelompok individu.


SIM menyediakan informasi untuk suatu area fungsional dalam organisasi. Menurut Raymmond McLeod dan George Shell (2001), pengguna informasi dapat meliputi para manajer, para personal non manajer, serta para personal dan organisasi dalam lingkungan perusahaan. Manajer memiliki tugas mengelola seluruh sumber daya yang ada dengan cara yang paling efektif, yaitu sumber daya fisik yang terdiri atas personal, material, mesin (termasuk fasilitas dan energi), dan uang; sumber daya konseptual berupa informasi (dan data) (McLeod dan Schell,
Para manajer membutuhkan informasi (dan data) untuk melaksanakan tugas-tugasnya. Informasi (dan data) telah menjadi bagian sumber daya penting dan strategis, sehingga perlu dikelola dengan baik sebagaimana sumber daya fisik lainnya. Perhatian khusus pada pengelolaan informasi diperlukan karena adanya dua pengaruh, yaitu kompleksitas kegiatan bisnis yang semakin meningkat dan kemampuan komputer yang semakin meningkat (McLeod dan Schell, 2001).


Secara fungsional, SIM dikembangkan untuk memenuhi kebutuhan-kebutuhan informasi dengan menekankan pada area fungsional penggunanya. Penyediaan informasi ini ditangani oleh software penyusun laporan (report writing software) yang mampu mencetak laporan-laporan periodik, khusus, dan perbedaan. Laporan periodik dibuat secara
berkala dan terjadwal, laporan khusus dibuat untuk kepentingan tertentu, sedangkan laporan perbedaan dibuat apabila terjadi sesuatu penyimpangan (McLeod dan Schell, 2001).

Cara lain memahami definisi SIM adalah dengan memahami dan menggabungkan 3 (tiga) istilah penyusun SIM, yaitu sistem, informasi, dan manajemen. Dengan menggabungkan ketiga istilah tersebut, maka SIM dapat dipahami sebagai sekumpulan subsistem yang saling berhubungan; berkumpul bersama-sama dan membentuk satu kesatuan; saling berinteraksi dan bekerjasama antara bagian satu dengan yang lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (input) berupa data, mengolahnya (processing), dan menghasilkan keluaran (output) berupa informasi sebagai dasar pengambilan keputusan yang berguna dan mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu juga maupun di masa mendatang; mendukung kegiatan operasional, manajerial, dan strategis; dengan memanfaatkan sumber daya yang ada guna mencapai tujuan.

2.1.2. Tujuan SIM
Suatu SIM dikembangkan untuk tujuan-tujuan tertentu sesuai dengan kebutuhan organisasi. SIM yang sederhana dikembangkan dengan tujuan untuk memenuhi kebutuhan informasi (dan data) unit-unit fungsional organisasi. SIM seperti ini mendukung pengolahan transaksi pada tingkat operasional dan sedikit dukungan pada tingkat manajerial. SIM yang lebih kompleks menangani pengolahan data transaksi pada tingkat operasional dan penekanan pada tingkat manajerial. SIM seperti ini terdiri atas modul-modul yang masing-masing menangani pengolahan data pada unit fungsional tertentu. Sedangkan SIM yang kompleks dikembangkan untuk menangani kebutuhan informasi (dan data) untuk semua pengguna pada semua tingkat kegiatan manajemen, pada semua unit fungsional yang ada. Oleh karena itu, maka setiap SIM yang dikembangkan oleh organisasi dapat mempunyai tujuan yang spesifik.
Menurut Charles S. Parker (1989), SIM secara umum mempunyai tujuan untuk operational efficiency, functional effectiveness, better service, product creation and improvement, altering the basis of competition, spotting and taking advantage of opportunities, serta client lock in-competitor lock out.

2.1.3. Karakteristik Kebutuhan Informasi dalam SIM

Pada hakekatnya, kebutuhan informasi pada setiap setiap tingkat manajemen adalah berbeda-beda, yaitu tergantung pada fungsi operasional, kegiatan manajemen, dan pembuatan keputusan (McLeod dan Schell, 2001). Secara ringkas, kebutuhan informasi menurut kategori keputusan ditunjukkan oleh Tabel 2.1, sedangkan Tabel 2.2 menunjukkan perbedaan jenis keputusan pada tiga tingkat kegiatan manajemen (Davis, 1987).

**Tabel 2.1: Kebutuhan informasi menurut kategori keputusan**

<table>
<thead>
<tr>
<th>Ciri Informasi</th>
<th>Perencanaan &amp; pengendalian operasional</th>
<th>Perencanaan taktis &amp; pengendalian manajemen</th>
<th>Perencanaan strategis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sumber</td>
<td>Sebagian besar internal → Eksternal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Luasnya</td>
<td>Sangat jelas, sempit → Sangat luas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tingkat agregasi</td>
<td>Sangat terinci → Agregat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horison waktu</td>
<td>Historis → Masa depan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keadaan waktu</td>
<td>Sangat baru → Agak lama</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kecermatan</td>
<td>Tinggi → Rendah</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frek. pemakaian</td>
<td>Sangat tinggi → Jarang</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sumber: Gordon B. Davis, 1987

**Tabel 2.2: Perbedaan jenis keputusan pada tiga tingkat kegiatan manajemen**

<table>
<thead>
<tr>
<th>Tingkat</th>
<th>Catatan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perencanaan strategis</td>
<td>Penetapan tujuan, definisi sasaran, kebijakan, pedoman umum, yang mengarahkan alur untuk organisasi, misal: bidang usaha, strategi pasar.</td>
</tr>
<tr>
<td>Perencanaan taktis &amp; pengendalian manajemen</td>
<td>Perolehan sumber daya, taktik perolehan, lokasi, produk baru, pemakaian anggaran, laporan perbedaan (variance)</td>
</tr>
<tr>
<td>Perencanaan operasional</td>
<td>Pendayagunaan fasilitas &amp; sumber daya yang ada untuk menyelenggarakan kegiatan</td>
</tr>
</tbody>
</table>

Sumber: Gordon B. Davis, 1987

Gambar 2.1: Karakteristik kebutuhan informasi

<table>
<thead>
<tr>
<th>Sumber informasi</th>
<th>Manajer Tertinggi</th>
<th>Manajer Menengah</th>
<th>Manajer Terendah</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frekuensi penggunaan</td>
<td>Eksternal</td>
<td>Eksternal &amp; Eksternal</td>
<td>Internal</td>
</tr>
<tr>
<td>Skala waktu</td>
<td>Jarang</td>
<td>Kadang-kadang</td>
<td>Jarang</td>
</tr>
<tr>
<td>Horison waktu</td>
<td>Lama bertahun-tahun</td>
<td>Sedang minggu/bulan</td>
<td>Singkat hari/jam</td>
</tr>
<tr>
<td>Lingkup</td>
<td>Masa depan</td>
<td>Masa depan &amp; histori</td>
<td>Histori</td>
</tr>
<tr>
<td>Hakikat keputusan</td>
<td>Tidak terstruktur</td>
<td>Terstruktur</td>
<td>Sangat terstruktur</td>
</tr>
</tbody>
</table>

2.1.4. Metodologi Pengembangan SIM

Menurut J. L. Whitten dan L. D. Bentley (1998), metodologi merupakan proses yang digunakan untuk pengembangan sistem. Seluruh metodologi pengembangan SIM diarahkan dari sebuah sistem logik proses pemecahan masalah yang sering disebut siklus hidup pengembangan sistem (system development life cycle/SDLC), kadang-kadang juga disebut siklus hidup pengembangan aplikasi (application development life cycle). SDLC merupakan proses logik membangun SIM dan aplikasi-aplikasi komputer untuk memecahkan masalah yang dilakukan oleh systems analyst, software engineer, programmer, dan end-user. Sebuah metodologi adalah implementasi fisik pada siklus hidup logik yang menggabungkan langkah demi langkah aktivitas untuk setiap tahapan; perseorangan dan kelompok yang menjadi pelaku dalam setiap aktivitas; kemampuan menghasilkan dan standard kwalitas untuk setiap aktivitas; penggunaan alat dan teknik untuk setiap aktivitas (Whitten dan Bentley, 1998). Metodologi yang benar akan memberikan arah pada SDLC.

Selanjutnya, J. L. Whitten dan L. D. Bentley (1998) menyampaikan delapan prinsip pengembangan sistem, yaitu:
1. Prinsip 1: Libatkan owner dan user semaksimal mungkin
2. Prinsip 2: Gunakan pendekatan penyelesaian masalah
3. Prinsip 3: Susun tahapan dan aktivitas
4. Prinsip 4: Susun standar untuk konsistensi pengembangan dan dokumentasi
5. Prinsip 5: Tetapkan sistem sebagai investasi modal
6. Prinsip 6: Jangan takut membatalkan atau merevisi lingkup sistem
7. Prinsip 7: Sistem dipecah-pecah agar lebih mudah dikuasai
8. Prinsip 8: Sistem dirancang untuk dapat tumbuh dan berubah


1. Permasalahan kinerja (Performance=P), yaitu:
   a. *Throughput*, merupakan jumlah pekerjaan yang dapat diselesaikan dalam periode waktu tertentu
   b. *Response time*, merupakan rata-rata penundaan yang terjadi antara saat permintaan transaksi dan respon yang diberikan terhadap transaksi

2. Permasalahan informasi (*Information*=I) (dan data), yaitu:
   a. *Output*, meliputi:
      1) Kekurangan pada beberapa informasi
      2) Kekurangan pada informasi penting
      3) Kekurangan pada informasi yang relevan
      4) Tertlalu banyak informasi (*information overload*)
      5) Informasi dalam format yang tidak berguna
      6) Informasi tidak akurat
      7) Informasi yang sulit diproduksi
      8) Informasi tidak sesuai urutan waktu penggunaan
   b. Input, meliputi:
      1) Data tidak tertangkap
2) Data tidak ditangkap pada waktu yang tepat
3) Data ditangkap secara tidak akurat, memuat kesalahan
4) Data sulit ditangkap
5) Data ditangkap secara berulang
6) Terlalu banyak data yang ditangkap
7) Data tidak sah ikut ditangkap
c. Penyimpanan data, meliputi:
   1) Data disimpan berulang dalam beberapa file dan/atau database
   2) Data disimpan tidak akurat
   3) Data disimpan secara tidak aman
   4) Data tidak diorganisasi dengan baik
   5) Data tidak fleksibel, sulit menemukan kebutuhan informasi baru dari data yang tersimpan
   6) Data tidak dapat diakses

2. Permasalahan ekonomis (Economic=E), yaitu:
   a. Beaya-beaya, meliputi:
      1) Beaya tidak terduga
      2) Beaya di luar kemampuan sumber daya yang tersedia
      3) Beaya terlalu tinggi
   b. Keuntungan, meliputi:
      1) Pangsa pasar baru dapat dieksplorasi
      2) Pasar yang ada saat ini dapat ditingkatkan
      3) Order dapat ditingkatkan

3. Permasalahan kontrol (Control=C) dan keamanan, yaitu:
   a. Keamanan atau kontrol terlalu minim, yaitu:
      1) Data input tidak diedit cukup memadai
      2) Pelanggaran etika pada data atau informasi
      3) Data yang disimpan secara berulang tidak konsisten pada file-file atau database yang berbeda
      4) Kebijakan privacy data dilanggar
5) Kesalahan proses (baik karena manusia, mesin, atau software)
6) Kesalahan pengambilan keputusan
b. Keamanan atau kontrol terlalu banyak, yaitu:
   1) Birokrasi memperlambat sistem
   2) Kontrol membuat para pelanggan atau karyawan
   3) Kontrol yang ketat menyebabkan penundaan proses

4. Permasalahan efisiensi (Efficiency=E), yaitu:
   a. Pemborosan waktu pada orang, mesin, atau komputer, meliputi:
      1) Data dininputkan atau di-copy secara berulang
      2) Data diproses secara berulang
      3) Informasi di-generate secara berulang
   b. Pemborosan material dan suplai pada orang, mesin, atau komputer
   c. Terlalu banyak upaya untuk tugas-tugas
   d. Material terlalu banyak diperlukan untuk tugas-tugas

5. Permasalahan pelayanan (Service=S), yaitu:
   a. Sistem memberikan hasil yang tidak akurat
   b. Sistem memberikan hasil yang tidak konsisten
   c. Sistem memberikan hasil yang tidak reliabel (unreliable)
   d. Sistem sulit dipelajari
   e. Sistem sulit digunakan
   f. Sistem janggal digunakan
   g. Sistem tidak fleksibel untuk hal-hal atau situasi-situasi baru
   h. Sistem tidak fleksibel untuk perubahan
   i. Sistem tidak kompatibel dengan sistem lain
   j. Sistem tidak dikoordinasikan dengan sistem-sistem lain

2.1.4.1. System Development Life Cycle/SDLC
Metodologi pengembangan sistem menurut J. L. Whitten dan L. D. Bentley (1998), meliputi 8 (delapan) tahap, yaitu:
1. Tahap *survey*, tahap ini memiliki tiga kegunaan, yaitu:
   a. Menjawab pertanyaan apakah proyek memiliki nilai;
   b. Mendefinisikan lingkup proyek dan masalah, peluang, arah yang ingin dicapai;
   c. Menentukan tim dan peserta, anggaran, jadwal untuk proyek.

2. Tahap *study*, tahap ini memiliki tiga kegunaan, yaitu:
   a. Pemahaman kembali permasalahan bisnis oleh tim proyek;
   b. Menentukan permasalahan dipecahkan dengan cara terbaik;
   c. Menentukan sistem dapat dikembangkan dengan cara terbaik.

3. Tahap *definition*, tahap ini berguna untuk mengidentifikasi data, proses, antarmuka, dan kebutuhan geografis pengguna sistem baru.

4. Tahap *targeting*, tahap ini berguna untuk mengidentifikasi kandidat solusi, menganalisis kandidat solusi, dan merekomendasikan solusi yang ditargetkan akan dirancang dan diimplementasikan.

5. Tahap *purchasing*, tahap ini berguna untuk riset pasar teknologi informasi, mengumpulkan proposal dari penjual, dan merekomendasikan proposal yang memenuhi kebutuhan bisnis dan teknologi informasi ke manajemen.

6. Tahap *design*, tahap ini berguna untuk mentransformasikan kebutuhan bisnis yang diperoleh pada tahap definisi ke bentuk rancangan teknis.

7. Tahap *construction*, tahap ini berguna untuk membangun dan menguji fungsional sistem yang memenuhi kebutuhan bisnis dan perancangan, dan untuk mengimplementasikan antarmuka sistem lama dan baru.

8. Tahap *delivery*, tahap ini berguna untuk meng-*install*, menyebar, dan menempatkan sistem baru ke dalam operasi atau produksi.

Dalam sumber referensi yang lain, Raymond McLeod, Jr. dan George Schell (2001) menjelaskan bahwa dalam siklus hidup sistem (*Systems Life Cycle/SLC*), metodologi diartikan sebagai cara yang direkomendasikan untuk melakukan sesuatu hal. Dalam aplikasi, pendekatan pada tugas pengembangan sistem dengan menggunakan
sistem berbasis komputer seringkali disebut pendekatan “air terjun” (waterfall) yang meliputi:

1. Perencanaan (planning)
2. Analisis (analysis)
3. Perancangan (design)
4. Implementasi (implementation)
5. Penggunaan (use)

Personal yang terlibat dalam SDLC meliputi para personal dalam SIM, pengguna, dan spesialis yang dapat memberikan konsultasi. Dalam pendekatan tradisional, para spesialis bekerjasama dengan para pengguna, sedangkan dalam pendekatan baru menggunakan strategi outsourcing.

Tahap perencanaan meliputi:
1. Pengakuan permasalahan
2. Definisi permasalahan
3. Menetapkan himpunan sasaran
4. Identifikasi batasan
5. Mengarahkan studi kelayakan (teknik, ekonomi, hukum dan etika, operasional, jadwal)
6. Menyiapkan proposal studi proyek
7. Menyetujui atau menolak dilanjutkan atau tidak dilanjutkan
8. Memberikan mekanisme kontrol

Tahap analisis meliputi:
1. Mempublikasikan
2. Mengorganisasikan tim proyek
3. Mendefinisikan kebutuhan informasi
4. Mendefinisikan kriteria kinerja sistem
5. Menyiapkan proposal perancangan (dibandingkan dengan proposal studi)
6. Menyetujui atau menolak ke perancangan proyek

Tahap parancangan meliputi:
1. Menyiapkan detail rancangan
2. Mengidentifikasi konfigurasi alternatif sistem
3. Evaluasi konfigurasi
4. Memilih konfigurasi terbaik
5. Mempersiapkan proposal implementasi
6. Menyetujui atau menolak melanjutkan ke implementasi sistem

Tahap implementasi meliputi:
1. Perencanaan implementasi
2. Mengumumkan;
3. Menjelaskan sumber daya hardware
4. Menjelaskan sumber daya software
5. Menyiapkan database
6. Menyiapkan fasilitas fisik
7. Mendidik para personal yang terlibat dan pengguna
8. Menyiapkan proposal cutover
9. Menyetujui/menolak melanjutkan ke penggunaan sistem baru
10. Cutover ke sistem baru

Tahap penggunaan meliputi:
1. Penggunaan
2. Audit
3. Pemeliharaan
4. Menyiapkan proposal rekayasa ulang
5. Menyetujui atau menolak rekayasa ulang

Dalam sumber referensi yang lain, Avison dan Fitzgerald (1995) menyatakan, bahwa metodologi memuat fase-fase, masing-masing fase memuat sejumlah sub fase, yang menjadi petunjuk bagi pengembang sistem dalam memilih teknik yang tepat pada setiap tahapan pada proyek dan membantu perencanaan, pengelolaan, kontrol, dan evaluasi proyek Sistem Informasi. Menurut mereka, ada ribuan metodologi pengembangan Sistem Informasi yang sudah pernah dipakai, dan setiap metodologi memiliki perbedaan satu sama lain karena adanya perbedaan
penekanan, misal penekanan terhadap dimensi manusianya, pendekatan keilmiahannya, pendekatan yang pragmatis, pendekatan yang otomatis.


2.1.4.2. Kerangka Kerja Zachman

2.1.5. Perancangan Sistem Dalam SIM

Perancangan sistem (system design) merupakan tahapan setelah analisis sistem (system analysis) yang dapat dikelompokkan dalam dua bagian, yaitu perancangan sistem secara umum (general system design) atau perancangan konseptual (conseptual design) atau perancangan sistem logika (logical design) dan perancangan sistem secara terinci (detail system design) atau perancangan sistem secara fisik (physical system design) atau perancangan internal (internal design).

Tahap perancangan sistem mempunyai dua tujuan utama, yaitu untuk memenuhi kebutuhan pengguna sistem dan untuk memberikan gambaran yang jelas tentang rancang bangun yang lengkap kepada pemprogram (programmer) dan ahli teknik lain. Kebutuhan sistem (system requirement) yang harus diperhatikan dalam merancang SIM meliputi:

1. Keandalan (reability)
2. Ketersediaan (availability)
3. Keluwasan (*fleksibility*),
4. Jadwal instalasi (*installation schedule*),
5. Umur yang diharapkan (*life expectancy*) dan potensi pertumbuhan (*growth potential*),
6. Kemudahan dipelihara (*maintainability*).


1. Identifikasikan semua *external entity* sistem yang terlibat.
2. Identisikasikan semua *input* dan *output* yang terlibat dengan *external entity*.
3. Gambarlah terlebih dahulu diagram konteks atau diagram induk untuk garis besar, kemudian dipecah untuk level-level berikutnya.
4. Gambarlah *hierarchy chart* untuk semua proses dalam sistem untuk mempersiapkan penggambaran DFD level berikutnya.
5. Gambarlah sketsa DFD untuk *overview diagram* (level 0) berdasarkan proses bagan berjenjang.
6. Gambarlah DFD untuk level-level berikutnya, yaitu level 1, kemudian dipecah dalam level 2, dan seterusnya.
7. Setelah semua level DFD digambarkan, selanjutnya menggambar DFD untuk laporan ke manajemen secara terpisah.
8. Semua level DFD yang telah digambar termasuk DFD untuk laporan ke manajemen digabung dalam satu diagram.

Salah satu alat dokumentasi yang banyak digunakan dalam perancangan sistem adalah diagram HIPO. Menurut *Al-Bahra bin Ladjamudin (2005)*, HIPO untuk sistem dapat ditunjukkan dalam tiga
2.2. Tinjauan Tentang Konsep Database
2.2.1. Definisi Database

Sebuah definisi database yang telah cukup lama, namun cukup lengkap dan masih relevan telah diberikan oleh James Martin (1975), yaitu sekumpulan data terhubung (interrelated data) yang disimpan secara bersamasama pada suatu media tanpa “mengatap” satu sama lain atau tidak perlu suatu kerangkapan data, kalaupun terjadi kerangkapan maka kerangkapan tersebut harus seminimal mungkin dan terkontrol (controlled redundancy); data disimpan dengan cara-cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali; data dapat digunakan oleh satu atau lebih program aplikasi secara optimal; data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakan; data disimpan sedemikian rupa sehingga proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

keseluruhan data yang disimpan dalam sistem komputer yang menjadi sumber daya organisasi.

2.2.2. Tujuan dan Keuntungan Database

James Martin (1975) mengelompokkan tujuan database menjadi dua kelompok, yaitu tujuan primer dan tujuan sekunder. Tujuan primer merupakan tujuan utama yang ingin dicapai dalam perancangan dan pengembangan database.

Tujuan primer database meliputi penggunaan database oleh banyak pengguna, menjaga investasi intelektual, penekanan biaya, menghilangkan proliferasi, kinerja (performance), kejelasan (clarity), kemudahan pemakaian, fleksibilitas (flexibility), kebutuhan data yang tidak terantisipasi dapat dipenuhi dengan cepat, perubahan yang mudah, akurasi (accuracy) dan konsistensi (consistency), privasi (privacy), keamanan (security), serta ketersediaan (availability) (Martin, 1975).

Tujuan sekunder database merupakan tujuan tambahan yang diperlukan untuk mencapai tujuan primer, meliputi kebebasan data secara fisik (physical data independency), kebebasan data secara logik (logical data independency), pengendalian atau minimalisasi kerangkapan (data redundancy), kecepatan akses, kecepatan pencarian, standarisasi data, tersedianya kamus data, antarmuka pemrogram tingkat tinggi, bahasa end-user, integritas (integrity), kecepatan pemulihan kembali dari kerusakan (fast recovery from failuries), kemudahan untuk pengaturan (tuning), perancangan dan pengawasan alat-alat, serta pengorganisasian kembali atau migrasi data dapat dilakukan secara otomatis (Martin, 1975).

Kamran Parsaye, Mark Chignell, Setrag Khoshafian, dan Harry Wong (1989), menyatakan bahwa database memberikan keuntungan sebagai berikut:

1. Akses bersama data untuk pengguna-pengguna yang berbeda
2. Keamanan data
3. Meningkatkan kemudahan dan efisiensi dalam update untuk pemeliharaan data

Dalam referensi yang lain, C.J. Date (1995) menyatakan bahwa pengolahan data dengan pendekatan database memberikan keuntungan:

1. Kerangkapan data dapat diminimalkan
2. Inkonsistensi data dapat dihindari
3. Data dalam database dapat digunakan bersama (multiuser)
4. Standarisasi data dapat dilakukan
5. Pembatasan untuk keamanan data dapat diterapkan
6. Integritas data dapat terpelihara
7. Perbedaan kebutuhan data dapat diseimbangkan

Sedangkan Raymond McLeod Jr. dan George Schell (2001), menyatakan bahwa database memberikan keuntungan sebagai berikut:

1. Mengurangi kerangkapan data
2. Menghindari ketergantungan data
3. Memungkinkan integrasi data dari banyak file
4. Pemanggilan data dan informasi lebih cepat
5. Meningkatkan keamanan data

2.2.3. Kekangan Dalam Database

Untuk memenuhi batasan/kriteria sebagaimana definisi database, maka perancangan database memiliki beberapa kekangan yang harus ditaati, yaitu (Martin, 1975):

1. Kerangkapan data (data redundancy)
2. Inkonsistensi data (data inconsistency)
3. Data terisolasi
4. Keamanan (security)
5. Integrity

2.2.4. Pandangan Terhadap Database

Sebuah database dapat dipandang dari dua sisi, yaitu sisi pengguna dan sisi perancang. Seorang perancang dapat memiliki pandangan secara konseptual dan secara fisis. Pandangan-pandangan tersebut menunjukkan level abstraksi database yang sering disebut sebagai arsitektur database atau abstraksi database. Dalam hal ini terdapat penyebutan yang berbeda untuk setiap level abstraksi database.

James Martin (1975) menyatakan bahwa abstraksi database terdiri atas tiga level, yaitu application programmer logical file atau user view, global logical data atau level konseptual (conceptual view), dan physical view atau level internal.

Jeffrey D. Ullman (1988) menyebutkan bahwa tiga level abstraksi database adalah meliputi level pandangan (view level), level database konseptual (conceptual database level), dan level database fisik (physical database level).


Abraham Silberschatz, Henry F. Korth, dan S. Sudarshan (2001) membedakan level abstraksi database menjadi tiga, yaitu: pandangan eksternal (external view) atau pandangan pengguna (user view), pandangan konseptual (conceptual view) atau pandangan komunitas pengguna (community user view), dan pandangan internal (internal view) atau pandangan media penyimpan sekunder (storage view).

Level user view merupakan pandangan para pengguna database dimana masing-masing dapat memiliki pandangan yang berbeda tergantung pada macam data apa saja yang tersedia. Pengguna tidak perlu mengetahui teknis penyimpanan database dalam media penyimpan. User view dijelaskan oleh schema dan subschema database, sedangkan nilai aktual data ditunjukkan oleh instance schema. Level conceptual view merupakan pandangan perancang database yang berkaitan dengan
data apa saja yang perlu disimpan dan penjelasan mengenai hubungan antara data yang satu dengan lainya. **Global logical data** ditunjukkan oleh definisi struktur tabel **database** menggunakan bahasa deskripsi data (**Data Definition Language/DDL**). Sedangkan **physical view** merupakan implementasi **conceptual view**, yaitu pandangan yang berkaitan dengan teknik penyimpanan data ke dalam fisik media penyimpanan data yang digunakan, meliputi metoda penyimpanan dan metoda akses data dalam media penyimpan sekunder (**storage device**).

### 2.2.5. Relational Database Model/RDBM


Relasi RDBM mempunyai beberapa karakteristik yang harus dipenuhi, yaitu (Martin, 1975):

1. Semua elemen data/entri pada suatu relasi harus mempunyai nilai tunggal (**single value**), bukan suatu larik atau grup perulangan dan harus berupa nilai yang tidak dapat dibagi lagi (**atomic value**)
2. Semua elemen data/entri pada suatu atribut tertentu dalam sebuah relasi harus mempunyai tipe dan ukuran yang sama
3. Masing-masing atribut dalam sebuah relasi mempunyai nama yang unik
4. Pada sebuah relasi tidak ada dua record data yang identik

Setiap relasi RDBM tersusun atas dua komponen, yaitu (Martin, 1975):

1. **Intension**, meliputi struktur penamaan (naming structure) relasi dan batasan integritas (**integrity contraint**) yaitu batasan integritas
kesatuan (entity integrity) dan batasan integritas referensial (referential integrity)

2. Extension, menunjukkan nilai-nilai aktual elemen data/entri yang tersimpan dalam relasi pada suatu saat tertentu

RDBM memiliki terminologi penggunaan istilah khusus yang berbeda dengan model data lainnya, seperti ditunjukkan pada Tabel 2.4.

<table>
<thead>
<tr>
<th>Istilah formal</th>
<th>Istilah non formal</th>
<th>Keterangan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element data (data element), nilai data (data item), entri (entry)</td>
<td>Kolom, median data, median, field</td>
<td>Sekelompok nilai data yang mempunyai arti. Atribut memiliki tipe, ukuran, dan domain yang sama.</td>
</tr>
<tr>
<td>Atribut (attribute)</td>
<td>Kolom, median data, median, field</td>
<td>Sekumpulan atribut yang mempunyai hubungan terhadap objek tertentu</td>
</tr>
<tr>
<td>Record/tuple</td>
<td>Baris/tabel</td>
<td>Jumlah atribut dalam sebuah relasi</td>
</tr>
<tr>
<td>Relation (relation)</td>
<td>Tabel</td>
<td>Jumlah record dalam sebuah relasi</td>
</tr>
<tr>
<td>Degree (degree)</td>
<td>Antas (arity)</td>
<td>Jumlah record dalam sebuah relasi</td>
</tr>
<tr>
<td>Cardinality (cardinality)</td>
<td>Ganda, sepasang</td>
<td>Hubungan antar relasi</td>
</tr>
<tr>
<td>Unary relation</td>
<td>Relasi yang tersusun oleh satu atribut</td>
<td></td>
</tr>
<tr>
<td>Binary relation</td>
<td>Relasi yang tersusun oleh dua atribut</td>
<td></td>
</tr>
<tr>
<td>Ternary relation</td>
<td>Relasi yang tersusun oleh tiga atribut</td>
<td></td>
</tr>
<tr>
<td>n-ary relation</td>
<td>Relasi yang tersusun oleh n atribut</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Setu atau gabungan atribut bersifat unik yang digunakan untuk membedakan setiap record dalam relasi</td>
<td></td>
</tr>
<tr>
<td>Candidate Key/CK</td>
<td>Setu atau gabungan minimal atribut bersifat unik yang dapat digunakan untuk mengidentifikasi setiap record dalam relasi</td>
<td></td>
</tr>
<tr>
<td>Primary Key/PK</td>
<td>Bagian dari CK yang dipilih digunakan sebagai kunci utama dalam relasi</td>
<td></td>
</tr>
<tr>
<td>Alternate Key/AK</td>
<td>Bagian dari CK yang tidak dipilih digunakan sebagai kunci utama dalam relasi</td>
<td></td>
</tr>
<tr>
<td>Foreign Key/FK</td>
<td>Kunci timu</td>
<td>Satu atau gabungan sembarangan atribut yang menjadi PK dalam relasi lain yang mempunyai hubungan secara logik</td>
</tr>
<tr>
<td>Domain</td>
<td>Himpunan nilai yang memenuhi syarat</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Deskripsi hubungan logik secara global, termasuk di dalamnya nama dan deskripsi tipe dan ukuran atribut dan hubungan logik antar nilai database dalam lingkup sebuah sistem</td>
<td></td>
</tr>
<tr>
<td>Subschema</td>
<td>Deskripsi hubungan logik secara terpisah, termasuk di dalamnya nama dan deskripsi tipe dan ukuran atribut dan hubungan logik antar relasi database dalam lingkup sebuah sub sistem aplikasi database</td>
<td></td>
</tr>
</tbody>
</table>

Sumber: James Martin, 1975

Dalam RDBM, kunci relasi diperlukan untuk pengaksesan data dalam relasi atau untuk menyusun kerelasan antar relasi. Kunci relasi merupakan satu atau gabungan atribut yang bersifat unik dan dapat digunakan untuk mengidentifikasi setiap record dalam relasi. Sifat unik tersebut diartikan bahwa nilai-nilai elemen data/entri dalam atribut yang digunakan sebagai kunci relasi tidak boleh ada yang sama untuk
seluruh record dalam relasi. Berdasarkan jumlah atribut penyusunnya, kunci relasi dapat dibedakan menjadi dua jenis, yaitu (Martin, 1975):

1. Kunci sederhana (simple key)
2. Kunci komposit (composite key)

Sedangkan berdasarkan macamnya, kunci relasi meliputi (Martin, 1975):

1. Kunci kandidat (Candidate Key/CK)
2. Kunci primer (Primary Key/PK)
3. Kunci alternatif (Alternate Key/AK)
4. Kunci penghubung/kunci tamu/kunci asing (Foreign Key/FK)

Pemilihan kunci relasi memiliki beberapa aturan (rules), yaitu:

1. Integritas kesatuan/integritas entitas (entity integrity), bahwa secara kesatuan nilai-nilai elemen data/entri pada atribut yang dipilih/digunakan sebagai PK tidak boleh null
2. Integritas referensial (referential integrity), bahwa di dalam kerelasian antar dua atau lebih relasi dalam database yang dihubungkan dengan suatu kunci penghubung (foreign key/FK), maka hubungan antar relasi tersebut harus menjamin bahwa setiap nilai elemen data/entri pada FK dalam relasi anak harus ada record dengan nilai elemen data/entri yang sama pada relasi induk yang menjadi referensi (Martin, 1975).


Penyimpangan dalam pengolahan data dapat terjadi akibat ketergantungan antar nilai rinci data. Ketergantungan antar nilai rinci data menjabarkan hubungan antara atribut-atribut dalam hal bagaimana suatu nilai menentukan nilai yang lain. Umumnya, yang paling baik dilakukan adalah jika struktur relasi dirancang sedemikian rupa sehingga
atribut-atribut non kunci hanya bergantung pada PK dan tidak memiliki ketergantungan pada atribut lainnya. Jenis-jenis ketergantungan antar nilai rinci data adalah (Martin, 1975):

   
   **Keterangan:**
   
   **FD** : Functionally Dependence
   **R** : nama relasi
   **X** : atribut penentu (determines)
   **Y** : atribut yang bergantung (dependent)


   **Keterangan:**
   
   **FFD** : Full Functionally Dependency
   **R** : nama relasi
   **X** : atribut penentu (determines)
   **Y** : atribut yang bergantung (dependent)


   **Notasi:** **TDF: R.X \(\rightarrow\) R.Y \(\rightarrow\) R.Z**

   **Keterangan:**
   
   **TDF** : Transitive Dependency
   **R** : nama relasi
   **X** : atribut penentu (determines)
   **Y** : atribut yang bergantung (dependent) terhadap X dan sekaligus penentu terhadap Z
   **Z** : atribut yang bergantung (dependent) terhadap Y dan sekaligus bergantung pada X

4. Ketergantungan total (Total Dependency/ TD), atribut Y dikatakan mengalami ketergantungan total, jika Y functionally dependency pada X dan X functionally dependency pada Y. Notasi: **TD: R.X \(\leftrightarrow\) R.Y**

   **Keterangan:**
   
   **TD** : Total Dependency
   **R** : nama relasi
   **X** : atribut penentu (determines), sekaligus bergantung pada Y
Y : atribut yang bergantung (dependent) sekaligus penentu pada X

Untuk memenuhi batasan/kriteria database, maka setiap rancangan relasi perlu diuji untuk menentukan apakah setiap relasi yang dirancang telah berada dalam kondisi optimal dan tidak menimbulkan permasalahan (anomalies) saat pengolahan data. Jika relasi belum optimal, maka perlu dilakukan proses normalisasi, yaitu suatu teknik yang menstrukturkan data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam database (Martin, 1975).


1. Memiliki struktur record yang konsisten secara logik
2. Memiliki struktur record yang mudah untuk dimengerti
3. Memiliki struktur record yang sederhana dalam pemeliharaan
4. Memiliki struktur record yang mudah untuk ditampilkan kembali
5. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem

Dalam sumber referensi yang lain Abraham Silberschatz, Henry F. Korth, dan S. Sudarshan (2001) menyatakan bahwa normalisasi bertujuan untuk memperoleh relasi-relasi dalam bentuk yang “baik”. Jika sebuah relasi R memiliki bentuk yang “tidak baik”, maka relasi tersebut dipecah menjadi sejumlah relasi \{R_1, R_2, ..., R_n\} dimana setiap relasi memiliki bentuk yang “baik” dan pemecahan dilakukan tanpa terjadi kehilangan informasi saat digabungkan kembali (losslessjoin).

Teori normalisasi didasarkan pada ketergantungan fungsional (functionally dependency/FD) dan ketergantungan pada banyak nilai (multivalue dependency). Bentuk-bentuk normal yang dikenal hingga saat ini adalah (Martin, 1975):

1. Bentuk tidak normal (Unnormalized Form/UNF)
Relasi UNF terjadi jika relasi mempunyai bentuk non flat file, relasi memuat set atribut berulang (non single value), dan relasi memuat atribut non atomic value.

2. Bentuk 1NF
Kriteria relasi 1NF adalah jika seluruh atribut dalam relasi bernilai atomik (atomic value), seluruh atribut dalam relasi bernilai tunggal (single value), relasi tidak memuat set atribut berulang, dan semua record mempunyai sejumlah atribut yang sama.

3. Bentuk 2NF
Kriteria relasi 2NF adalah jika memenuhi kriteria 1NF dan semua atribut non kunci FD pada PK.

4. Bentuk 3NF
Kriteria relasi 3NF adalah jika memenuhi kriteria 2NF dan setiap atribut non kunci tidak TDF terhadap PK.

5. Bentuk BCNF
Kriteria relasi BCNF adalah jika memenuhi kriteria 3NF dan semua atribut penentu (determinan) merupakan CK.

6. Bentuk 4NF
Kriteria relasi 4NF adalah jika memenuhi kriteria BCNF dan setiap atribut tidak mengalami ketergantungan pada banyak nilai.

5. Bentuk 5NF
Kriteria relasi 3NF adalah jika kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang memuat atribut yang lebih sedikit.

6. Bentuk DKNF
Kriteria relasi DKNF adalah jika setiap batasan dapat disimpulkan secara sederhana dengan mengetahui sekumpulan nama atribut dan domainnya selama menggunakan sekumpuan atribut pada kuncinya.

Kerelasian antar data yang disimpan dalam database sangatlah kompleks. Suatu schema dan subschema diperlukan untuk mendeskripsikan hubungan logik antar data dalam database. Schema
memberikan deskripsi hubungan logik antar data dalam database secara lengkap, termasuk di dalamnya nama dan deskripsi dari semua atribut, record, dan batasan nilai untuk semua aplikasi yang menggunakan. Sedangkan subschema merupakan deskripsi terpisah dari atribut, record, dan batasan nilai yang akan digunakan oleh sebuah program aplikasi. Dengan demikian, dari sebuah schema dapat diturunkan ke dalam beberapa subschema. Suatu schema menunjukkan pandangan konseptual seorang perancang, sedangkan subschema menunjukkan pandangan seorang application programmer terhadap data yang digunakannya.

Raymond McLeod dan George Schell (2001) menyatakan bahwa schema memuat deskripsi yang meliputi:

1. Nama field data
2. Alias (nama lain yang digunakan untuk field data yang sama)
3. Tipe data (numerik atau alphabet)
4. Jumlah digit pada posisi angka
5. Jumlah digit pada posisi desimal
6. Sejumlah aturan integritas

Sebuah relasi RDBM direpresentasikan sebagai sebuah tabel datar yang memuat beberapa keterangan, yaitu:

1. Nama relasi
2. Kunci-kunci relasi (tidak selalu dituliskan)
3. Kunci-kunci indeks (tidak selalu dituliskan)
4. Sejumlah record (elemen data/entri dalam bentuk baris data)
5. Nama-nama atribut

Notasi schema dan subschema:

namarelasi_schemas : (namaattribut1 tipe[ukuran]1, namaattribut2 tipe[ukuran]2, ….,namaattributn tipe[ukuran]n, foreign key (namaattributfk) references namarelasiinduk, primary key (namaatributpk))

Keterangan:
namarelasi : nama relasi
schema : schema (bisa juga subschema)
2.3. Database dan SIM


Sementara itu, Raymond McLeod Jr. dan George Shell (2001) menyatakan bahwa salah satu sumber daya konseptual informasi dalam organisasi adalah berupa database. Dapat dipahami bahwa database merupakan sumber daya penting dalam SIM.

2.4. Perancangan Database Untuk SIM

Menurut Brian Mullen (2005), penyusunan database merupakan tugas multidisipliner. Pada satu sisi, perancang database sebagai bagian staf teknik memahami konsep database dengan baik, tetapi sering tidak mengetahui bagaimana membuat data relevan bagi orang lain dalam

Brian Mullen (2005) juga menyatakan, bahwa rancangan database yang benar akan memberikan landasan yang solid untuk SIM.

Menurut Jan L. Harrington (2004), suatu rancangan database yang buruk akan mengakibatkan efek negatif, antara lain:
1. Munculnya kerangkapan data
2. Munculnya inkonsistensi data
3. Munculnya permasalahan pada penyisipan data
4. Munculnya permasalahan pada penghapusan data

Dalam referensi yang lain, Abraham Silberschatz, Henry F. Korth, dan S. Sudarshan (2001) menyatakan bahwa perancangan database relasional diperlukan untuk mendapatkan sekumpulan schema relasi yang baik. Rancangan yang buruk akan mengakibatkan perulangan informasi dan tidak dapat menampilkan kembali informasi tertentu. Tujuan utama perancangan database adalah:
1. Menghindari kerangkapan data
2. Menjamin bahwa kerelasian antar atribut dapat direpresentasikan
3. Memberikan fasilitas pengecekan batasan integritas pada proses update
Sementara itu, J. L. Whitten dan L. D. Bentley (1998), menyatakan bahwa tujuan dan prasyarat perancangan database adalah:

1. Database harus memberikan efisiensi media penyimpan (storage), update, dan penampilan kembali data-data.
2. Database harus andal, yaitu memiliki integritas tinggi yang memberikan kepercayaan bagi para pengguna terhadap data.
3. Database harus dapat beradaptasi (adaptable) dan dapat berkembang (scaleable) untuk memenuhi kebutuhan dan aplikasi baru.

Untuk memperoleh SIM yang baik, maka pengembang SIM perlu memahami metode perancangan database yang baik, yaitu:

1. Mengidentifikasi kebutuhan informasi pada sebuah bisnis
2. Merancang database relasional yang didasarkan pada kebutuhan bisnis
3. Membuat dokumentasi database
4. Meningkatkan fleksibilitas database untuk mengantisipasi perubahan
5. Menyederhanakan database dengan jumlah tabel
6. Membangun database relasional dan menyesuaikannya untuk memperoleh kinerja yang optimal
7. Meningkatkan kinerja database dengan indexing dan mengkontrol kerangkapan
8. Mengurangi beaya pengembangan software dengan generalisasi (http://www2.cstudies.ubc.ca/~mullen/IEDBS.html).


1. Kamus data (Data Dictionary/DD)
2. Kebutuhan pengguna
3. Diagram Aliran Data/DAD (Data Flow Diagram/DFD)
4. Bagan struktur (structure chart)
5. Model data (data model)
6. Prototipe tampilan layar  
7. Model keadaan (*state model*)  
8. Diagram tugas (*task diagram*)  
9. Diagram kelas (*class diagram*)  
10. Diagram obyek (*object diagram*)

Dalam referensi yang lain, Raymond McLeod dan George Schell (2001) menyatakan bahwa penyusunan *database* dapat dilakukan dengan menggunakan dua pendekatan, yaitu:

1. Pendekatan berorientasi proses (pendekatan pemecahan permasalahan)  
2. Pendekatan model perusahaan. Selanjutnya, penyusunan *database* dilakukan melalui tiga langkah utama, yaitu:
   a. Mendeskripsikan data  
   b. Memasukkan data  
   c. Menggunakan *database* (menggunakan bahasa *query*, *QBE*, atau *DML*)


Berikut adalah saran umum yang diberikan oleh Brian Mullen (2005) untuk memperoleh rancangan struktur tabel database yang fleksibel, yaitu (http://www2.cstudies.ubc.ca/~mullen/IEDBS.html#General Tables):
1. Meningkatkan fleksibilitas database dimana kebutuhan-kebutuhan baru dapat diakomodasi dengan tabel-tabel database yang ada
2. Mengurangi beaya pembuatan dan pemeliharaan aplikasi-aplikasi
3. Meningkatkan penggunaan kembali dan penggunaan bersama antar aplikasi
4. Mengurangi software yang dibutuhkan untuk membuat aplikasi
5. Mengurangi duplikasi dan inkonsistensi informasi


J. L. Whitten dan L. D. Bentley (1998) juga menyatakan bahwa keuntungan maksimal dari database hanya bisa dicapai jika database


Menurut Paul Litwin (2003), perancangan database lebih merupakan suatu seni daripada suatu ilmu pengetahuan. Sekalipun tidak mengungkap seluruh tahapan dalam proses perancangan, Paul Litwin memberikan kerangka kerja yang sesuai digunakan dalam perancangan database, yaitu sebagai berikut (http://r937.com/relational.html):
1. Pelajariilah proses bisnis dan proses lain dalam organisasi untuk mencoba membuat model.
2. Tulislah pernyataan mendasar atau misi yang harus dilakukan oleh sistem. Kemudian dilanjutkan dengan membuat daftar kebutuhan pada sistem.


5. Perhatikan seluruh laporan yang tercetak pada kertas atau komputer.

6. Pastikan bahwa setiap data dalam laporan tersedia di dalam tabel. Jika data belum tersedia, tambahkan data tersebut dalam tabel-tabel yang ada atau buatlah tabel baru.

7. Tambahkan dan isikan beberapa baris data pada setiap tabel.


10. Pastikan bahwa tabel memenuhi 1NF, yaitu seluruh atribut bernilai atomik dan tidak memuat grup perulangan. Jika diperlukan, pecahkan tabel untuk memperoleh 1NF.

11. Pastikan bahwa tabel memenuhi 2NF, yaitu setiap tabel hanya mendeskripsikan sebuah entitas dan semua atribut non-key bergantung sepenuhnya (FFD) pada PK. Jika diperlukan, pecahkan tabel untuk memperoleh 2NF. Jika tabel memiliki PK berupa kunci komposit, maka pecahkan PK menjadi atribut-attribut yang seluruhnya ditempatkan pada tabel itu juga.
12. Pastikan bahwa tabel memenuhi 3NF, yaitu hilangkan atribut yang nilainya dapat dihitung dan hilangkan atribut yang bersifat mutual dependent dengan cara membuat tabel lookup.
17. Kembali ke rancangan tabel dan aturan bisnis.

Paul Litwin (2003) juga menyatakan bahwa penggunaan model RDBM dalam perancangan database akan memberikan beberapa keuntungan, antara lain:
1. Entri data, update, dan penghapusan data akan lebih efisien
2. Penampilan kembali, peringkasan, dan pelaporan data akan lebih efisien
3. Jika database dirancang dengan menggunakan model yang diformulasikan dengan baik, maka perilakunya akan dapat diprediksi
4. Jika informasi disimpan dalam database (bukan dalam program aplikasi) maka database memiliki dokumentasi tersendiri
5. Perubahan pada schema database dapat dilakukan dengan lebih mudah

Dalam referensi yang lain, Jun Yang (1999) menyatakan bahwa penyusunan database dapat dilakukan melalui empat tahapan, yaitu:
1. Memahami domain dunia nyata yang akan ditangkap
2. Menspesifikasikan domain dunia nyata tersebut dengan menggunakan model data (menggunakan ER_M atau Object Definition Language/ODL)
3. Menterjemahkan spesifikasi ke model database (relasional, ODL)


<table>
<thead>
<tr>
<th>Kriteria</th>
<th>Deskripsi</th>
</tr>
</thead>
</table>

Tabel 2.5: Aspek-aspek kualitas schema database yang sesuai untuk dilakukan pengukuran
Tingkat detail (level of detail)
Apakah tingkat detail schema telah sesuai. Aspek ini penting untuk mengetahui apakah schema diterima oleh pengguna atau tidak. Pengukuran dilakukan menggunakan kepakaran teknik untuk menentukan tingkat detail yang diinginkan dan membandingkannya dengan schema.

Kelengkapan (completeness)
Apakah schema telah lengkap (dengan mengacu pada kebutuhan). Aspek ini penting untuk mengetahui apakah schema dapat diterima oleh pengguna atau tidak. Pengukuran dapat dilakukan dari aspek jangkauan dan tingkat detail (sebelumnya).

Minimalitas (minimality)
Apakah schema dimodelkan secara kompak dan tidak ada perulangan. Aspek ini penting karena schema konseptual harus tepat. Pengukuran dilakukan menggunakan kepakaran teknik untuk mengecek apakah terdapat aspek-aspek yang dimodelkan secara berulang.

Kemampuan untuk diintegrasikan (ability of integration)
Apakah schema telah disesuaikan untuk standarisasi dalam organisasi secara menyeluruh atau pemodelan. Aspek ini tertanggung pada konteks proyek atau organisasi, yaitu apakah lebih banyak terlibat pada asosiasi ekonomi atau berorientasi internasional, tetapi yang lebih relevan jika memiliki kemampuan untuk diiintegrasikan. Pengukuran dilakukan untuk menentukan apakah penggunaan nama-nama hanya dapat digunakan untuk cabang atau dapat diterima secara internasional.

Kemampuan untuk dibaca (readability)
Apakah semua istilah dalam schema dijelaskan atau tidak. Aspek ini penting untuk dialog dengan staf teknik dan orang yang datang belakangan. Pengukuran dilakukan dengan meninjau dokumen untuk menentukan apakah setiap aspek mudah dipahami.

Sumber: Olaf Herden (2001)

<table>
<thead>
<tr>
<th>Kriteria</th>
<th>Kriteria</th>
<th>Alasan ketidaksesuaian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kemampuan diperluas (extensibility)</td>
<td>Dapatkah kebutuhan baru dapat diintegrasikan ke dalam schema secara mudah?</td>
<td>Sangat sesuai untuk setiap schema, karena 80% beaya SIM digunakan untuk pemeliharaan. Namun orientasi obyek sebagai model data untuk level konseptual baik digunakan untuk dapat diperluas.</td>
</tr>
<tr>
<td>Kemampuan diubah penting karena schema Normalisasi</td>
<td>Apakah schema memenuhi aturan normalisasi?</td>
<td>Kriteria ini sangat relevan dalam RDBM, sehingga penggunaan ER_M atau perlucasanya sebagai model konseptual perlu segera dan aturan normalisasi harus</td>
</tr>
</tbody>
</table>

| Tabel 2.6: Aspek-aspek kualitas schema database berorientasi obyek yang tidak sesuai untuk dilakukan pengukuran |
Verifikasi struktur database adalah suatu proses memverifikasi lay out database apakah benar dan tepat sesuai dengan situasi-keadaannya. Verifikasi database melibatkan pengecekan rancangan database mengikuti praktek terbaik yang diusulkan oleh para pakar industri dan kebutuhan-kebutuhan khusus yang meyakinkan organisasi atau proyek; serta memberikan jaminan bahwa struktur database tidak diubah secara tidak terduga. Lebih khusus, verifikasi struktur database adalah proses untuk mengecek bahwa item-item diindeks dengan tepat; nilai-nilai disimpan dalam tipe data yang tepat; hubungan antar tabel-tabel diekspresikan dengan semestinya; dan seterusnya.

Jika struktur database dibuat mengikuti praktek yang terbaik, maka beberapa permasalahan yang terkait dengan database akan dapat dihindari. Sebagai contoh adalah:

1. Indeks yang tepat akan mempercepat kecepatan query (khususnya database berukuran besar)
2. Normalisasi database sesuai digunakan untuk menyederhanakan tabel-tabel pada saat perancangan (namun akan menambah jumlah tabel dan akan mengurangi integritas referensial)
3. Statemen SQL merupakan fasilitas untuk penamaan, pembentukan, dan pencegahan kesalahan tabel dan field logik selama pengujian, debugging, dan pemeliharaan
4. Tipe dan ukuran data yang tepat akan menghindari kemungkinan kehilangan dan kesalahan nilai-nilai data
5. Namanama tabel dan field yang berbeda dengan kata-kata kunci tercadang (reserved keywords) mencegah permasalahan fungsionalitas aplikasi.
Perancangan struktur *database* memerlukan waktu yang lama dan upaya-upaya perencanaan yang baik, yaitu bagaimana mengorganisasikan data ke dalam tabel-tabel; bagaimana merelasikan antar tabel; serta *field* mana yang akan diindeks, disimpan, dan diakses. Jika struktur tabel perlu diubah secara tiba-tiba dan perubahan itu tidak terdeteksi sebelumnya, maka permasalahan ini dapat lebih mudah ditangani (http://www.parasoft.com/jsp/home.jsp).

2.6. Schema Database Universal

Perancangan *schema database* yang bersifat universal untuk memenuhi berbagai macam kebutuhan para pengguna telah dilakukan dalam sebuah diskusi di Internet (http://www.webservertalk.com/message1055091-2.html). Diskusi diawali oleh pernyataan yang tidak mempercayai bahwa seseorang mampu membangun tabel/view database yang multiguna dan tidak memerlukan perancangan ulang saat terjadi perubahan kebutuhan dari para pengguna ("Chris"<cw23@greenmail.ch). Tanggapan pertama atas penyataan di atas disampaikan oleh Lauri Pietarinen (lauri.pietarinen@atbusiness.com). Aspek pertama fleksibilitas struktur *database* adalah permasalahan yang terkait dengan isolasi data dalam *database*, yaitu kopel antar tabel dalam *database*.

Berikut diberikan contoh definisi tabel *database* sederhana, yaitu:

```
CREATE TABLE OneGiantTable
(row_type char(20), row_id int, column_name char(20), column_value vchar(100))
```

Contoh di atas memerlukan inventarisasi kembali seluruh aspek penanganan *database* pada alat bantu yang digunakan. Aspek kedua, terkait dengan penggunaan pendekatan dalam struktur data. Penggunaan XML (yang merupakan pendekatan tidak/semi terstruktur dimana sebuah file dapat memuat berbagai macam nilai yang dituliskan di antara tanda *tag*) akan menemui permasalahan jika file memuat *tag* yang tidak dikenali oleh aplikasi. Sehingga di kemudian hari perlu inventarisasi kembali konteks data dan sekaligus menjadi alasan mengapa sebaiknya
kembali ke data terstruktur menggunakan tabel. Hal ini menjadi alasan, bahwa RDBM merupakan pendekatan paling umum untuk perancangan schema database. Selanjutnya diberikan dua contoh definisi tabel database, yaitu:

**DB1:**
```sql
create table Customer
(cust_id integer primary key, cust_name varchar(30),
 voice_phone varchar(20), fax_phone varchar(20))
```

**DB2:**
```sql
create table Customer
(cust_id integer primary key, cust_name varchar(30))
create table CustPhone
(phone_id integer primary key, cust_id integer references Customer,
 phone_number_type char(5), phone_number varchar(20))
```

Dalam contoh di atas, **DB2** lebih fleksibel daripada **DB1** dan schema tidak harus diubah apabila terjadi suatu saat terdapat tipe baru pada nomor telepon. Pada sisi yang lain, **DB1** lebih sesuai digunakan dan lebih mudah diakses (menggunakan query) apabila aturan yang berlaku adalah customer hanya memiliki sebuah nomor telepon. Fleksibilitas akan mengurangi kebutuhan perubahan schema, program-program aplikasi, dan membuat schema menjadi lebih komunikatif.

Aspek ketiga adalah terkait dengan penggunaan tipe data dalam schema tabel database. Permasalahannya adalah apakah semua tipe data harus didefinisikan secara eksplisit dalam struktur tabel atau dapat disembunyikan. Berikut contoh yang relevan untuk dipikirkan:

**DB1:**
```sql
create table Picture
(pict_id integer primary key, pict_name varchar(30),
 .. dan seterusnya ...
picture image)
```

**DB2:**
```sql
create table Picture
(pict_id integer primary key, pict_name varchar(30),
 .. dan seterusnya ...) create table Pixel
(pixel_id integer,
pixel_id integer, pixel_color integer, primary key(pixel_id, pixel_id))
```
Dalam DB1 kerumitan disembunyikan dari tipe data gambar, sedangkan dalam DB2 tipe data dituliskan secara eksplisit. Dalam sebagian besar kasus, pendekatan pada DB1 lebih disarankan. Pada sisi lain, tidak baik menyembunyikan tipe data untuk kasus seperti berikut:

```sql
create table Customer
  cust_id integer primary key, cust_details cust_detail_datatype)
select cust_details.getName()
from Customer
where cust_id=1
```

Sebenarnya, sah-sah saja menyembunyikan definisi tipe data, namun demikian untuk kasus terakhir ini bukan merupakan sebuah rancangan yang baik (lauri.pietarinen@atbusiness.com). Pendapat tersebut disanggah oleh Kenneth Downs (Kenneth@(Sec)ure (Dat)a(.com)), menurutnya pendekatan yang dipilih adalah tergantung pada permasalahan mana yang ingin diatasi. Schema tidak fleksibel dan tidak dapat memenuhi segala kebutuhan karena jenis kebutuhan apa yang akan terjadi di masa mendatang dan harus disiapkan dalam schema tidak diketahui. Sebuah pendekatan dapat digunakan, yaitu schema dirancang sesuai dengan kebutuhan dalam sistem dan mengungkapkan kelemahan-kelemahan yang terkait dengan struktur tabel untuk kebutuhan dalam jangka panjang. Fleksibilitas struktur database merupakan sebuah pilihan yang perlu memikirkan keseimbangan antara beban kerja, beaya, dan waktu yang diperlukan.

2.7. Pendekatan Dalam Pengukuran Kualitas Informasi


1. Pendekatan intuitif (*intuitive approach*), pendekatan ini mengusulkan atribut atribut kualitas informasi yang didasarkan pada wawasan/pengetahuan subyektif seseorang.


3. Pendekatan teoritis (*theoretical approach*), pendekatan ini berusaha memunculkan usulan teori baru didasarkan pada teori-teori yang sudah ada.

   Kelemahan utama pendekatan intuitif dan empiris adalah pengaruh tingkat pengetahuan peneliti terhadap penilaian yang diberikan. Sedangkan kelemahan pendekatan teoritis adalah kemungkinan terjadinya kesalahan pada teori baru yang diusulkan (Klesse, Herrmann, Brändli, Mügeli, Maier, http://wotan.liu.edu/docis/dbl/iqiqiq/2004_418_CIPACS.htm).
BAB III
TUJUAN DAN MANFAAT PENELITIAN

3.1. Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan sebagai berikut:
1. Melakukan analisis aspek-aspek kualitas schema database pada database akademik di ISTA
2. Mengetahui kualitas schema database yang digunakan pada database akademik di ISTA

3.2. Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat berupa rekomendasi alternatif-alternatif solusi yang mungkin dilakukan oleh pengelola/penaggungjawab database akademik di ISTA, terkait dengan kualitas schema database pada database akademik, meliputi 9 (sembilan) kriteria berikut:
1. Kebenaran (correctness)
2. Konsistensi (consistency)
3. Relevansi (relevance)
4. Jangkauan (scope)
5. Tingkat detail (level of detail)
6. Kelengkapan (completeness)
7. Minimalitas (minimality)
8. Kemampuan untuk diintegrasikan (ability of integration)
9. Kemampuan untuk dibaca (readability)
BAB IV
METODOLOGI PENELITIAN

4.1. Metode Penelitian

1. Sumber Data Penelitian
   Data yang akan dianalisis dalam penelitian ini adalah bersumber dari UPT PUSKOM, ISTA, Yogyakarta.

2. Data Penelitian
   Data yang akan dianalisis dalam penelitian ini adalah berupa schema database akademik yang digunakan di UPT PUSKOM, ISTA, Yogyakarta.

3. Metode Pengumpulan Data
   Pengumpulan data yang akan dianalisis dalam penelitian ini adalah dilakukan dengan menggunakan metode observasi.

4. Peralatan Yang Digunakan:
   a. Perangkat Keras
      Perangkat keras yang digunakan dalam penelitian ini adalah seperangkat komputer dengan spesifikasi sebagai berikut:
      1). Processor AMD Athlon(tm) XP 2500 + cache size: 512 KB
      2). RAM 256 Mbyte
      3). Hardisk Seagate ST340014A, 78165360 sectors (40021 MB) w/2048KiB Cache, CHS=4865/255/63, UDMA(100)
      4). IDE Controller: nVidia Corporation nForce2 IDE
   b. Perangkat Lunak
      Perangkat lunak yang digunakan dalam penelitian ini adalah:
      1). Sistem Operasi GNU/Linux Distribusi Slackware 9.1 Kernel 2.6.7.
      2). PostgreSQL 7.4.14

5. Langkah Penelitian
   Penelitian ini dilaksanakan dengan langkah sebagai berikut:
   a. Studi pendahuluan
      Pada langkah ini dilakukan kajian teori/konsep SIM, sistem informasi akademik, database, perancangan database, serta aspek-aspek kualitas schema database. Kajian teori/konsep ini
dilakukan sebagai dasar untuk melakukan analisis aspek-aspek kualitas schema database untuk SIAKAD.

b. Pengumpulan data
Pada langkah ini dilakukan pengumpulan data schema database untuk SIAKAD dengan menggunakan metode yang telah ditetapkan.

c. Instalasi Database Server PostgreSQL
Instalasi Database Server PostgreSQL dengan menggunakan sistem operasi GNU-/Linux debian dilakukan oleh user root dengan perintah berikut:

```
apt-get install Postgresql-7.4.
```

d. Pembuatan Duplikasi Database Akademik
Setelah Database Server terinstall maka dibuat duplikasi database akademik dengan melakukan dump pada schema database di server pada UPT Puskom ISTA dengan menjalankan perintah berikut:

```
pg dump -s akademik > akademik
```

Program pg dump merupakan utilitas bawaan dari PostgreSQL yang digunakan untuk membackup database ke dalam suatu file teks. Perintah di atas akan melakukan backup pada database akademik pada bagian schema saja.

e. Membuat Database Duplikasi
Setelah dilakukan perintah pg dump maka dilanjutkan dengan memasukan hasil backup ke dalam komputer yang digunakan untuk melakukan penelitian. Sedikit perubahan dilakukan dalam file hasilbackup disesuaikan dengan keadaan database server tempat penelitian dilakukan. Perubahan yang dilakukan adalah dengan menganti hak kepemilikan database dari user jack menjadi wa2n. Selanjutnya memasukan script backup yang telah dimodifikasi dalam database server.

Untuk melakukan kegiatan ini digunakan perintah,

```
a2n@debian:/home/data/artikel/penelitian/database/data$ createdb akademik
wa2n@debian:/home/data/artikel/penelitian/database/data$ psql akademik < akademik1
```

f. Analisis/pengujian aspek-aspek kualitas schema database
Pada langkah ini dilakukan analisis untuk setiap aspek kualitas schema database yang diperoleh. Analisis dilakukan dengan cara melakukan uji coba langsung pada schema database untuk SIAKAD yang digunakan di ISTA.

g. Penyusunan dokumentasi
Pada langkah ini disusun dokumentasi laporan penelitian sesuai organisasi laporan penelitian yang direncanakan.

4.2. Lingkup Permasalahan
Lingkup permasalahnan yang dianalisis dalam penelitian ini adalah sebagai berikut:
1. Analisis dilakukan pada rancangan schema database yang digunakan pada database akademik di ISTA
2. Analisis dilakukan pada sceham/rancangan logikal database, bukan implementasi secara teknis pada DBMS, teknologi perangkat keras, maupun program aplikasi yang digunakan
3. Analisis dilakukan dengan menggunakan business rule yang digunakan pada ISTA

4.3. Aspek Permasalahan

1. Kebenaran (correctness)
2. Konsistensi (consistency)
3. Relevansi (relevance)
4. Jangkauan (scope)
5. Tingkat detail (level of detail)
6. Kelengkapan (completeness)
7. Minimalitas (minimality)
8. Kemampuan untuk diintegrasikan (*ability of integration*)
9. Kemampuan untuk dibaca (*readability*)

### 4.4. Jadwal Waktu Penelitian

Penelitian ini dilaksanakan dalam jangka waktu 5 (lima) bulan dimulai sejak 06 Februari 2007 hingga 06 Juli 2007, dengan jadwal pelaksanaan penelitian seperti ditampilkan dalam Tabel 2.

<table>
<thead>
<tr>
<th>Kegiatan</th>
<th>Bulan ke</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Studi pendahuluan</td>
<td></td>
</tr>
<tr>
<td>Pengumpulan data</td>
<td></td>
</tr>
<tr>
<td>Analisis aspek-aspek kualitas <em>schema database</em></td>
<td></td>
</tr>
<tr>
<td>Penyusunan dokumentasi</td>
<td></td>
</tr>
</tbody>
</table>

Tabel 4.1: Jadwal pelaksanaan penelitian
BAB V
HASIL DAN PEMBAHASAN

5.1. Hasil
5.1.1. Sistem Informasi di Perguruan Tinggi (SIPT)
5.1.1.1. Sistem Informasi Akademik dalam Peraturan Perundangan
5.1.1.1.1. Sistem Informasi Akademik dalam UU SISDIKNAS


5.1.1.1.2. Sistem Informasi Akademik dalam Peraturan Pemerintah


5.1.1.1.3. Sistem Informasi Akademik dalam KPPTJP

Kerangka Pengembangan Pendidikan Tinggi Jangka Panjang (KPPTJP) 1996-2005 tidak secara eksplisit mencantumkan aturan tentang Sistem Informasi Akademik, namun demikian keterangan yang tercantum
didalamnya memiliki implikasi terhadap Sistem Informasi Akademik pada institusi Pendidikan Tinggi.

Dalam Program Utama nomor A.2.3. tentang Peningkatan Peran Perguruan Tinggi (PT) dalam Perencanaan Pengembangannya dijelaskan bahwa Program Utama Peningkatan Peran PT Dalam Perencanaan Pengembangannya mempunyai tujuan meningkatkan kemampuan untuk:
1. Menetapkan tujuan dan sasaran
2. Mengupayakan dan meningkatkan kelaiakan memperoleh sumberdaya
3. Merencanakan dan melaksanakan proses pencapaian tujuan

Tolok ukur pencapaian tujuan tersebut adalah adanya Unit Kerja Tetap untuk perencanaan pengembangan di masing-masing PT yang ditunjang oleh Sistem Informasi Manajemen yang efektif.

Dalam Program Utama nomor A.3.1 tentang Pengembangan Pengelolaan Sumberdaya Terpadu di PT, dijelaskan bahwa melalui terselenggaranya sistem Penganggaran Belanja Terpadu yang menjadi bagian dari Sistem Informasi Manajemen, Pimpinan PT dapat merencanakan dan melaksanakan manajemen sumberdaya yang efisien dan efektif. Tolok ukur pencapaian tujuan tersebut adalah adanya tata buku aliran sumberdaya terpadu di PT dan adanya manajemen sumberdaya yang efisien dan efektif.

Dalam Program Utama nomor A.4.2 tentang Peningkatan Kesiapan PTN dan PTS Terhadap Fungsi dan Pelaksanaan Fungsi BAN dijelaskan bahwa suatu sistem informasi manajemen yang baik, yang mengandung unsur-unsur pengelolaan sumberdaya terpadu akan membantu kelancaran proses akreditasi oleh BAN terhadap PT yang bersangkutan. Di samping itu hasil evaluasi-diri PT yang bersangkutan dapat digunakan sebagai titik tolak proses akreditasi tersebut. Bagi PT yang bersangkutan, hasil proses akreditasi, menjadi acuan untuk merancang dan merencanakan program-program peningkatan kualitas kinerja. Tolok ukur pencapaian tujuan tersebut adalah tersedianya data, informasi dan
laporan evaluasi-diri di PT yang dapat digunakan sebagai titik tolak proses akreditasi (http://www.dikti.org).

5.1.1.1.4. Sistem Informasi Akademik dalam Naskah Akademik Sistem Akreditasi


Keterangan-keterangan yang dimaksud tercantum pada Bagian B tentang Standar Akreditasi Institusi dan PS. Dalam naskah tersebut dijelaskan bahwa standar akreditasi institusi dan PS terdiri atas (BAN PT Depdiknas, 2001):

1. Standar Akreditasi Institusi Peguruan Tinggi (IPT)
2. Standar Akreditasi PS

Standar yang digunakan untuk menggambarkan komitmen inti kapasitas Institusi PT, meliputi (BAN PT Depdiknas, 2001):

1. Integritas
2. Visi, misi, tujuan dan sasaran
3. Tata pamong (governance)
4. Sumberdaya manusia
5. Sarana dan prasarana
6. Keuangan
7. Sistem informasi
8. Keberlanjutan

Standar yang digunakan untuk menggambarkan komitmen inti terhadap efektivitas pendidikan PT, meliputi (BAN PT Depdiknas, 2001):
9. Mahasiswa
10. Kurikulum
11. Sistem pembelajaran
12. Penelitian dan Pengabdian kepada masyarakat
13. Sistem penjaminan mutu
14. Sistem pengelolaan
15. Suasana akademik
16. PS

Standar tentang Sistem Informasi memiliki enam parameter, yaitu (BAN PT Depdiknas, 2001):

1. Sistem informasi yang telah terlembaga
2. Implementasi rencana pengembangan sistem informasi
3. Kemudahan penggunaan, meliputi akses bagi mahasiswa, akses bagi staf, dan akses dari luar kampus
4. Penggunaan sebagai sarana informasi baik internal (local) maupun eksternal (global)
5. Pelatihan penggunaan IT untuk sivitas akademika dan karyawan
6. Sarana, yang meliputi: kecukupan, yaitu rasio mahasiswa/terminal, rasio dosen/terminal, bandwidth, serta rasio mahasiswa/bandwidth; efisiensi penggunaan; kesesuaian, kemutakhiran

5.1.1.1.5. Sistem Informasi Akademik dalam Portofolio Akreditasi

memberikan penjelasan pada bagian Sistem Informasi dalam penyusunan Portofolio Akreditasi Institusi PT, yaitu memuat uraian mengenai hal-hal berikut:

1. Sistem informasi manajemen untuk mendukung kegiatan akademik dan nonakademik, meliputi:
   a. Pengelolaan sistem informasi
   b. Jenis informasi yang disediakan
   c. Ketersediaan perangkat keras, perangkat lunak dan sumber daya manusia
   d. Keberadaan dan pemanfaatan jaringan lokal (LAN), serta Keberadaan dan pemanfaatan jaringan luas (WAN)

2. Jenis informasi yang disediakan

3. Kemudahan penggunaan (akses) bagi dosen, mahasiswa, orang luar

4. Pemanfaatan informasi oleh pimpinan, dosen, mahasiswa, dan masyarakat

5. Kecukupan sarana informasi, yaitu rasio mahasiswa/terminal, rasio dosen/terminal, bandwidth, serta rasio mahasiswa/bandwidth

6. Kesesuaian dengan kebutuhan

7. Kemutakhiran

8. Pelatihan penggunaan

5.1.1.2. Kebutuhan Sistem Informasi di PT (SIPT)

Kebutuhan SIM di PT (SIPT) dapat berbeda antara PT (PT) satu dengan yang lainnya. Perbedaan ini utamanya dipengaruhi oleh 3 (tiga) hal, yaitu perbedaan aturan bisnis (business rule), perbedaan ukuran organisasi, dan perbedaan area fungsional dalam organisasi PT.

Perbedaan aturan bisnis terjadi karena perbedaan peraturan dan kebijakan yang digunakan, perbedaan ukuran organisasi terjadi karena perbedaan jumlah Fakultas, Jurusan, dan PS yang diselenggarakan, serta jumlah mahasiswa, sedangkan perbedaan area fungsional dalam organisasi terjadi karena perbedaan kompleksitas pengolahan dan struktur

Berdasarkan hasil identifikasi yang dilakukan, maka SIPT dapat tersusun atas sejumlah sub sistem berikut:

1. **Sub Sistem Informasi Akademik (SIAKAD)**
   Sub sistem ini berfungsi untuk mengolah data Penmaru, heregisterasi, perwalian, KRS, perkuliahan, praktikum, Kerja Praktek, Tugas Akhir, KKN, yudisium kelulusan, wisuda, serta alumni

2. **Sub Sistem Informasi Perpustakaan (SIPERPUSTAKAAN)**
   Sub sistem ini berfungsi untuk mengolah data buku, anggota, transaksi peminjaman, transaksi pengembalian, denda keterlambatan, lainnya

3. **Sub Sistem Informasi Keuangan (SIKEUANGAN)**
   Sub sistem ini berfungsi untuk mengolah data anggaran, pemasukan dana (SPP, DPP, lainnya), pengeluaran dana (gaji, pengembangan, dan lainnya), akuntansi, pelaporan, lainnya

4. **Sub Sistem Informasi Kepegawaian (SIPEGAWAI)**
   Sub sistem ini berfungsi untuk mengolah data kepangkatan, pendidikan dan pelatihan, prestasi, lainnya

5. **Sub Sistem Informasi Penggajian (SIGAJI)**
   Sub sistem ini berfungsi untuk mengolah data gaji dosen, staf administrasi, sopir, staf keamanan, lainnya

6. **Sub Sistem Informasi Inventaris (SIINVENTARIS)**
   Sub sistem ini berfungsi untuk mengolah data inventaris, inventori, kendaraan, lainnya

7. **Sub Sistem Informasi Kemahasiswaan (SIKEMAHASISWAAN)**
   Sub sistem ini berfungsi untuk mengolah data kegiatan kemahasiswaan, ekstrakurikuler, beasiswa, prestasi, lainnya
8. Sub Sistem Informasi Kerjasama dan Kehumasan (SIKEHUMAS)
   Sub sistem ini berfungsi untuk mengolah data-data kegiatan kerjasama, kehumasan, lainnya

9. Sub Sistem Informasi Penelitian (SIPENELITIAN)
   Sub sistem ini berfungsi untuk mengolah data-data kegiatan penelitian dosen, prestasi penelitian dosen, lainnya

10. Sub Sistem Informasi Pengabdian Kepada Masyarakat (SIPENGABDIAN)
    Sub sistem ini berfungsi untuk mengolah data-data kegiatan Pengabdian Kepada Masyarakat yang dilakukan oleh dosen, Pengabdian Kepada Masyarakat yang dilakukan oleh dosen, lainnya


   Sekalipun SIPT tersusun atas sejumlah sub sistem, namun sesuai dengan konsep database, maka setiap sub sistem dalam SIPT akan mengakses ke sebuah database yang sama, yaitu database PT. Database PT ini akan melayani berbagai kebutuhan pengolahan data dan penyajian informasi untuk seluruh pengguna pada seluruh level manajemen dan seluruh sub sistem di sebuah PT.

5.1.1.3. Pengguna Sistem Informasi di PT (SIPT)

   Berdasarkan konsep SIM, pengguna dapat berupa orang-orang atau program-program aplikasi. Pada suatu PT, orang-orang yang menggunakan SIPT dapat berupa perseorangan atau sekelompok orang yang berada di dalam maupun di luar struktur organisasi PT. Dalam SIPT, personil pengguna di dalam struktur organisasi PT dapat dikelompokkan menjadi 3 (tiga), yaitu:

   1. Para pengguna pada tingkat manajemen tertinggi (perencanaan strategis), misal Rektor pada Universitas atau Institut, Ketua pada Sekolah Tinggi, Direktur pada Akademi, para Pembantu Rektor (Warek/Purek/PR) pada Universitas atau Institut, para Pembantu Ketua pada Sekolah Tinggi, para Pembantu Direktur pada Akademi,
para Pejabat di tingkat Yayasan pada PT Swasta, dan Pejabat lain yang setingkat
2. Para pengguna pada tingkat manajemen menengah (perencanaan taktis dan pengendalian manajemen), misal para Dekan, para Pembantu Dekan (Pudek/PD), dan Pejabat lain yang setingkat
3. Para pengguna pada tingkat manajemen terendah (perencanaan dan pengendalian operasional), misal para Ketua Jurusan, para Sekretaris Jurusan, para Ketua PS, staf administrasi, dan Pejabat lain yang setingkat.

Sedangkan personil atau institusi pengguna yang berada di luar struktur organisasi PT dapat terdiri atas:
1. Dosen
2. Mahasiswa
3. Alumni
4. Masyarakat pengguna lulusan
5. Pemerintah (DIKTI/DIKNAS/BAN-PT)
6. Penyandang dana (Yayasan pada PT Swasta)
7. Orang tua/wali mahasiswa
8. Masyarakat umum


5.1.1.4. Kebutuhan Informasi dan data untuk SIPT
Berdasarkan hasil identifikasi sub sistem informasi dalam SIPT pada bagian sebelumnya, maka secara garis besar kebutuhan informasi dan data dalam SIPT dapat meliputi:
1. Informasi dan data terinci, baik insidentil maupun rutin, dalam bentuk teks, tabel maupun grafis yang terkait dengan semua sub sistem dalam SIPT
2. Informasi dan data teringkas, baik insidentil maupun rutin, dalam bentuk teks, tabel maupun grafis yang terkait dengan semua sub sistem dalam SIPT
3. Informasi dan data spesifik, baik insidentil maupun rutin, dalam bentuk teks, tabel maupun grafis yang terkait dengan semua sub sistem dalam SIPT

5.1.1.4.1. Kebutuhan Informasi dan data untuk Penyusunan Portofolio

Kebutuhan informasi dan data untuk penyusunan Portofolio Institusi PT dapat diidentifikasikan berdasarkan Pedoman Penyusunan Portofolio Institusi PT. Portofolio adalah dokumen yang menggambarkan proses perkembangan dan rencana pencapaian visi di masa yang akan datang. Dalam hal akreditasi, Portofolio digunakan sebagai instrumen penilaian yang bersifat terbuka (open-ended) dan multipurpose. Portofolio harus disusun berdasarkan evaluasi diri melalui suatu analisis kekuatan, kelemahan, peluang, dan tantangan (SWOT analysis) serta mencakup informasi komprehensif tentang indikator kinerja kunci yang dinilai (dalam hal ini adalah penyelenggaraan, aset fisik, aset finansial, aset sumberdaya manusia, dan aset informasi).

Uraian SWOT analysis portofolio tingkat institusi harus mencakup aspek-aspek berikut (BAN PT Depdiknas, 2001):

1. Visi, misi, dan tujuan PS
2. Rancangan, substansi, implementasi, dan evaluasi kurikulum
3. Sistem instruksional: belajar, mengajar, dan evaluasi hasil belajar
4. Evaluasi kemajuan dan pencapaian hasil belajar
5. Bimbingan dan konseling untuk mahasiswa
6. Sumberdaya pendukung Tridharma PT
7. Suasana akademis
8. Penelitian, publikasi, pengabdian kepada masyarakat
9. Pengelolaan penyelenggaraan institusi, serta pemenuhan dan peningkatan kualitas

Dengan demikian, bahan penilaian akreditasi tingkat institusi adalah intisari hasil SWOT analysis dan portofolio.

5.1.1.4.2. Kebutuhan Informasi dan data untuk Peningkatan Mutu

Perkembangan teknologi informasi dan komunikasi saat ini memicu globalisasi di berbagai aspek kehidupan. Globalisasi ekonomi memacu PT untuk melakukan internasionalisasi, kerjasama, serta kompetisi dalam berbagai aspek pendidikan. Agar dapat berhasil dalam kompetisi, maka mutu harus merupakan fokus pengembangan. Aspek mutu dalam paradigma baru PT meliputi:

1. Otonomi, yaitu mandiri di berbagai aspek
2. Akuntabilitas, yaitu terukur dengan indikator
3. Akreditasi, yaitu dapat disetarakan
4. Evaluasi, yaitu selalu melakukan evaluasi diri

Pendidikan Tinggi dikatakan bermutu jika mampu menghasilkan lulusan yang bermutu, yaitu (Suwardi, 2003):

1. Cepat lulus
2. Lulus dengan nilai tinggi
3. Cepat bekerja
4. Dapat diterima di banyak bidang
5. Gaji tinggi

Peningkatan Mutu Pendidikan Tinggi dilakukan dengan RAISE++, yaitu (BAN PT Depdiknas, 2001):

1. Relevance
2. Academic Atmosphere
3. Internal management and Organization
4. Sustainability
5. *Efficiency and Productivity*

6. + *Access Equity*

7. + *Leadership*

Mutu lulusan Pendidikan Tinggi dapat diukur dengan indikator kinerja. Penyusunan indikator kinerja dapat dilakukan dengan Evaluasi Diri. Evaluasi Diri dapat disusun dengan baik jika tersedia data yang lengkap dan akurat. Data-data dasar yang perlu dianalisis sebelum penyusunan laporan Evaluasi Diri untuk keperluan peningkatan mutu Pendidikan Tinggi meliputi *(BAN PT Depdiknas, 2001):*

1. IPK Lulusan
2. Waktu Tunggu Lulusan
3. Lama Studi Mahasiswa
4. Lama Skripsi Mahasiswa
5. Status Mahasiswa
6. Potret IPK Mahasiswa
7. *English Proficiency Test* Mahasiswa
8. Proses Pendidikan
9. Keketatan, *Score* dan NEM Mahasiswa Baru
10. Asal Propinsi Mahasiswa Baru
11. Asal Kodya/Kabupaten Mahasiswa Baru
12. Asal Pendaftar/Pemilih
13. Kerjasama
14. Pendidikan dan Umur Dosen
15. Dosen *Full/Part Time* dan Studi
16. SKS Dosen
17. Pendidikan dan Umur Staf Administrasi
18. Luas Gedung/Ruang
19. Kapasitas Ruang Kuliah
20. Laboratorium dan pemanfaatnya
21. Umur Koleksi di Perpustakaan
22. Koleksi untuk Mata Kuliah di Perpustakaan
5.1.1.4.3. Kebutuhan Informasi dan data untuk Evaluasi Diri

Kebutuhan informasi dan data untuk penyusunan Evaluasi Diri dapat diidentifikasi berdasarkan Pedoman Evaluasi Diri PS (BAN PT Depdiknas, 2001). Evaluasi Diri adalah penilaian, telaah, dan analisis keseluruhan sistem PS/lembaga PT, yang mencakup masukan, proses, dan keluaran berdasarkan data, informasi dan bukti-bukti lainnya yang berkaitan dengan komponen-komponen sistemik dari penyelenggaraan PS.

Berdasarkan analisis tersebut, maka dapat dijabarkan dimensi penilaian yang digunakan dalam akreditasi PS yang secara garis besar terdiri atas komponen-komponen berikut (BAN PT Depdiknas, 2001):

A. Masukan, mencakup komponen berikut:
   1. Visi dan misi PS
   2. Sasaran dan tujuan
   3. Mahasiswa
   4. Dosen dan tenaga pendukung
   5. Kurikulum
   6. Sarana dan prasarana
   7. Biaya dan sumber dana (pendanaan)

B. Proses, mencakup komponen berikut:
   8. Tata pamong (governance)
   9. Pengelolaan program
   10. Proses pembelajaran
   11. Suasana Akademik
   12. Penelitian dan skripsi
   13. Pengabdian kepada masyarakat

C. Keluaran/Hasil, mencakup komponen berikut:
14. Lulusan
15. Keluaran lainnya: publikasi hasil penelitian dan atau produk penelitian dalam bentuk paten, rancang bangun, prototipe, perangkat lunak, lainnya

D. Balikan dan tindakan lanjut, mencakup komponen berikut:
16. Sistem informasi
17. Sistem peningkatan dan pengendalian mutu

5.1.1.4.4. Kebutuhan Informasi dan data untuk Borang Akreditasi


<table>
<thead>
<tr>
<th>Tabel 5.1: Kebutuhan informasi dan data untuk penyusunan borang</th>
<th>Butir No. Kolom</th>
<th>Kebutuhan Informasi dan data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identitas PS dan pengisi borang</td>
<td>(1) - (4)</td>
<td>Nomor mata kuliah, nama mata kuliah, kelompok mata kuliah</td>
</tr>
<tr>
<td></td>
<td>(5) - (12)</td>
<td>Status mata kuliah: (wajib, pilihan; KURNAS, lokal; beban sks perkuliahan, praktikum, total)</td>
</tr>
<tr>
<td></td>
<td>(13)</td>
<td>Nama jurusan/fakultas/universitas pembina setiap mata kuliah yang digunakan oleh PS</td>
</tr>
<tr>
<td></td>
<td>(14)</td>
<td>Nama dosen penanggung jawab dan keahlian dosen yang menangani mata kuliah</td>
</tr>
<tr>
<td>Lampiran 2</td>
<td></td>
<td>Nomor, semester, kode mata kuliah, dan nama mata kuliah pilihan yang diselenggarakan dalam tiga tahun terakhir, bobot sks setiap mata kuliah, jurusan/fakultas yang membina mata kuliah, dan nama dosen yang mengajar</td>
</tr>
<tr>
<td>Lampiran 3</td>
<td></td>
<td>Nama mata kuliah praktikum dan atau praktek, pokok masalah dan jumlah jam praktikum per semester untuk setiap judul praktikum, serta tempat pelaksanaan praktek/praktikum</td>
</tr>
<tr>
<td>2a</td>
<td>(1) - (5)</td>
<td>Nama lengkap dosen tetap (tidak disingkat, gelar di belakang nama) berserta tempat dan tanggal lahir, tuliskan nomor identitas dosen, pendidikan S1, S2 dan S3 dan asal universitas, jabatan fungsional akademik</td>
</tr>
<tr>
<td></td>
<td>(7)</td>
<td>Bidang keahlian dosen berdasarkan SK jabatan atau pendidikan tertinggi</td>
</tr>
<tr>
<td>(8)</td>
<td>Bidang mata kuliah yang diajarkan dosen</td>
<td></td>
</tr>
<tr>
<td>(9)</td>
<td>Jumlah sks yang diajarkan dosen pada mahasiswa tahun pertama</td>
<td></td>
</tr>
<tr>
<td>(10) - (11)</td>
<td>Jumlah sks yang diajarkan dosen per semester pada PS yang bersangkutan pada semester ganjil dan genap untuk 3 tahun terakhir</td>
<td></td>
</tr>
</tbody>
</table>

2b
| (1) - (5) | Nama lengkap dosen tidak tetap (tidak disingkat, gelar di belakang nama) berserta tempat dan tanggal lahir, nomor identitas dosen, pendidikan S1, S2 dan S3 dan asal universitas, jabatan fungsional akademik |
| (7) | Bidang keahlian dosen berdasarkan SK jabatan atau pendidikan tertinggi |
| (8) | bidang mata kuliah yang diajarkan dosen |
| (9) | Jumlah sks yang diajarkan dosen pada mahasiswa tahun pertama |
| (10) | Jumlah rata-rata sks yang diajarkan dosen per semester untuk 5 tahun terakhir |

3 (1) - (6) Jumlah karyawan menurut kualifikasinya
| (7) | Unit kerja adalah jenis tenaga penunjang yang mempunyai akses pada PS yang bersangkutan baik di Fakultas dan PS |

4 Suasana akademis yang kondusif adalah iklim yang mendorong interaksi positif antara dosen dan dosen, dosen-mahasiswa, serta mahasiswa-mahasiswa. Jenis dan mekanisme pelaksanaan upaya untuk mewujudkan suasana akademis (dalam lingkup pendidikan pengajaran, penelitian, dan pengabdian pada masyarakat) |

5a (1) TS adalah tahun akademik utuh terakhir sebelum saat pengisian Borang |
| (2) | Jumlah mahasiswa yang bermimat/mendaftar pada setiap tahun pendaftaran |
| (3) | Daya tampung nyata PS sesuai dengan kapasitas fasilitas yang ada untuk menerima mahasiswa baru untuk setiap tahun pendaftaran |
| (4) | Jumlah mahasiswa yang lulus seleksi dan diterima menjadi peserta PS pada setiap tahun pendaftaran |
| (5) | Jumlah mahasiswa yang telah diterima yang benar-benar melakukan registrasi (mendaftar) untuk mengikuti kegiatan perkuliahan |
| (6) | Jumlah lulusan tahun yang bersangkutan |
| (7)- (9) | IPK minimal yang dicapai mahasiswa yang lulus pada tahun yang bersangkutan |
| | IPK rata-rata yang dicapai mahasiswa yang lulus pada tahun yang bersangkutan |
| | IPK maksimal yang dicapai mahasiswa yang lulus pada tahun yang bersangkutan |
| (10) - (12) | Jumlah lulusan pada tiap tahun kelulusan yang memperoleh IPK < 2,75 |
| | Jumlah lulusan pada tiap tahun kelulusan yang memperoleh IPK antara 2,75 - 3,50 |
| | Jumlah lulusan pada setiap tahun kelulusan yang memperoleh IPK > 3,50 |

5b (2) Jumlah mahasiswa yang mendaftar pertama kali pada tahun TS-8
<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Jumlah mahasiswa yang mendaftar pertama kali pada tahun: TS-8 yang masih teregistrasi pada tahun TS-7 TS-7 yang masih teregistrasi pada tahun TS-6 TS-6 yang masih teregistrasi pada tahun TS-5 TS-5 yang masih teregistrasi pada tahun TS-4 TS-4 yang masih teregistrasi pada tahun TS-3 TS-3 yang masih teregistrasi pada tahun TS-2 TS-2 yang masih teregistrasi pada tahun TS-1 TS-1 yang masih teregistrasi pada tahun TS-0 TS-0 yang masih teregistrasi pada tahun TS-0</td>
</tr>
<tr>
<td>4</td>
<td>Jumlah mahasiswa yang mendaftar pertama kali pada tahun: TS-8 yang masih teregistrasi pada tahun TS-6 TS-7 yang masih teregistrasi pada tahun TS-5 TS-6 yang masih teregistrasi pada tahun TS-5 TS-5 yang masih teregistrasi pada tahun TS-4 TS-4 yang masih teregistrasi pada tahun TS-4 TS-3 yang masih teregistrasi pada tahun TS-3 TS-2 yang masih teregistrasi pada tahun TS-3 TS-1 yang masih teregistrasi pada tahun TS-2 TS-0 yang masih teregistrasi pada tahun TS-2</td>
</tr>
<tr>
<td>5</td>
<td>Jumlah mahasiswa yang mendaftar pertama kali pada tahun: TS-8 yang masih teregistrasi pada tahun TS-5 TS-7 yang masih teregistrasi pada tahun TS-5 TS-6 yang masih teregistrasi pada tahun TS-5 TS-5 yang masih teregistrasi pada tahun TS-4 TS-4 yang masih teregistrasi pada tahun TS-4 TS-3 yang masih teregistrasi pada tahun TS-3 TS-2 yang masih teregistrasi pada tahun TS-3 TS-1 yang masih teregistrasi pada tahun TS-2 TS-0 yang masih teregistrasi pada tahun TS-1</td>
</tr>
<tr>
<td>6</td>
<td>TS-8 yang masih teregistrasi pada tahun TS-4 TS-7 yang masih teregistrasi pada tahun TS-4 TS-6 yang masih teregistrasi pada tahun TS-4 TS-5 yang masih teregistrasi pada tahun TS-4 TS-4 yang masih teregistrasi pada tahun TS-3 TS-3 yang masih teregistrasi pada tahun TS-3 TS-2 yang masih teregistrasi pada tahun TS-2 TS-1 yang masih teregistrasi pada tahun TS-1 TS-0 yang masih teregistrasi pada tahun TS-1</td>
</tr>
<tr>
<td>7</td>
<td>TS-8 yang masih teregistrasi pada tahun TS-3 TS-7 yang masih teregistrasi pada tahun TS-3 TS-6 yang masih teregistrasi pada tahun TS-3 TS-5 yang masih teregistrasi pada tahun TS-3 TS-4 yang masih teregistrasi pada tahun TS-3 TS-3 yang masih teregistrasi pada tahun TS-2 TS-2 yang masih teregistrasi pada tahun TS-2 TS-1 yang masih teregistrasi pada tahun TS-1 TS-0 yang masih teregistrasi pada tahun TS-1</td>
</tr>
<tr>
<td>9</td>
<td>TS-8 yang masih teregistrasi pada tahun TS-1 TS-7 yang masih teregistrasi pada tahun TS-1 TS-6 yang masih teregistrasi pada tahun TS-1 TS-5 yang masih teregistrasi pada tahun TS-1 TS-4 yang masih teregistrasi pada tahun TS-1 TS-3 yang masih teregistrasi pada tahun TS-1 TS-2 yang masih teregistrasi pada tahun TS-1 TS-1 yang masih teregistrasi pada tahun TS-1 TS-0 yang masih teregistrasi pada tahun TS-1</td>
</tr>
<tr>
<td>10</td>
<td>TS-8 yang masih teregistrasi pada tahun TS-0 TS-7 yang masih teregistrasi pada tahun TS-0 TS-6 yang masih teregistrasi pada tahun TS-0 TS-5 yang masih teregistrasi pada tahun TS-0 TS-4 yang masih teregistrasi pada tahun TS-0 TS-3 yang masih teregistrasi pada tahun TS-0 TS-2 yang masih teregistrasi pada tahun TS-0 TS-1 yang masih teregistrasi pada tahun TS-0 TS-0 yang masih teregistrasi pada tahun TS-0</td>
</tr>
<tr>
<td>11</td>
<td>Jumlah lulusan total dari mahasiswa untuk setiap angkatan berdasarkan tahun masuk sampai tahun TS</td>
</tr>
</tbody>
</table>

(1) - (2) Jenis fasilitas/peralatan utama yang digunakan dalam proses perkuliahan yang terkait dengan mata kuliah MKK, tidak termasuk ATK (Alat Tulis Kantor) habis pakai berdasarkan
kelompok jenis fasilitasnya (Prasarana atau Sarana). Prasarana adalah fasilitas yang berupa asset infrastruktur (tidak bergerak) seperti tanah, gedung, ruang perkuliahan, ruang laboratorium, ladang/lahan kebun percobaan, dll. Sarana adalah fasilitas/peralatan (bergerak) yang digunakan dalam proses pembelajaran seperti komputer, alat-alat laboratorium, media belajar, mesin-mesin, dll.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(3)</td>
<td>Rasio masing-masing prasarana/sarana dengan mahasiswa pemakai</td>
</tr>
<tr>
<td>(4)</td>
<td>Kondisi kelayakan penggunaan (rusak, tidak rusak)</td>
</tr>
<tr>
<td>(5) - (6)</td>
<td>Kepemilikan fasilitas (milik sendiri/disewa)</td>
</tr>
<tr>
<td>(7)</td>
<td>Jumlah jam penggunaan fasilitas rata-rata per minggu dalam satu semester, oleh bidang studi yang bersangkutan yang terkait dengan pelaksanaan Tridharma PT.</td>
</tr>
</tbody>
</table>

| 7 | (2) - (3) Jumlah dan luas ruang kerja bagi dosen tetap |
|   | (4) - (5) Jumlah dan luas ruang kerja bagi dosen tidak tetap/luar biasa |

| 8a | (1) - (2) Rekapan jumlah ketersediaan pustaka yang relevan |
|   | Lampiran Judul, penulis/penerbit, dan tahun penerbitan pustaka, serta jumlah eksemplar/copy per judul yang dimiliki oleh perpustakaan di lingkungan kampus |

| 8b | Nama perpustakaan di luar PT yang dapat diakses |

| 9a | (1) - (3) Daftar nama Mata kuliah kelompok MKK, MKDK, dan MKP dan bobot sks |
|   | (4) - (10) Jumlah jam rata-rata per minggu yang dialokasikan untuk masing-masing kegiatan untuk setiap mata kuliah |
|   | (11) - (12) SAP dan modul |

| 9b-e | Mekanisme monitoring, penelaahan, dan evaluasi untuk melihat apakah kegiatan-kegiatan yang telah direncanakan pada butir 9a dilaksanakan atau tidak |

| 10a | Jenis kegiatan dalam lingkup pendidikan dan penelitian dan pengabdian masyarakat yang melibatkan dosen dan mahasiswa di luar kegiatan kuliah selama 3 tahun terakhir yang memungkinkan dosen dan mahasiswa berinteraksi |

| 10b | Kontribusi kegiatan pada butir 10.a. terhadap peningkatan kualitas proses pembelajaran |

| 11a | Panduan |

| 11b | Jumlah rata-rata mahasiswa per dosen PA |

| 11c | Jumlah pertemuan rata-rata pembimbingan per mahasiswa per dosen per semester |

| 12a1 | (1) - (4) Nama dosen tamu/tenaga ahli, tugas dan lama tugas dosen tamu, serta tugas dan lama tugas tenaga ahli |

| 12a2 | (1) - (6) Nama dosen yang diberi tugas belajar, tempat tugas belajar (nama institusi dan negara), jenjang studi, bidang studi, dan tahun mulai tugas belajar |

| 12a3 | (1) - (5) Nama dosen yang mengikuti kegiatan, tempat, dan waktu kegiatan, sebagai penyaji atau sebagai peserta |

<p>| 12b | Tujuan, kriteria keberhasilan dan kemanfaatan upaya dalam |</p>
<table>
<thead>
<tr>
<th>No.</th>
<th>Butir</th>
<th>Deskripsi</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>(1) - (6)</td>
<td>Jenis evaluasi kesiapan awal kuliah adalah jenis penilaian yang bertujuan untuk menentukan kemampuan awal mahasiswa pada awal perkuliahan</td>
</tr>
<tr>
<td>14</td>
<td>(1) - (4)</td>
<td>Jumlah dosen tetap yang mengikuti kegiatan</td>
</tr>
<tr>
<td>15</td>
<td>(1) - (4)</td>
<td>Rata-rata beban kerja per semester dosen di PT sendiri dan di luar PT, dalam satuan sks dosen (berdasarkan SK Dirjen DIKTI No. 48 Th. 1983)</td>
</tr>
<tr>
<td>16</td>
<td>(1)</td>
<td>Kode etik akademik umum adalah suatu pedoman yang memberikan rambu-rambu etika ilmiah dalam pelaksanaan Tridharma PT, seperti rambu-rambu <em>information sharing</em>, komitmen terhadap pelaksanaan tugas, akuntabilitas pelaksanaan tugas, dll. Plagiat adalah pengakuan hasil kerja/karya ilmiah orang lain sebagai hasil karya sendiri. Misalnya, mencuplik konsep, gagasan, hubungan kesejawatan, pemikiran mengenai suatu topik secara 100% sama (persis) tanpa menyebutkan sumber aslinya.</td>
</tr>
<tr>
<td>17 a</td>
<td>(2) - (3)</td>
<td>Kepemilikan pedoman yang tertera</td>
</tr>
<tr>
<td>17 b</td>
<td>(2) - (4)</td>
<td>Metode sosialisasi pedoman yang ada, cara penanganan kasus pelanggaran</td>
</tr>
<tr>
<td>18 a</td>
<td>(2) - (4)</td>
<td>Jumlah karya ilmiah dosen tetap yang sesuai dengan bidang 3 tahun terakhir</td>
</tr>
<tr>
<td>18 b</td>
<td>(5) - (7)</td>
<td>Jumlah karya ilmiah dosen tetap yang tidak sesuai dengan bidang 3 tahun terakhir</td>
</tr>
<tr>
<td>18 c</td>
<td>(2) - (4)</td>
<td>Jumlah penelitian dosen tetap yang sesuai dalam 3 tahun terakhir</td>
</tr>
<tr>
<td>18 d</td>
<td>(5) - (7)</td>
<td>Jumlah penelitian dosen tetap yang tidak sesuai</td>
</tr>
<tr>
<td>18 e</td>
<td>(2) - (4)</td>
<td>Jumlah penelitian dosen tetap yang sesuai dalam 3 tahun terakhir</td>
</tr>
<tr>
<td>18 f</td>
<td>(5) - (7)</td>
<td>Jumlah penelitian dosen tetap yang tidak sesuai</td>
</tr>
<tr>
<td>19 a</td>
<td>(2)</td>
<td>Metode pelaksanaan kegiatan yang digunakan, serta informasi terkait. Contoh profil akademis lulusan, kuesioner pelacakan alumni, bukti pertemuan dengan calon pengguna lulusan, dll</td>
</tr>
<tr>
<td>19 b</td>
<td></td>
<td>Upaya pemanfaatan umpan balik bagi peningkatan PS</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>Kerjasama/kemitraan adalah bentuk kerjasama antar-institusi baik di dalam maupun di luar negeri dalam pelaksanaan aspek-aspek Tridharma PT, misal penelitian bersama, tukar menukar dosen dan mahasiswa, penyelenggaraan seminar bersama, dll</td>
</tr>
</tbody>
</table>
5.1.2. Sub Sistem Sistem Informasi Akademik (SIAKAD) dalam SIPT

5.1.2.1. Ruang Lingkup SIAKAD

Dalam penelitian ini, SIAKAD dimaksudkan sebagai sistem yang berfungsi menerima data masukan (input), mengolahnya, dan kemudian menyajikan informasi yang relevan dengan bidang tugas dan tanggung jawab pejabat pada Sub Bidang Akademik. Sedangkan berbagai kegiatan yang menyangkut keuangan, misal pembayaran SPP, DPP, dan lainnya merupakan beban tugas dan tanggungjawab yang ditangani oleh Sub Sistem Informasi Keuangan (SIKEUANGAN). SIAKAD dapat memiliki sejumlah unit fungsional yang merupakan sub sistem dalam SIAKAD, dan SIAKAD sendiri merupakan sub sistem dari SIPT.

Salah satu cara untuk menentukan lingkup SIAKAD pada institusi PT adalah didasarkan pada beban tugas dan tanggungjawab yang diemban oleh Pejabat Pimpinan pada Sub Bidang Akademik pada PT. Pada tingkat tertinggi, Pejabat Pimpinan pada Sub Bidang Akademik adalah Wakil/Pembantu Rektor (Warek/Purek/PR) Bidang Akademik pada Universitas atau Institut, Pembantu Ketua (Puket) Bidang Akademik pada Sekolah Tinggi. Pada beberapa PT, Warek/Purek/PR, atau Puket tersebut juga dibantu oleh pejabat pada tingkat yang lebih rendah, misal Pembantu Dekan (Pudek/PD) di tingkat Fakultas. Dan secara struktural, Sub Bidang Akademik juga dapat dibantu oleh beberapa Unit Kerja pendukung, misal Direktorat, Subdirektorat, Biro, Bagian, BiroAdministrasi Akademik (BAA), Biro Kemahasiswaan dan Alumni, atau lainnya.

Berdasarkan identifikasi beban tugas dan tanggungjawab Sub Bidang Akademik, selanjutnya dapat diidentifikasikan proses atau alur kegiatan dalam SIAKAD. Secara umum SIAKAD terdiri atas sejumlah sub sistem, yaitu:

1. Sub Sistem Informasi Penerimaan Mahasiswa Baru (SIPENMARU)
2. Sub Sistem Informasi Heregisterasi (SIHEREGISTERASI)
3. Sub Sistem Informasi Perwalian, Pra-KRS, KRS, Perubahan KRS (SIKRS)
4. Sub Sistem Informasi Perkuliahan (SIKULIAH)
5. Sub Sistem Informasi Praktikum (SIPRAKTIKUM)
6. Sub Sistem Informasi Kerja Praktek (SIKP)
7. Sub Sistem Informasi Tugas Akhir (SITA)
8. Sub Sistem Informasi KKN (SIKKN)
9. Sub Sistem Informasi Yudisium Kelulusan (SIYUDISIUM)
10. Sub Sistem Informasi Wisuda (SIWISUDA)
11. Sub Sistem Informasi Alumni (SIALUMNI)

5.1.2.2. Pengguna dalam SIAKAD

Berdasarkan konsep SIM, pengguna dapat berupa orang-orang atau program-program aplikasi. Pada SIAKAD, pengguna dapat berupa perseorangan atau sekelompok orang yang berada di dalam maupun di luar struktur organisasi. Dalam SIAKAD, personil pengguna di dalam struktur organisasi PT dapat dikelompokkan menjadi 3 (tiga), yaitu:

1. Para pengguna pada tingkat manajemen tertinggi (perencanaan strategis), misal Rektor pada Universitas atau Institut, Ketua pada Sekolah Tinggi, Direktur pada Akademi, para Pembantu Rektor (Warek/Purek/PR) pada Universitas atau Institut, para Pembantu Ketua pada Sekolah Tinggi, para Pembantu Direktur pada Akademi, para Pejabat di tingkat Yayasan pada PT Swasta, dan Pejabat lain yang setingkat

2. Para pengguna pada tingkat manajemen menengah (perencanaan taktis dan pengendalian manajemen), misal para Dekan, para Pembantu Dekan (Pudek/PD), dan Pejabat lain yang setingkat

3. Para pengguna pada tingkat manajemen terendah (perencanaan dan pengendalian operasional), misal para Ketua Jurusan, para Sekretaris Jurusan, para Ketua PS, staf administrasi, dan Pejabat lain yang setingkat.
Sedangkan personil atau institusi pengguna yang berada di luar struktur organisasi PT dapat terdiri atas:

1. Dosen
2. Mahasiswa
3. Alumni
4. Masyarakat pengguna lulusan
5. Pemerintah (DIKTI/DIKNAS/BAN-PT)
6. Penyandang dana (Yayasan pada PT Swasta)
7. Orang tua/wali mahasiswa
8. Masyarakat umum


5.1.2.3. Kebutuhan Informasi dan Data pada SIAKAD

Sebagaimana terjadi dalam SIPT, Sub Sistem Informasi Akademik (SIAKAD) sebagai bagian dari SIPT juga dapat berbeda di antara PT satu dengan yang lainnya. Kondisi seperti ini menunjukkan bahwa fungsi, tujuan, lingkup, definisi, serta implementasi SIAKAD pada masing-masing institusi PT tidaklah seragam. SIAKAD sebagai bagian dari SIPT dapat diintegrasikan dengan sub sistem lain dalam SIPT ataupun berupa sebuah aplikasi yang bersifat otonom.

Sebagian besar SIAKAD menangani tugas-tugas yang terkait dengan proses perkuliahan saja. Sebagian yang lain, SIAKAD juga menangani tugas-tugas lainnya, misal Penmaru, heregisterasi, perkuliahan, evaluasi, hingga alumni.
5.1.2.4. Kebutuhan Informasi dan Data untuk Setiap Sub Sistem SIAKAD


Berdasarkan hasil identifikasi kebutuhan informasi dan data untuk setiap Sub Sistem Informasi dalam SIAKAD, maka selanjutnya dapat diidentifikasi kebutuhan rinci data untuk setiap Sub Sistem Informasi dalam SIAKAD. Identifikasi kebutuhan informasi dan data dilakukan dengan asumsi bahwa SIAKAD adalah sub sistem dari SIPT yang merupakan sistem informasi berbasis komputer (Computer Based Information Systems/CBIS).

5.1.2.4.1. Sub Sistem Informasi Penmaru (SIPENMARU)

Sub Sistem Informasi Penerimaan Mahasiswa Baru (SIPENMARU) ini dimaksudkan untuk menangani kegiatan Penerimaan Mahasiswa Baru. Kebutuhan informasi dan data dan nilai rinci data untuk SIPENMARU dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan Portofolio, peningkatan mutu, evaluasi diri, borang, perencanaan strategis, perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIPENMARU adalah sebagai berikut:

1. Daftar pembeli formulir pendaftaran keseluruhan
2. Daftar pembeli formulir pendaftaran berdasarkan tanggal
3. Daftar pembeli formulir yang tidak mendaftar keseluruhan
4. Daftar pembeli formulir yang tidak mendaftar berdasarkan PS
5. Daftar pembeli formulir yang tidak mendaftar berdasarkan jalur pendaftaran
6. Daftar pembeli formulir yang tidak mendaftar berdasarkan tanggal pendaftaran
7. Daftar pembeli formulir yang tidak mendaftar berdasarkan gelombang pendaftaran
8. Daftar rincian nilai ujian pendaftar keseluruhan
9. Daftar nilai ujian pendaftar berdasarkan mata ujian
10. Rekapitulasi pembeli formulir berdasarkan PS
11. Rekapitulasi pembeli formulir berdasarkan jalur pendaftaran
12. Rekapitulasi pembeli formulir berdasarkan gelombang pendaftaran
13. Rekapitulasi jumlah pembeli formulir berdasarkan asal Propinsi
14. Rekapitulasi jumlah pembeli formulir berdasarkan asal Kabupaten/Kodya
15. Rekapitulasi jumlah pembeli formulir berdasarkan jalur pendaftaran
16. Rekapitulasi jumlah pembeli formulir berdasarkan jenis kelamin
17. Rekapitulasi jumlah pembeli formulir berdasarkan usia
18. Rekapitulasi jumlah pembeli formulir berdasarkan NEM
19. Rekapitulasi jumlah pembeli formulir berdasarkan pekerjaan ortu
20. Rekapitulasi jumlah pembeli formulir berdasarkan penghasilan ortu
21. Daftar pendaftar keseluruhan
22. Daftar pendaftar berdasarkan PS
23. Daftar pendaftar berdasarkan jalur pendaftaran
24. Daftar pendaftar berdasarkan tanggal pendaftaran
25. Daftar pendaftar berdasarkan gelombang pendaftaran
26. Rekapitulasi pendaftar berdasarkan PS
27. Rekapitulasi pendaftar berdasarkan jalur pendaftaran
28. Rekapitulasi pendaftar berdasarkan gelombang pendaftaran
29. Rekapitulasi jumlah pendaftar berdasarkan asal Propinsi
30. Rekapitulasi jumlah pendaftar berdasarkan asal Kabupaten/Kodya
31. Rekapitulasi jumlah pendaftar berdasarkan jalur pendaftaran
32. Rekapitulasi jumlah pendaftar berdasarkan jenis kelamin
33. Rekapitulasi jumlah pendaftar berdasarkan usia
34. Rekapitulasi jumlah pendaftar berdasarkan NEM
35. Rekapitulasi jumlah pendaftar berdasarkan nilai ujian Bahasa Inggris
36. Rekapitulasi jumlah pendaftar berdasarkan rata-rata nilai ujian
37. Rekapitulasi jumlah pendaftar berdasarkan pekerjaan ortu
38. Rekapitulasi jumlah pendaftar berdasarkan penghasilan ortu
39. Daftar pendaftar yang tidak hadir ujian masuk keseluruhan
40. Daftar pendaftar yang tidak hadir ujian masuk berdasarkan PS
41. Daftar pendaftar yang tidak hadir ujian masuk berdasarkan gelombang pendaftaran
42. Daftar pendaftar diterima keseluruhan
43. Daftar pendaftar diterima berdasarkan PS
44. Daftar pendaftar diterima berdasarkan jalur pendaftaran
45. Daftar pendaftar diterima berdasarkan gelombang pendaftaran
46. Rekapitulasi jumlah pendaftar diterima berdasarkan PS
47. Rekapitulasi jumlah pendaftar diterima berdasarkan jalur pendaftaran
48. Rekapitulasi jumlah pendaftar diterima berdasarkan gelombang pendaftaran
49. Rekapitulasi jumlah pendaftar diterima berdasarkan asal Propinsi
50. Rekapitulasi jumlah pendaftar diterima berdasarkan asal Kabupaten/Kodya
51. Rekapitulasi jumlah pendaftar diterima berdasarkan jalur pendaftaran
52. Rekapitulasi jumlah pendaftar diterima berdasarkan jenis kelamin
53. Rekapitulasi jumlah pendaftar diterima berdasarkan usia
54. Rekapitulasi jumlah pendaftar diterima berdasarkan NEM
55. Rekapitulasi jumlah pendaftar diterima berdasarkan nilai ujian Bahasa Inggris
56. Rekapitulasi jumlah pendaftar diterima berdasarkan rata-rata nilai ujian
57. Rekapitulasi jumlah pendaftar diterima berdasarkan pekerjaan ortu
58. Rekapitulasi jumlah pendaftar diterima berdasarkan penghasilan ortu
59. Daftar pendaftar diterima yang tidak heregisterasi keseluruhan
60. Daftar pendaftar diterima yang tidak heregisterasi berdasarkan PS
61. Daftar pendaftar diterima yang tidak heregisterasi berdasarkan gelombang pendaftaran
62. Daftar mahasiswa baru keseluruhan
63. Daftar mahasiswa baru berdasarkan PS
64. Daftar mahasiswa baru berdasarkan jalur pendaftaran
65. Daftar mahasiswa baru berdasarkan gelombang pendaftaran
66. Rekapitulasi mahasiswa baru berdasarkan PS
67. Rekapitulasi mahasiswa baru berdasarkan jalur pendaftaran
68. Rekapitulasi mahasiswa baru berdasarkan gelombang pendaftaran
69. Rekapitulasi mahasiswa baru berdasarkan asal Propinsi
70. Rekapitulasi mahasiswa baru berdasarkan asal Kabupaten/Kodya
71. Rekapitulasi mahasiswa baru berdasarkan jalur pendaftaran
72. Rekapitulasi mahasiswa baru berdasarkan jenis kelamin
73. Rekapitulasi mahasiswa baru berdasarkan usia
74. Rekapitulasi mahasiswa baru berdasarkan NEM
75. Rekapitulasi mahasiswa baru berdasarkan nilai ujian Bahasa Inggris
76. Rekapitulasi mahasiswa baru berdasarkan rata-rata nilai ujian
77. Rekapitulasi mahasiswa baru berdasarkan pekerjaan ortu
78. Rekapitulasi mahasiswa baru berdasarkan penghasilan ortu
79. Tingkat keketatan persaingan pendaftaran
80. Tingkat keberhasilan Penmaru
5.1.2.4.2. **Sub Sistem Informasi Heregisterasi (SIHEREGISTERASI)**

Sub Sistem Informasi Heregisterasi (SIHEREGISTERASI) ini dimaksudkan untuk menangani kegiatan heregisterasi khusus untuk mahasiswa lama (selain mahasiswa baru). Kebutuhan informasi dan data dan nilai rinci data untuk SIHEREGISTERASI dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan Portofolio, peningkatan mutu, evaluasi diri, borang, perencanaan strategis, perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIHEREGISTERASI adalah sebagai berikut:

1. Daftar mahasiswa heregisterasi keseluruhan
2. Daftar mahasiswa heregisterasi berdasarkan PS
3. Daftar mahasiswa heregisterasi berdasarkan tahun angkatan
4. Daftar mahasiswa heregisterasi berdasarkan tanggal
5. Daftar mahasiswa tidak heregisterasi keseluruhan
6. Daftar mahasiswa tidak heregisterasi berdasarkan PS
7. Daftar mahasiswa tidak heregisterasi berdasarkan tahun angkatan
8. Daftar mahasiswa cuti keseluruhan
9. Daftar mahasiswa cuti berdasarkan PS
10. Daftar mahasiswa cuti berdasarkan tahun angkatan
11. Rekapitulasi mahasiswa heregisterasi berdasarkan PS
12. Rekapitulasi mahasiswa heregisterasi berdasarkan tahun angkatan
13. Rekapitulasi mahasiswa heregisterasi berdasarkan jenis kelamin
14. Tingkat keberhasilan mahasiswa heregisterasi

5.1.2.4.3. **Sub Sistem Informasi KRS (SIKRS)**

Sub Sistem Informasi KRS (SIKRS) ini dimaksudkan untuk menangani kegiatan Perwalian, Pra-KRS, KRS, dan perubahan KRS untuk seluruh mahasiswa. Kebutuhan informasi dan data dan nilai rinci data untuk SIKRS dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIKRS adalah sebagai berikut:

1. Daftar Dosen Wali keseluruhan
2. Jadwal bimbingan Dosen Wali keseluruhan
3. Rekapitulasi pembimbingan oleh Dosen Wali berdasarkan PS
4. Rekapitulasi beban bimbingan Dosen Wali berdasarkan PS
5. Kalender Akademik
6. Daftar mata kuliah dan prasyarat keseluruhan
7. Daftar mata kuliah berdasarkan kelompok mata kuliah wajib
8. Daftar mata kuliah berdasarkan kelompok mata kuliah peminatan
9. Daftar mata kuliah berdasarkan kelompok mata kuliah pilihan
10. Daftar ruangan dan kapasitas ruangan kuliah dan laboratorium keseluruhan
11. Daftar penawaran mata kuliah dan praktikum
12. Jadwal kuliah dan praktikum keseluruhan
13. Jadwal kuliah dan praktikum berdasarkan PS
14. Jadwal kuliah berdasarkan dosen pengampu mata kuliah
15. Jadwal praktikum berdasarkan laboratorium
16. Daftar mahasiswa tidak ikut Pra-KRS keseluruhan
17. Daftar mahasiswa tidak ikut Pra-KRS berdasarkan PS
18. Daftar mahasiswa tidak ikut Pra-KRS berdasarkan Dosen Wali
19. Daftar mahasiswa peserta kuliah berdasarkan kelas mata kuliah
20. Daftar mahasiswa heregisterasi habis teori berdasarkan PS
21. Daftar mahasiswa heregisterasi habis teori berdasarkan tahun angkatan
22. Daftar mahasiswa peserta KP berdasarkan PS
23. Daftar mahasiswa peserta KP berdasarkan tahun angkatan
24. Daftar mahasiswa peserta KKN berdasarkan PS
25. Daftar mahasiswa peserta KKN berdasarkan tahun angkatan
26. Daftar mahasiswa peserta TA berdasarkan PS
27. Daftar mahasiswa peserta TA berdasarkan tahun angkatan
28. Jadwal kuliah berdasarkan ruangan kuliah
29. Daftar beban SKS dosen dan asisten dosen keseluruhan
30. Daftar beban SKS dosen dan asisten dosen berdasarkan PS
31. Rekapitulasi beban SKS dosen dan asisten dosen keseluruhan
32. Rekapitulasi beban SKS dosen dan asisten dosen berdasarkan PS
33. Rekapitulasi tingkat keberhasilan PRA-KRS keseluruhan
34. Rekapitulasi tingkat keberhasilan PRA-KRS berdasarkan PS
35. Rekapitulasi tingkat keberhasilan KRS keseluruhan
36. Rekapitulasi tingkat keberhasilan KRS berdasarkan PS

5.1.2.4.4. Sub Sistem Informasi Perkuliahan (SIKULIAH)

Sub Sistem Informasi Perkuliahan (SIKULIAH) ini dimaksudkan untuk menangani kegiatan perkuliahan teori untuk seluruh mahasiswa. Kebutuhan informasi dan data dan nilai rinci data untuk SIKULIAH dapat diidentifikasikan berdasarkan kebutuhan informasi dan data untuk
keperluan penyusunan Portofolio, peningkatan mutu, evaluasi diri, borang, perencanaan strategis, perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIKULIAH adalah sebagai berikut:

1. Daftar dosen keseluruhan
2. Daftar dosen tetap
3. Daftar dosen tidak tetap
4. Daftar kehadiran mengajar dosen sebelum UTS keseluruhan
5. Daftar kehadiran mengajar dosen sebelum UTS berdasarkan PS
6. Daftar dosen hadir mengajar <5 kali sebelum UTS keseluruhan
7. Daftar dosen hadir mengajar <5 kali sebelum UTS berdasarkan PS
8. Jadwal UTS berdasarkan PS
9. Daftar Hadir mahasiswa peserta UTS berdasarkan mata kuliah
10. Daftar nilai UTS berdasarkan mata kuliah
11. Daftar kehadiran mengajar dosen sebelum UAS keseluruhan
12. Daftar kehadiran mengajar dosen sebelum UAS berdasarkan PS
13. Daftar dosen hadir mengajar <10 kali sebelum UAS keseluruhan
14. Daftar dosen hadir mengajar <10 kali sebelum UAS berdasarkan PS
15. Daftar jumlah kehadiran kuliah mahasiswa berdasarkan mata kuliah
16. Daftar mahasiswa hadir kuliah <75% sebelum UAS berdasarkan mata kuliah
17. Jadwal UAS berdasarkan PS
18. Daftar Hadir mahasiswa peserta UAS berdasarkan mata kuliah
19. Daftar rincian nilai akhir kuliah berdasarkan mata kuliah
20. Daftar rincian evaluasi prestasi akhir dosen mengajar keseluruhan
21. Daftar rincian evaluasi prestasi akhir dosen mengajar berdasarkan PS
22. Daftar IPK dan batas SKS mahasiswa bimbingan berdasarkan Dosen Wali
23. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 2 semester awal keseluruhan
24. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 2 semester awal berdasarkan PS
25. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 2 semester awal berdasarkan tahun angkatan
26. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 4 semester awal keseluruhan
27. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 4 semester awal berdasarkan PS
28. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 4 semester awal berdasarkan tahun angkatan
29. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 8 semester awal keseluruhan
30. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 8 semester awal berdasarkan PS
31. Daftar mahasiswa tidak memenuhi syarat IPK 2,0 pada 8 semester awal berdasarkan tahun angkatan
32. Daftar prediksi peserta kuliah berdasarkan mata kuliah
33. Daftar prediksi peserta praktikum berdasarkan mata praktikum
34. Daftar prediksi peserta kuliah berdasarkan PS
35. Daftar prediksi peserta praktikum berdasarkan PS
36. Daftar nama Pengawas keseluruhan
37. Daftar materi kuliah dosen berdasarkan mata kuliah
38. Rekapitulasi kehadiran Pengawas UTS
39. Rekapitulasi kehadiran Pengawas UAS
40. Rekapitulasi kehadiran dosen sebelum UTS
41. Rekapitulasi kehadiran dosen sebelum UAS
42. Rekapitulasi kehadiran dosen pada akhir kuliah keseluruhan
43. Rekapitulasi kehadiran dosen pada akhir kuliah berdasarkan PS
44. Rekapitulasi nilai UTS
45. Rekapitulasi Nilai Akhir mata kuliah
46. Rekapitulasi evaluasi prestasi akhir dosen mengajar keseluruhan
47. Rekapitulasi evaluasi prestasi akhir dosen mengajar berdasarkan PS
48. Rekapitulasi penggunaan ruangan kuliah

5.1.2.4.5. Sub Sistem Informasi Praktikum (SIPRAKTIKUM)

Sub Sistem Informasi Praktikum (SIPRAKTIKUM) ini dimaksudkan untuk menangani kegiatan praktikum untuk seluruh mahasiswa. Kebutuhan informasi dan data dan nilai rinci data untuk SIPRAKTIKUM dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan Portofolio, peningkatan mutu, evaluasi diri, borang, perencanaan strategis, perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIPRAKTIKUM adalah sebagai berikut:

1. Kurikulum dan silabus praktikum
2. Ketersediaan dan isi Modul Pratikum
3. Daftar Asisten praktikum keseluruhan
4. Jadwal praktikum berdasarkan laboratorium
5. Jadwal praktikum berdasarkan PS
6. Daftar Hadir Pre-Test mahasiswa peserta praktikum berdasarkan mata praktikum
7. Daftar Hadir mahasiswa peserta praktikum berdasarkan mata praktikum
8. Daftar jumlah kehadiran praktikum mahasiswa berdasarkan mata praktikum
9. Daftar mahasiswa hadir praktikum <75% sebelum responsi berdasarkan mata praktikum
10. Jadwal responsi praktikum berdasarkan mata praktikum
11. Daftar Hadir mahasiswa peserta responsi berdasarkan mata praktikum
12. Daftar rincian nilai akhir praktikum berdasarkan mata praktikum
13. Daftar kehadiran asisten praktikum keseluruhan
14. Daftar kehadiran asisten praktikum berdasarkan mata praktikum
15. Daftar materi praktikum berdasarkan mata praktikum
16. Rekapitulasi kehadiran asisten praktikum berdasarkan laboratorium
17. Rekapitulasi Nilai Akhir praktikum
18. Rekapitulasi penggunaan laboratorium

5.1.2.4.6. Sub Sistem Informasi Kerja Praktek (SIKP)

Sub Sistem Informasi Kerja Praktek (SIKP) ini dimaksudkan untuk menangani kegiatan Kerja Praktek untuk seluruh mahasiswa. Kebutuhan informasi dan data dan nilai rincian data untuk SIKP dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rincian data untuk keperluan penyusunan perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIKP adalah sebagai berikut:

1. Daftar Pembimbing KP keseluruhan
2. Daftar Pembimbing KP berdasarkan PS
3. Daftar rincian KP berdasarkan Dosen Pembimbing KP
4. Daftar rincian KP berdasarkan PS
5. Daftar KP berdasarkan lokasi obyek penelitian
6. Daftar KP berdasarkan topik-tema KP dan PS
7. Daftar KP berdasarkan jenis penelitian KP dan PS
8. Daftar KP lamanya waktu penyelesaian KP
9. Daftar KP yang waktu penyelesaiannya melewati >1 tahun
10. Rekapitulasi jumlah bimbingan KP berdasarkan Dosen Pembimbing KP
11. Rekapitulasi KP berdasarkan lokasi obyek penelitian
12. Rekapitulasi KP berdasarkan topik-tema KP
13. Rekapitulasi Nilai Akhir KP

5.1.2.4.7. Sub Sistem Informasi Kuliah Kerja Nyata (SIKKN)

Sub Sistem Informasi Kuliah Kerja Nyata (SIKKN) ini dimaksudkan untuk menangani kegiatan Kuliah Kerja Nyata untuk seluruh mahasiswa
(S1). Kebutuhan informasi dan data dan nilai rinci data untuk SIKKN dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIKKN adalah sebagai berikut:

1. Daftar Dosen Pembimbing KKN keseluruhan
2. Daftar Dosen Pembimbing KKN berdasarkan PS
3. Daftar mahasiswa peserta KKN berdasarkan PS
4. Daftar mahasiswa peserta KKN berdasarkan kelompok KKN
5. Daftar mahasiswa peserta KKN berdasarkan tahun angkatan
6. Daftar rincian nilai KKN keseluruhan
7. Daftar rincian nilai KKN berdasarkan PS
8. Daftar rincian nilai KKN berdasarkan kelompok KKN
9. Rekapitulasi jumlah peserta KKN berdasarkan PS
10. Rekapitulasi jumlah peserta KKN berdasarkan kelompok KKN
11. Rekapitulasi jumlah peserta KKN berdasarkan tahun angkatan
12. Rekapitulasi Nilai Akhir KKN

5.1.2.4.8. **Sub Sistem Informasi Tugas Akhir (SITA)**

Sub Sistem Informasi Tugas Akhir (SITA) ini dimaksudkan untuk menangani kegiatan Tugas Akhir/Skripsi untuk mahasiswa (S1). Kebutuhan informasi dan data dan nilai rinci data untuk SITA dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan Portofolio, peningkatan mutu, evaluasi diri, borang, perencanaan strategis, perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SITA adalah sebagai berikut:

1. Daftar Pembimbing TA keseluruhan
2. Daftar Pembimbing TA berdasarkan PS
3. Daftar rincian TA berdasarkan Dosen Pembimbing TA
4. Daftar rincian TA berdasarkan PS
5. Daftar TA berdasarkan lokasi obyek penelitian
6. Daftar TA berdasarkan topik-tema TA dan PS
7. Daftar TA berdasarkan lamanya waktu penyelesaian TA
8. Daftar TA yang waktu penyelesaiannya >1 tahun
9. Rekapitulasi jumlah bimbingan TA berdasarkan Dosen Pembimbing TA
10. Rekapitulasi TA berdasarkan lokasi obyek penelitian
11. Rekapitulasi TA berdasarkan topik-tema TA
12. Rekapitulasi Nilai Akhir TA

5.1.2.4.9. Sub Sistem Informasi Yudisium Kelulusan (SIYUDISIUM)

Sub Sistem Informasi Yudisium (SIYUDISIUM) ini dimaksudkan untuk menangani kegiatan Yudisium kelulusan mahasiswa (S1). Kebutuhan informasi dan data dan nilai rinci data untuk SIYUDISIUM dapat diidentifikasi berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIYUDISIUM adalah sebagai berikut:
1. Daftar mahasiswa peserta yudisium keseluruhan
2. Daftar mahasiswa peserta yudisium berdasarkan PS
3. Daftar mahasiswa peserta yudisium berdasarkan tahun angkatan
4. Rekapitulasi mahasiswa peserta yudisium berdasarkan PS
5. Rekapitulasi mahasiswa peserta yudisium berdasarkan tahun angkatan
6. Rekapitulasi mahasiswa peserta yudisium berdasarkan lama masa Studi
7. Rekapitulasi mahasiswa peserta yudisium berdasarkan kelompok IPK

5.1.2.4.10. Sub Sistem Informasi Wisuda (SIWISUDA)

Sub Sistem Informasi Wisuda (SIWISUDA) ini dimaksudkan untuk menangani kegiatan wisuda mahasiswa (S1) yang dinyatakan lulus dalam yudisium. Kebutuhan informasi dan data dan nilai rinci data untuk SIWISUDA dapat diidentifikasikan berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIWISUDA adalah sebagai berikut:
1. Daftar mahasiswa peserta wisuda keseluruhan
2. Daftar mahasiswa peserta wisuda berdasarkan PS
3. Daftar mahasiswa peserta wisuda berdasarkan tahun angkatan
4. Daftar mahasiswa peserta wisuda berdasarkan periode yudisium
5. Rekapitulasi mahasiswa peserta wisuda berdasarkan PS
6. Rekapitulasi mahasiswa peserta wisuda berdasarkan tahun angkatan
7. Rekapitulasi mahasiswa peserta wisuda berdasarkan lama masa Studi
8. Rekapitulasi mahasiswa peserta wisuda berdasarkan kelompok IPK

5.1.2.4.11. Sub Sistem Informasi Alumni (SIALUMNI)

Sub Sistem Informasi Alumni (SIALUMNI) ini dimaksudkan untuk menangani pendataan alumni. Kebutuhan informasi dan data dan nilai rinci data untuk SIALUMNI dapat diidentifikasi berdasarkan kebutuhan informasi dan data dan nilai rinci data untuk keperluan penyusunan perencanaan taktis dan pengendalian manajemen, serta perencanaan dan pengendalian operasional. Hasil identifikasi kebutuhan informasi dan data untuk SIALUMNI adalah sebagai berikut:

1. Daftar alumni keseluruhan
2. Daftar alumni berdasarkan PS
3. Daftar alumni berdasarkan tahun angkatan
4. Daftar alumni berdasarkan periode yudisium
5. Daftar alumni berdasarkan periode wisuda
6. Daftar alumni berdasarkan propinsi
7. Daftar alumni berdasarkan lama waktu tunggu pekerjaan pertama
8. Daftar alumni berdasarkan tingkatan jabatan pekerjaan
9. Daftar alumni berdasarkan kelompok jumlah penghasilan
10. Daftar alumni berdasarkan kelompok bidang pekerjaan
11. Rekapitulasi alumni berdasarkan PS
12. Rekapitulasi alumni berdasarkan tahun angkatan
13. Rekapitulasi alumni berdasarkan periode yudisium
14. Rekapitulasi alumni berdasarkan periode wisuda
15. Rekapitulasi alumni berdasarkan propinsi
16. Rekapitulasi alumni berdasarkan lama waktu tunggu pekerjaan pertama
17. Rekapitulasi alumni berdasarkan tingkatan jabatan pekerjaan
18. Rekapitulasi alumni berdasarkan kelompok jumlah penghasilan
19. Rekapitulasi alumni berdasarkan kelompok bidang pekerjaan

5.1.3. Implementasi Sistem Informasi di ISTA

5.1.3. 1. Implementasi SIAKAD di ISTA

5.1.3.1.1. Kebijakan Pengelolaan Sistem Informasi di ISTA

5.1.3.1.1.1. Kebijakan Sentralisasi Pengelolaan Sistem Informasi

Secara umum, pengelolaan Sistem Informasi di ISTA utamanya dilakukan oleh UPT PUSKOM. Kebijakan sentralisasi seperti ini mempunyai beberapa keuntungan dan faktor pendukung antara lain:
1. Penghematan khususnya dalam *hardware* dan pengadaan personalia
2. Penghematan dalam pengembangan sistem
3. Manfaat standarisasi
4. Sistem yang seragam
5. Kemudahan dalam pengendalian

5.1.3.1.2. **Unit Pelaksana Teknis Pusat Komputer (UPT PUSKOM)**

Sesuai dengan penjelasan di dalam Buku Organisasi dan Tata Kerja ISTA tahun 2000, maka pengembangan sistem, pengolahan data, dan pengelolaan TIK di ISTA dipusatkan di Unit Pelaksana Teknis Pusat Komputer (UPT PUSKOM). UPT PUSKOM merupakan unit pelaksana teknis di bidang pengolahan data yang berada di bawah dan bertanggung jawab langsung kepada Rektor, dan pembinaanya dilakukan oleh Pembantu Rektor I. UPT PUSKOM mempunyai tugas mengumpulkan, mengolah, dan menyimpan data, menyajikan informasi, serta memberikan layanan untuk program-program pendidikan, penelitian, dan pengabdian pada masyarakat.

Untuk melaksanakan tugas tersebut, UPT PUSKOM dibagi menjadi 2 (dua) bagian kelompok tugas, yaitu:
1. Bagian Pemrograman dan Pengolahan Data
2. Bagian Sistem Informasi Terpadu

Hingga saat ini, UPT PUSKOM telah berhasil mengembangkan aplikasi-aplikasi berbasis komputer yang digunakan untuk menangani proses pengolahan data di lingkungan ISTA, yaitu:
1. Sub Sistem Informasi Penerimaan Mahasiswa Baru
2. Sub Sistem Informasi Akademik
3. Sub Sistem Informasi Perpustakaan
4. Sub Sistem Informasi Keuangan Mahasiswa
5. Sub Sistem Informasi Kepegawaian
6. Sub Sistem Informasi Inventaris
Selain itu, aplikasi-aplikasi berukuran kecil dengan karakteristik yang spesifik juga telah dikembangkan dan diimplementasikan sebagai modul-modul yang terpisah, yaitu:

1. Aplikasi untuk pengolahan data Penggajian dosen-karyawan
2. Aplikasi untuk pengolahan data wisuda
3. Aplikasi untuk pengolahan data Praktikum
4. Aplikasi untuk pengolahan data Kuliah Kerja Nyata
5. Aplikasi untuk pengolahan data PKN-KP-Merencana-TA-Skripsi
6. Aplikasi untuk pengolahan data Evaluasi Kinerja Dosen Mengajar
7. Aplikasi untuk pengolahan data Realisasi dan Pencairan Anggaran

5.1.3.1.3.  5.1.3.1.3. Biro Administrasi Perencanaan dan Sistem Informasi (BAPSI)

Biro Administrasi Perencanaan dan Sistem Informasi (BAPSI) merupakan salah satu biro di lingkungan ISTA yang mempunyai tugas utama melaksanakan layanan administrasi perencanaan dan sistem informasi. BAPSI dipimpin oleh seorang Kepala Biro, yang mempunyai tugas sebagai berikut:

1. Menyusun rencana dan program kerja BAPSI
3. Mengumpulkan, mengolah, dan mengevaluasi data: (1) penerimaan mahasiswa baru; (2) semua jenis yang berkaitan dengan pelaksanaan sistem pembelajaran; (3) yudisium dan wisuda; (4) kepegawaian, baik tenaga edukatif maupun tenaga non-edukatif; (5) sarana dan prasarana sistem pembelajaran; (6) hasil penelitian dosen; (7) hasil pengabdian kepada masyarakat; (8) pelaksanaan proyek, pelatihan, dan lokakarya; (9) kegiatan yang berkaitan kerjasama dengan lembaga lain.
4. Menyusun laporan semesteran untuk Dikti dan Kopertis (EPSBED)
5. Menyusun data yang diperlukan untuk perencanaan pengembangan sistem pembelajaran
6. Koordinator dan kendali tugas Kepala Bagian
7. Pencairan anggaran biro

BAPSI terdiri dari 2 (dua) bagian, yaitu:

1. Bagian Analisis dan Perencanaan, mempunyai tugas: (1) mengumpulkan data akademik, dosen, pegawai, mahasiswa; (2) mengumpulkan data penerimaan mahasiswa baru; (3) melakukan analisa data; (4) membuat laporan hasil analisa data.
2. Bagian Sistem Informasi, mempunyai tugas: (1) melengkapi data yang dikumpulkan oleh Bagian Analisis dan Perencanaan dengan data kegiatan institut lainnya; (2) memelihara dan mengembangkan situs website http://www.akprind.ac.id; (3) memelihara dan mengembangkan sarana e-learning.

Dengan adanya 6 (enam) sub sistem informasi dan 7 (tujuh) aplikasi pengolahan data berbasis komputer yang telah dikembangkan oleh UPT PUSKOM dan kemudian diterapkan di lingkungan ISTA, maka kebutuhan informasi (dan data) untuk sebagian besar pengguna di lingkungan ISTA sudah terpenuhi. Sekalipun demikian, pada kenyataannya, institut, fakultas, biro, lembaga, jurusan/prodi, atau unit-unit lainnya masih memerlukan informasi (dan data) yang bersifat spesifik, seperti laporan Evaluasi Program Studi Berdasarkan Evaluasi Diri (EPSBED) yang dikirim ke Kopertis (SK 013) pada setiap akhir semester (Lampiran 5), Laporan Tahunan Rektor pada setiap akhir tahun, penyusunan proposal PHK dan lainnya, ijin penyelenggaraan program studi, re-akreditasi program studi, dan lainnya. Pengolahan data untuk keperluan spesifik tersebut dilakukan oleh BAPSI. Sesuai dengan fungsi dan tugasnya, untuk memenuhi keperluan informasi (dan data) yang bersifat spesifik, BAPSI mengumpulkan data dari unit-unit kerja terkait, dan kemudian diolah lebih lanjut oleh BAPSI, dan kemudian disampaikan kepada pihak yang membutuhkannya.
5.1.3.2. Kondisi Infrastruktur TIK

5.1.3.2.1. Hardware

1. LAN-Intranet

ISTA memiliki prasarana berupa 3 (tiga) buah gedung pada lokasi yang terpisah satu dengan yang lain, yaitu Kampus Balapan (I), Kampus Kotabaru (II), dan Kampus Bima Sakti (III). Di setiap gedung tersebut telah dipasang jaringan LAN-Intranet yang menghubungkan dengan Kampus Balapan. Untuk menjaga keandalan interkoneksi jaringan LAN-Intranet antar kampus, digunakan peralatan dengan spesifikasi yang memadai.

Interkoneksi antara Kampus I dan Kampus II yang berjarak sekitar 2 Km, dan Kampus I dan Kampus III yang berjarak sekitar 1 Km menggunakan teknologi Wireless LAN 2.4 Ghz ke ISP. Pada Kampus III yang di dalamnya terdapat ruangan kantor Fakultas Sains Terapan (FST), Jurusan di bawah FST, Lemlit, kantor LPM, dan 6 (enam) laboratorium komputer juga telah terhubung dalam jaringan LAN-Intranet. Interkoneksi tersebut juga merupakan embrio yang potensial bagi upaya pengembangan ke arah digital campus.

2. Internet

Sejak 4 (empat) tahun terakhir ini, ISTA telah menyediakan ruangan khusus yang di dalamnya tersedia 24 terminal yang terhubung dengan LAN-Intranet, dan digunakan untuk proses entry KRS secara mandiri bagi seluruh mahasiswa ISTA. Dengan melakukan pengaturan jadwal entry pada masa pengisian KRS, maka pemanfaatan ini dirasakan sangat efektif, dimana keterlambatan pengisian KRS yang banyak terjadi pada masa sebelumnya menjadi sangat jauh berkurang. Selanjutnya, sarana dan fasilitas yang disediakan tersebut dikembangkan dan ditingkatkan pemanfaatannya untuk pelayanan akses internet bagi mahasiswa ISTA.
Saat ini, ISTA terkoneksi ke dalam jaringan global internet dengan menggunakan Wireless LAN 2,4 Ghz ke ISP dengan koneksi sebesar 256 Kbps. ISTA juga sudah terhubung dengan jaringan local loop jaringan DIY-Jateng Jogja Internet Exchange (JIX).

Untuk memberikan pelayanan akses internet bagi dosen, telah isediakan sebuah komputer yang terkoneksi ke internet pada setiap ruang kantor jurusan dan unit-unit kerja tertentu di lingkungan ISTA. Koneksi ini telah dimanfaatkan oleh dosen di lingkungan ISTA sebagai sarana mengakses berbagai macam informasi yang bersumber dari internet, baik untuk keperluan pendidikan dan pengajaran, penelitian, pengabdian kepada masyarakat, dan lain-lain. Spesifikasi peralatan yang tersedia untuk koneksi internet tersebut telah memadai, dalam kondisi baik, dan dapat digunakan dengan baik.

Warnet Batistanet juga menyediakan koneksi internet kepada mahasiswa di sekitar Kampus Balapan dengan menggunakan antena omnidirectional 2.4 Ghz, yang merupakan area hotspot di lingkungan kampus.


Sebagian dosen ISTA juga memanfaatkan fasilitas Kontak yang tersedia dalam website tersebut untuk komunikasi maupun memberikan layanan konsultasi kepada mahasiswa bimbingannya, baik yang menyangkut tentang masalah perkuliahan, tugas, bimbingan penulisan laporan penelitian, KRS, dan lainnya. Website yang telah dikembangkan
tersebut merupakan embrio yang diharapkan dapat dikembangkan menjadi fasilitas yang mendukung proses pembelajaran melalui e-learning.

5.1.3.2.2. Software

Pemilihan software untuk pengembangan, implementasi, dan pengelolaan pada sub sistem informasi dan aplikasi di lingkungan ISTA, utamanya didasarkan pada alasan kemudahan migrasi dari sistem lama ke sistem yang baru, keandalan, mengarah ke open source, serta memungkinkan untuk dihubungkan dengan sistem yang lebih luas secara lebih mudah karena mendukung aplikasi web based.

5.1.3.2.3. Kondisi Sumberdaya Pendukung dan Pengelola TIK

Untuk mendukung terlaksananya tugas dan fungsi pada UPT PUSKOM, maka UPT PUSKOM ditangani oleh personil yang memiliki kompetensi yang memadai di dalam bidangnya. Untuk mendukung terlaksananya tugas dan fungsi BAPSI, maka BAPSI ditangani oleh personil yang memiliki kompetensi yang memadai di dalam bidangnya.

Untuk menjaga kelancaran kegiatan operasional laboratorium, maka setiap laboratorium dipimpin oleh seorang Kepala Laboratorium (berstatus dosen) dan 1 (satu) orang tenaga teknik (teknisi). Tenaga teknik (teknisi) telah diberikan pelatihan yang sesuai dengan tugasnya dan mampu menggunakan / mengoperasikan sarana / peralatan yang tersedia dengan terampil / baik.

5.1.3.3. Sistem Informasi dan Pemanfaatannya di ISTA

Kebutuhan Sistem Informasi di Perguruan Tinggi (SIPT) dapat berbeda antara Perguruan Tinggi satu dengan yang lainnya. Perbedaan ini utamanya dipengaruhi oleh 3 (tiga) hal, yaitu perbedaan aturan bisnis (business rule), perbedaan ukuran organisasi, dan perbedaan area fungsional dalam organisasi Perguruan Tinggi. Perbedaan aturan bisnis
terjadi karena perbedaan peraturan dan kebijakan yang digariskan; perbedaan ukuran organisasi terjadi karena perbedaan jumlah Fakultas, Jurusan, dan Program Studi yang diselenggarakan, serta jumlah mahasiswa; sedangkan perbedaan area fungsional dalam organisasi terjadi karena perbedaan kompleksitas pengolahan dan struktur organisasi. Oleh karena itu, kebutuhan SIPT di masing-masing Perguruan Tinggi dapat memiliki fungsi, tujuan, ruang lingkup, serta implementasi yang berbeda-beda. SIPT dapat berupa sebuah sistem terintegrasi ataupun sekumpulan aplikasi yang bersifat otonom.

Saat ini ISTA telah menerapkan beberapa sub sistem informasi berbasis komputer yang dikembangkan oleh UPT PUSKOM, yaitu:

1. Sub Sistem Informasi Penerimaan Mahasiswa Baru.
2. Sub Sistem Informasi Akademik.
4. Sub Sistem Informasi Keuangan Mahasiswa.
5. Sub Sistem Informasi Kepegawaian.

Sementara itu, aplikasi-aplikasi berukuran kecil dengan karakteristik yang spesifik juga telah dikembangkan dan diimplementasikan sebagai modul-modul yang terpisah, yaitu:

1. Aplikasi untuk pengolahan data Penggajian dosen-karyawan.
2. Aplikasi untuk pengolahan data wisuda.
3. Aplikasi untuk pengolahan data Praktikum.
4. Aplikasi untuk pengolahan data Kuliah Kerja Nyata.
7. Aplikasi untuk pengolahan data Realisasi dan Pencairan Anggaran.

Pengembangan dan penerapan sistem informasi berbasis komputer di lingkungan ISTA memiliki 2 (dua) macam tujuan, yaitu:

1. Untuk memenuhi kebutuhan administrasi umum, keuangan, dan administrasi akademik.
2. Untuk memenuhi kebutuhan data statistik yang diperlukan dalam pengelolaan setiap Jurusan/Program Studi.

Secara umum, setiap sub sistem informasi dan aplikasi untuk pengolahan data yang telah dikembangkan oleh UPT PUSKOM dan diterapkan di lingkungan ISTA, mampu memberikan output berupa informasi (dan data) dalam format terinci, teringkas/rekapitulasi, maupun spesifik; baik yang bersifat insidental maupun rutin; dalam bentuk teks, tabel, maupun grafis; serta dapat ditampilkan pada media softcopy maupun hardcopy. Dengan tampilan dialog berbasis visual yang user friendly, maka informasi (dan data) yang dihasilkan dapat diakses secara mudah dan cepat. Output informasi (dan data) tersebut dapat diakses/digunakan oleh para pengguna di lingkungan ISTA maupun pihak luar; sesuai dengan kewenangannya, yang dapat dikelompokkan menjadi 4 (empat), yaitu:

1. Pengguna pada tingkat manajemen tertinggi (perencanaan strategis).
2. Para pengguna pada tingkat manajemen menengah (perencanaan taktis dan pengendalian manajemen).
3. Para pengguna pada tingkat manajemen terendah (perencanaan dan pengendalian operasional).
4. Personil atau institusi pengguna lainnya, yaitu dosen, mahasiswa, alumni, masyarakat pengguna lulusan, pemerintah (Depdiknas, Dikti, BAN-PT, Kopertis), penyandang dana (Yayasan Pembina Potensi Pembangunan/YPPP), orang tua/wali mahasiswa, serta masyarakat umum.

5.1.3.3.1. Sub Sistem Informasi Penerimaan Mahasiswa Baru

Sub Sistem Informasi Penerimaan Mahasiswa Baru (SIPENMARU) di ISTA mulai diimplementasikan sejak tahun 1999. SIPENMARU terdiri atas 6 (enam) modul, yaitu:

1. Penjualan formulir
2. Pendaftaran mahasiswa baru
3. Ujian penerimaan mahasiswa baru
4. Yudisium penerimaan mahasiswa baru
5. Heregisterasi mahasiswa baru
6. KRS mahasiswa baru, yaitu secara paket semester I bagi mahasiswa baru jalur reguler dan tanpa tes, serta mengisikan matakuliah berdasarkan KRS mahasiswa baru jalur pindahan dan alih jalur yang sebelumnya telah disetujui oleh Dosen Wali

Hardware, software, dan database pendukung pada SIPENMARU dikelola oleh UPT PUSKOM. SIPENMARU dapat diakses melalui terminal/client dari semua unit kerja di lingkungan ISTA yang telah terhubung dengan jaringan LAN-Intranet. Selain itu, SIPENMARU juga didukung dengan 12 unit komputer khusus untuk menangani pengolahan data Penmaru yang dipasang dalam ruang Sekretariat Penmaru. Terminal/client digunakan oleh Unit P3MB dan BAA untuk melakukan transaksi dalam modus Web Base dengan menggunakan Sistem Operasi Windows XP.

SIPENMARU dapat diakses oleh para pengguna secara lokal menggunakan jaringan LAN-Intranet di ISTA. Sejak tahun 2004, ISTA menyediakan layanan informasi Pengumuman Hasil Seleksi Penmaru melalui SMS dengan tarip khusus. Pendaftaran mahasiswa baru secara on line sudah bisa dilakukan sejak 2 (dua) tahun terakhir, namun sistem pendaftaran mahasiswa baru secara on line ini belum terintegrasi dengan SIPENMARU. Data-data pendaftaran mahasiswa baru yang di-input-kan secara on line ini masih harus di-entry-kan kembali oleh petugas di bagian pendaftaran sehingga masih perlu dikembangkan lebih lanjut. Pengembangan SIPENMARU perlu dilakukan agar dapat memenuhi fungsifungsi pengolahan data perpustakaan secara lengkap, dan diintegrasikan menjadi Sistem Informasi Perguruan Tinggi ISTA (SIPT) yang terpadu. SIPENMARU juga berpotensi untuk di-upload ke jaringan Internet setelah dilakukan perbaikan pada sisi keamanannya.
5.1.3.3.2. Sub Sistem Informasi Akademik

Sub Sistem Informasi Akademik (SIA) di ISTA memiliki fungsi utama untuk mengolah data kegiatan akademik mahasiswa mulai dari hereregisterasi mahasiswa baru hingga lulus. SIA versi pertama mulai diimplementasikan pada tahun 1996, pada tahun-tahun awal ini SIA menggunakan server *Novel Netware 4.01* dan dibangun dengan bahasa pemrograman *Clipper 5.0* dengan database *dBase+ Plus* (dbf). Terminal/client yang digunakan saat itu masih menggunakan PC 386 dan PC 486.

Setelah beroperasi dua tahun, pada tahun 1998 beberapa *interface* dan database dialihkan ke *MS-Access 97*. Bersamaan dengan itu, terminal/client yang digunakan dialihkan ke *Netware 4.01* dengan menggunakan sistem operasi *Windows 95* dan *Windows 98* dengan spesifikasi komputer Pentium 233 memori 64 MB.

Seiring dengan perkembangan teknologi dan kebutuhan pengguna, pada tahun 1999 SIA dikembangkan menjadi *Web Server*, dan hingga saat ini sebagian dari sistem tersebut masih digunakan untuk beberapa pekerjaan. Kemajuan teknologi saat ini, memungkinkan penggunaan banyak *interface* dan banyak *database* namun tetap saling terhubung. SIA yang diterapkan saat ini memiliki 12 modul, yaitu:

1. Hereregisterasi mahasiswa. Modul ini digunakan untuk melakukan validasi mahasiswa yang melakukan herregistrasi, berdasarkan jenis her yang dilakukan, serta jumlah biaya yang disetorkan melalui bank. Herregistrasi mahasiswa dikelompokkan menjadi 3 (tiga) jenis, yaitu herregistrasi aktif, herregistrasi aktif terbatas, dan cuti.
2. Penentuan dosen wali
3. Penawaran mata kuliah dan dosen pengampu mata kuliah
4. Penjadwalan dan pembagian kelas perkuliahan
5. Pengisian Kartu Rencana Studi (KRS)
6. Beban mengajar dosen
7. Pelaksanaan perkuliahan, yaitu presensi kuliah (dosen dan mahasiswa) serta catatan materi perkuliahan yang diberikan oleh dosen di kelas
8. Pembagian ruang Ujian Tengah Semester (UTS) dan Akhir Semester (UAS), presensi ujian
9. Pemasukan nilai UTS dan UAS
10. Distribusi nilai setiap matakuliah
11. Kartu Hasil Studi (KHS)
12. Lulusan, mencakup yudisium, serta pencetakan transkrip nilai sementara dan pencetakan ijazah

Hardware, software, dan database pendukung pada SIA dikelola oleh UPT PUSKOM. SIA dapat diakses melalui terminal/client dari semua unit kerja di lingkungan ISTA yang telah terhubung dengan jaringan LAN-Intranet, yang dibagi menjadi 3 (tiga) kelompok.

Dengan menggunakan spesifikasi tersebut, maka SIA masih mungkin untuk dikembangkan lebih lanjut. Pengembangan SIA perlu dilakukan agar dapat memenuhi fungsi-fungsi pengolahan data akademik secara lengkap, dan diintegrasikan menjadi SIPT yang terpadu. SIA juga berpotensi untuk di-upload ke jaringan Internet setelah dilakukan perbaikan pada sisi keamanannya. SIA juga merupakan embrio yang potensial bagi upaya pengembangan ke arah digital campus.

5.1.3.3.3. Sub Sistem Informasi Perpustakaan

Pengolahan data pada UPT Perpustakaan telah menerapkan sistem informasi perpustakaan (SIPRUS) yang memiliki fungsi utama untuk pengelolaan koleksi bahan pustaka (katalog, penambahan, penghapusan), anggota, dan pelayanan transaksi buku (peminjaman, pengembalian, denda terlambat/hilang), termasuk kepada anggota dari luar ISTA (Guest), serta fasilitas pencarian. SIPRUS dikembangkan oleh pengembang dari luar ISTA bersama-sama dengan UPT PUSKOM dan mulai
diimplementasikan sejak tahun 2004. SIPRUS dikembangkan dengan menggunakan *PHP* dan database *MySQL* dan *PostgreSQL*.

*Hardware, software, dan database* pendukung pada SIPRUS dikelola oleh UPT PUSKOM. SIPRUS dapat diakses melalui terminal/client dari semua unit kerja di lingkungan ISTA yang telah terhubung dengan jaringan LAN-Intranet, yang terbagi dalam 2 (dua) kelompok.

Dengan menggunakan spesifikasi tersebut, maka SIPRUS sangat mungkin untuk dikembangkan lebih lanjut. Pengembangan SIPRUS perlu dilakukan agar dapat memenuhi fungsi-fungsi pengolahan data perpustakaan secara lengkap, dan diintegrasikan menjadi SIPT yang terpadu.

Pada saat ini UPT Perpustakaan sedang mengupayakan ke arah *digital library* dengan melakukan proses digitalisasi bahan-bahan pustaka antara lain laporan tugas akhir/skripsi mahasiswa, tesis, dan disertasi, sehingga nantinya SIPRUS dapat di-upload ke jaringan Internet setelah dilakukan perbaikan pada sisi keamanannya. Untuk meningkatkan pelayanan perpustakaan kepada mahasiswa dan pengguna dari institusi lain, maka penyempurnan *digital library* perlu diupayakan, terutama bahan-bahan pustaka yang mendukung untuk penelitian dosen dan mahasiswa.

5.1.3.3.4. **Sub Sistem Informasi Keuangan**


*Hardware, software, dan database* pendukung pada SIKEUANGAN dikelola oleh UPT PUSKOM. SIKEUANGAN bisa diakses dalam format
web/PHP melalui terminal/client dari semua unit kerja di lingkungan ISTA yang telah terhubung dengan jaringan LAN-Intranet, yang terbagi dalam 2 (dua) kelompok.

Dengan menggunakan spesifikasi tersebut, maka SIKEUANGAN sangat mungkin untuk dikembangkan lebih lanjut. Pengembangan SIKEUANGAN perlu dilakukan agar dapat memenuhi fungsi-fungsi pengolahan data keuangan dan akuntansi secara lengkap, dan diintegrasikan menjadi SIPT yang terpadu. SIKEUANGAN juga berpotensi untuk di-upload ke jaringan Internet setelah dilakukan perbaikan pada sisi keamanannya.

5.1.3.3.5. Sub Sistem Informasi Kepegawaian


Dengan menggunakan spesifikasi tersebut, maka SIPEGAWAI masih mungkin untuk dikembangkan lebih lanjut. Pengembangan SIPEGAWAI perlu dilakukan agar dapat memenuhi fungsi-fungsi pengolahan data kepegawaian secara lengkap, dan diintegrasikan menjadi SIPT yang terpadu. Akhirnya, SIPEGAWAI juga berpotensi untuk di-upload ke jaringan Internet setelah dilakukan perbaikan pada sisi keamanannya.

5.1.3.3.6. Sub Sistem Informasi Inventaris

Sub Sistem Informasi Inventaris (SIINVENTARIS) di ISTA memiliki fungsi utama untuk mengolah data semua barang inventaris (pembelian, distribusi, monitoring barang-barang inventaris) yang dimiliki oleh ISTA.

*Hardware, software, dan database* pendukung pada SIINVENTARIS dikelola oleh UPT PUSKOM, dan memiliki 1 (satu) kelompok terminal/client.

Dengan menggunakan spesifikasi tersebut, maka SIINVENTARIS masih mungkin untuk dikembangkan lebih lanjut. Pengembangan SIINVENTARIS perlu dilakukan agar dapat memenuhi fungsi-fungsi pengolahan data inventaris secara lengkap, dan diintegrasikan menjadi SIPT yang terpadu. Akhirnya, SIINVENTARIS juga berpotensi untuk di-upload ke jaringan Internet setelah dilakukan perbaikan pada sisi keamanannya.

5.1.3.3.7. Aplikasi Untuk Pengolahan Data Penggajian Dosen-Karyawan


5.1.3.3.8. Aplikasi Untuk Pengolahan Data Praktikum

Pengolahan data praktikum di laboratorium yang ada dalam lingkungan ISTA masih dilaksanakan secara berbeda-beda untuk setiap laboratorium. Hal ini terjadi karena setiap laboratorium memiliki
karakteristik yang bersifat spesifik, sehingga hingga saat ini belum tertangani oleh UPT PUSKOM. Sebagian besar pengolahan data di laboratorium masih dilakukan secara manual, walaupun menggunakan alat bantu komputer, karena belum tersedia aplikasi khusus untuk pengolahan data praktikum.

Sedangkan khusus untuk laboratorium komputer yang dikelola oleh Jurusan Teknik Informatika, dengan adanya dukungan SDM yang memadai, maka pengolahan data praktikum dilaksanakan dengan menggunakan aplikasi pengolahan data berbasis komputer yang dikembangkan oleh masing-masing laboratorium komputer. Aplikasi ini bersifat stand alone. Secara umum, aplikasi pengolahan data praktikum yang dikembangkan dapat menangani pengolahan data yang terkait dengan pendaftaran praktikum, asisten, jadwal, daftar hadir praktikan dan asisten, responsi, penilaian, dan pelaporan praktikum ke Jurusan. Hardware, software, dan database pendukung yang digunakan pada aplikasi ini menggunakan peralatan dan fasilitas yang tersedia di masing-masing laboratorium komputer.


5.1.3.3.9. Aplikasi Untuk Pengolahan Data KP-Merencana-TA-Skripsi

Pengolahan data Kerja Praktek (KP), Merencana, Tugas Akhir, dan Skripsi di lingkungan ISTA masih dilaksanakan secara terpisah pada setiap jurusan. Hal ini terjadi karena setiap jurusan memiliki karakteristik yang bersifat spesifik, sehingga hingga saat ini belum tertangani oleh UPT USKOM. Pengolahan data ini masih dilakukan secara manual, walaupun
menggunakan alat bantu komputer, tetapi belum tersedia aplikasi khusus untuk pengolahan data.


5.1.3.3.10. Aplikasi Untuk Pengolahan Data Kuliah Kerja Nyata


Aplikasi ini masih potensial untuk dikembangkan lebih lanjut. Pengembangan aplikasi ini perlu dilakukan agar dapat memenuhi fungsi-
fungsi pengolahan data KKN secara lengkap, dan diintegrasikan menjadi SIPT yang terpadu. Akhirnya, aplikasi ini juga berpotensi untuk di-upload ke jaringan Internet setelah dilakukan perbaikan pada sisi keamanannya. Aplikasi ini diharapkan dapat memberikan dukungan yang sangat berarti pada upaya pengembangan menuju digital campus.

5.1.3.3.11. Aplikasi Untuk Pengolahan Data Evaluasi Kinerja Dosen Mengajar


Untuk mendukung pengolahan data evaluasi kinerja dosen mengajar di ISTA, telah dikembangkan aplikasi khusus yang bersifat stand alone dan telah diterapkan sejak tahun 2003. Hardware, software, dan database pendukung yang digunakan pada aplikasi ini menggunakan peralatan dan fasilitas yang tersedia di BAA dan UPT PUSKOM.

5.1.3.3.12. Aplikasi Untuk Pengolahan Data Realisasi dan Pencairan Anggaran

Aplikasi untuk pengolahan data realisasi dan pencairan anggaran dikembangkan oleh UPT PUSKOM dengan fungsi utama untuk melayani pengolahan data realisasi dan pencairan anggaran. Aplikasi ini dikembangkan dengan menggunakan MS-Access 97, dan mulai diimplementasikan sejak tahun 2002. Aplikasi ini masih diterapkan secara stand alone. Aplikasi ini menyediakan fasilitas pengolahan data untuk: pengisian data unit, pengisian anggaran unit, pengisian detail anggaran per mata anggaran, mencetak anggaran, pencairan anggaran, mencetak detail anggaran, koreksi pencairan anggaran, persetujuan realisasi, serta laporan-laporan

Hardware, software, dan database pendukung yang digunakan pada aplikasi ini menggunakan peralatan dan fasilitas yang tersedia di Sekretariat Rektor dan UPT PUSKOM.

5.1.4. Schema Database SIAKAD di ISTA

Penerapan SIAKAD di ISTA saat ini didukung oleh database akademik yang diimplementasikan dengan menggunakan database server model RDBMS, yaitu PostgreSQL. Schema database akademik ISTA tersusun atas 181 tabel, secara rinci ditampilkan pada Lampiran 1.
5.2. Pembahasan

5.2.1. Instalasi Database Server PostgreSQL

Untuk melakukan analisis/pengujian schema database akademik ISTA, dilakukan instalasi Database Server PostgreSQL dengan menggunakan sistem operasi GNU-/Linux debian dilakukan oleh user root dengan perintah berikut:

```
apt-get install Postgresql-7.4.
```

5.2.2. Pembuatan Duplikasi Database Akademik

Setelah Database Server terinstall maka dibuat duplikasi database akademik dengan melakukan dump pada schema database di server pada UPT Puskom ISTA dengan menjalankan perintah berikut:

```
pg dump -s akademik > akademik
```

Program pg dump merupakan utilitas bawaan dari PostgreSQL yang digunakan untuk membackup database ke dalam suatu file teks. Perintah di atas akan melakukan backup pada database akademik pada bagian schema saja. Cuplikan dari hasil perintah di atas adalah sebagai berikut:

```
-- PostgreSQL database dump
--
SET client_encoding = 'LATIN1';
SET check_function_bodies = false;
SET SESSION AUTHORIZATION 'postgres';
--
-- TOC entry 4 (OID 2200)
-- Name: public; Type: ACL; Schema: ; Owner: postgres
--
REVOKE ALL ON SCHEMA public FROM PUBLIC;
GRANT ALL ON SCHEMA public TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';
SET search_path = public, pg_catalog;
--
-- TOC entry 32 (OID 1218695)
-- Name: aa; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE aa (aa character(1) DEFAULT 'n'::bpchar);
```
5.2.3. Membuat Database Duplikasi

Setelah dilakukan perintah **pg dump** maka dilanjutkan dengan memasukan hasil backup ke dalam komputer yang digunakan untuk melakukan penelitian. Sedikit perubahan dilakukan dalam file hasil backup disesuaikan dengan keadaan database server tempat penelitian
dilakukan. Perubahan yang dilakukan adalah dengan menganti hak kepemilikan database dari user jack menjadi wa2n. Selanjutnya memasukan script backup yang telah dimodifikasi dalam database server. Untuk melakukan kegiatan ini digunakan perintah,

a2n@debian:/home/data/artikel/penelitian/database/data$ createdb akademik
CREATE DATABASE
wa2n@debian:/home/data/artikel/penelitian/database/data$ psql akademik < akademik1

Pada database akademik terdapat 181 tabel. Gambar 5.1 menampilkan beberapa tabel dalam database akademik.

Gambar 5.1: Tabel dalam database akademik

5.2.4. Analisis/Pengujuan Aspek-Aspek Kualitas Schema Database

Pada langkah ini dilakukan analisis untuk setiap aspek kualitas schema database yang diperoleh. Analisis dilakukan dengan cara...
melakukan ujicoba langsung pada schema database untuk SIAKAD yang digunakan di ISTA. Aspek-aspek kualitas schema database yang akan dianalisis meliputi:

1. Kebenaran (correctness)
2. Konsistensi (consistency)
3. Relevansi (relevance)
4. Jangkauan (scope)
5. Tingkat detail (level of detail)
6. Kelengkapan (completeness)
7. Minimalitas (minimality)
8. Kemampuan untuk diintegrasikan (ability of integration)
9. Kemampuan untuk dibaca (readability)

5.2.4.1. Analisis/Pengujian Aspek Kebenaran (Correctness)

Kebenaran data merupakan aspek teknik, yaitu apakah semua aspek dimodelkan secara benar sesuai dengan kebutuhan dan batasan sistem. Aspek ini dapat digunakan untuk mengukur semua schema. Pengukuran dilakukan menggunakan kepakaran dengan meninjau tingkat kedalaman pengetahuan pada aspek teknik ini. Berikut schema mahasiswa:

```
akademik=>
Table "public.mahasiswa"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>nomhs</td>
<td>character varying(10)</td>
<td>not null</td>
</tr>
<tr>
<td>nama</td>
<td>character varying(50)</td>
<td></td>
</tr>
<tr>
<td>tgl_lahir</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>lelaki</td>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td>agama</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>warga</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>goldar</td>
<td>character varying(2)</td>
<td></td>
</tr>
<tr>
<td>kdosen</td>
<td>character varying(10)</td>
<td></td>
</tr>
<tr>
<td>nikah</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>jalur</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>kota_lahir</td>
<td>character varying(50)</td>
<td></td>
</tr>
</tbody>
</table>
```
Pada suatu database kebenaran data merupakan hal yang mutlak harus dipenuhi, data yang dimasukan ke dalam database haruslah merupakan data yang legal. Legal di sini berarti sesuai dengan syarat dan kententuan yang harus dipenuhi oleh data. Database yang baik sebisa mungkin harus mampu mem-filter suatu data sehingga data yang dimasukan melalui interface apapun dapat diseleksi.

Pada contoh schema di atas terlihat bahwa hampir semua kolom memungkinkan menerima data yang tidak legal. Pada schema di atas tidak terdapat suatu batasan (constraint) yang dapat mem-filter masuknya data ke dalam tabel sehingga tidak ada jaminan bahwa data yang dimasukan merupakan data yang benar.

Sebagai contoh adalah pada kolom dengan tipe data smallint maka jika tidak ada constraint apapun, semua data yang memenuhi syarat smallint yaitu -32.000 sampai dengan 32.000 akan dapat masuk ke dalam database tersebut. Padahal beberapa kolom tidak memerlukan data sebanyak itu, misalnya dalam kolom agama dipastikan cacahnya tidak melebihi dari 10. Kolom nama juga demikian, pada kolom tersebut terlihat bahwa nama dapat berisi NULL ataupun spasi kosong (white space). Sebagai contoh dilakukan perintah INSERT sebagai berikut,

```sql
akademik=> INSERT INTO mahasiswa(nomhs,nama,tgllahir,lelaki,agama,warga,goldar,kdosen,nikah,jalur,kotalahir,jur,ang,alamatyk,kdpos)
VALUES ('ABCDEFGHIJ','','1/1/2007',TRUE,29000,30000,'XZ','BCDEEFGHIJ',31999,-31999,'Sleman','XX','ABCD','Sleman','123456789A');
```

```sql
akademik=> SELECT nomhs,nama,nikah,jalur from mahasiswa;
```

<table>
<thead>
<tr>
<th>nomhs</th>
<th>nama</th>
<th>nikah</th>
<th>jalur</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDEFGHIJ</td>
<td></td>
<td>31999</td>
<td>-31999</td>
</tr>
</tbody>
</table>
Dari data di atas terlihat bahwa isi tabel mahasiswa terlihat janggal atau aneh karena data yang dimasukan memang merupakan data yang tidak legal atau tidak sah. Kolom nama dapat dipastikan merupakan sebuah kolom yang pasti terisi minimal 1 karakter dan tidak NULL. Sehingga untuk kolom ini dapat diberikan suatu constraint yang dapat mengurangi kemungkinan masuknya data yang tidak legal.

```
akademik=> ALTER TABLE mahasiswa ALTER COLUMN nama SET NOT NULL;
AKETER TABLE
akademik=> ALTER TABLE mahasiswa ADD CHECK (nama <> '');
AKETER TABLE
```

Perintah di atas akan membuat tabel mahasiswa memiliki suatu constraint sehingga jika dilakukan pemasukan data dimana nama mahasiswa berupa karakter kosong atau NULL maka database akan menolaknya. Dengan pernambahan constraint di atas maka schema menjadi berubah seperti berikut,

```
akademik=> 'd mahasiswa;
Table "public.mahasiswa"
Column | Type                | Modifiers
-------+--------------------+-----------
nomhs  | character varying(10) | not null
nama   | character varying(50) | not null
tgllahir | date                | |
lelaki  | boolean             | |
agama   | smallint            | |
warga   | smallint            | |
goldar  | character varying(2) | |
kdosen  | character varying(10) | |
nikhah  | smallint            | |
jalur   | smallint            | |
kotalahir | character varying(50) | |
jur     | character varying(2) | |
ang     | character varying(4) | |
alamatyk | character varying(100) | |
kdpos   | character varying(10) | |

Indexes:
"mahasiswa_pkey" primary key, btree (nomhs)
```

Check constraints:
"$1" CHECK (nama::text <> ""::text)

Pemasukan data yang tidak sesuai dengan syarat akan berakibat ditolaknya data tersebut,

akademik=> DELETE from mahasiswa ;
DELETE 1
akademik=> INSERT INTO
mahasiswa(nomhs,nama,tgllahir,lelaki,agama,warga,
goldar,kdosen,nikah,jalur,kotalahir,jur,ang,alamatyk,kdpos)
VALUES
('ABCDEFGHIJ','','1/1/2007',TRUE,29000,30000,'XZ','BCDEEFGHIJ',
31999,31999,'Sleman','XX','ABCD','Sleman','123456789A');
ERROR: new row for relation "mahasiswa" violates check constraint "$1"

akademik=> INSERT INTO
mahasiswa(nomhs,nama,tgllahir,lelaki,agama,warga,
goldar,kdosen,nikah,jalur,kotalahir,jur,ang,alamatyk,kdpos)
VALUES
('ABCDEFGHIJ',NULL,'1/1/2007',TRUE,29000,30000,'XZ','BCDEEFGHIJ',
31999,31999,'Sleman','XX','ABCD','Sleman','123456789A');
ERROR: null value in column "nama" violates not-null constraint

Pada database akademik masih sangat banyak tabel yang tidak memiliki constraint sehingga pem-filter-an data yang masuk ke dalam database dilakukan melalui interface databasenya yaitu dari bahasa pemograman yang digunakan.

Dalam kasus database akademik ISTA digunakan PHP dan Microsoft Acces, dengan cara ini maka tingkat kesulitan pembuatan program cukup tinggi karena harus melakukan pengecekan data sehingga data tersebut legal dan kemungkinan terjadi kesalahannya cukup besar. Pem-filter-an melalui interface juga memiliki kekurangan jika terjadi perubahan interface atau penambahan interface. Jika terjadi migrasi interface dari satu bahasa pemograman ke bahasa pemograman yang lain maka dipastikan terjadi kesulitan, apalagi jika hanya diberikan schema database tanpa disertai dengan kode sumber interface sebelumnya.
5.2.4.2. Analisis/Pengujiand Aspek Konsistensi (Consistency)

Konsistensi data dalam suatu database merupakan salah satu aspek teknik, yaitu apakah semua aspek dalam model terbebas dari kontradiksi. Aspek konsistensi dan kebenaran sangat penting untuk mengukur apakah schema diterima oleh pengguna atau tidak. Pengukuran dilakukan menggunakan kepakaran teknik dengan menganalisis konsistensi setiap aspek teknik pada model dan membandingkannya dengan aspek teknik berikutnya.

Konsistensi data dalam suatu database merupakan hal yang mutlak. Perbedaan isi data pada suatu kolom atau kumpulan kolom dengan maksud sama dalam satu schema dengan schema lain merupakan hal yang tidak diperbolehkan dalam suatu database. Pada database akademik beberapa schema memiliki informasi yang seharusnya bisa didapatkan dari schema lain (menggunakan kerelasian), akan tetapi tidak dilakukan pada database tersebut, seperti contoh berikut,

```sql
akademik=> \d mahasiswa;
Table "public.mahasiswa"
Column | Type            | Modifiers
-------|----------------|-------------
nomhs  | character varying(10) | not null
nama   | character varying(50) | not null
tgllahir | date          | 
lelaki | boolean        | 
agama  | smallint       | 
warga  | smallint       | 
goldar | character varying(2) | 
kdozen | character varying(10) | 
nikah  | smallint       | 
jalur  | smallint       | 
kotalahir | character varying(50) | 
jur    | character varying(2) | 
ang    | character varying(4) | 
alamatyk | character varying(100) | 
kdpos  | character varying(10) | 
Indexes:
"mahasiswa_pkey" primary key, btree (nomhs)
Check constraints:
"$1" CHECK (nama::text <> "":text)
Pada tabel akademik terdapat sebuah kolom kdosen yang merupakan kode dosen ISTA yang berasal dari schema dosen.

<p>| akademik=&gt; \d dosen |
|---------------|------------------|------------------|</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>kode</td>
<td>character varying(6)</td>
<td>not null</td>
</tr>
<tr>
<td>nama</td>
<td>character varying(50)</td>
<td></td>
</tr>
<tr>
<td>gelar1</td>
<td>character varying(25)</td>
<td></td>
</tr>
<tr>
<td>gelar2</td>
<td>character varying(25)</td>
<td></td>
</tr>
<tr>
<td>nodos</td>
<td>character varying(25)</td>
<td></td>
</tr>
<tr>
<td>kodefak</td>
<td>character varying(4)</td>
<td></td>
</tr>
<tr>
<td>goldos</td>
<td>character varying(10)</td>
<td></td>
</tr>
<tr>
<td>jabolos</td>
<td>character varying(25)</td>
<td></td>
</tr>
<tr>
<td>jsk</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>jklas</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>jgabung</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>character varying(4)</td>
<td></td>
</tr>
<tr>
<td>kodegol</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>aktif</td>
<td>character varying(25)</td>
<td></td>
</tr>
<tr>
<td>urut</td>
<td>integer</td>
<td></td>
</tr>
</tbody>
</table>

Indexes:
"dosen_pkey" primary key, btree (kode)

Namun dari dua schema tersebut tidak memiliki relasi, sehingga memungkinkan kode dosen di schema akademik akan berbeda dengan dosen. Kedua schema tersebut seharusnya direlasikan untuk menjaga konsistensi data.

| akademik=> ALTER TABLE mahasiswa ADD CONSTRAINT referensi_kode_dosen |
| akademik=> FOREIGN KEY (kdosen) REFERENCES dosen; |
| ALTER TABLE |

Pada tabel mahasiswa jika dilihat akan ada penambahan constraint:

Foreign-key constraints:
"referensi_kode_dosen" FOREIGN KEY (kdosen) REFERENCES dosen(kode)

Perintah di atas akan merelasikan tabel mahasiswa dan dosen, sehingga isi dari kdosen di tabel mahasiswa hanya akan dapat berisi sesuai dengan kode dosen yang ada di tabel dosen. Jika dicoba diisi dengan kode yang tidak ada di tabel dosen maka data tersebut akan
ditolak. Sebagai contoh misalnya di tabel dosen belum berisi data apapun sehingga tidak memungkinkan untuk mengisi tabel mahasiswa.

\[\text{akademik} \Rightarrow \text{INSERT INTO}\]
\[\text{mahasiswa(nomhs, nama, tgl_lahir, lelaki, agama, warga, goldar, kdosen)\}
\[\text{ERROR: new row for relation "mahasiswa" violates check constraint "}1”\]
\[\text{akademik} \Rightarrow \]

Pada ujicoba di atas terlihat bahwa terjadi penolakan data karena melanggar constraint yang ada.

5.2.4.3. Analisis/Pengujian Aspek Relevansi (Relevance)

Relevansi merupakan salah satu aspek teknik, yaitu apakah aspek- aspek teknik pada schema relevan digunakan. Aspek ini penting untuk mengetahui apakah schema diterima oleh pengguna atau tidak. Pengukuran dilakukan menggunakan kepakaran teknik untuk menentukan seluruh aspek yang relevan dan membandingkannya dengan schema.

Dalam database akademik ISTA terdapat beberapa schema yang tidak ada relevansinya dengan schema lainnya. Sebagai contoh terdapat sebuah schema buuning seperti tertampil berikut,

\[\text{akademik} \Rightarrow \text{d buuning}\]
\[\text{Table "public.buuning"}\]
\[\text{Column} | \text{Type} | \text{Modifiers}\]
\[\text{tgl} | \text{date} | \text{1}\]
\[\text{level} | \text{smallint} | \text{1}\]
\[\text{jumlah} | \text{integer} | \text{1}\]

Peneliti tidak melihat relvansi dari schema ini dengan yang lainnya. Adanya permintaan suatu sistem tambahan baru pada sebuah sistem yang berjalan memungkinkan terjadinya hal ini.

5.2.4.4. Analisis/Pengujian Aspek Jangkauan (Scope)

Jangkauan merupakan salah satu aspek teknik, yaitu apakah jangkauan schema telah sesuai dengan kebutuhan pengguna. Aspek ini penting untuk mengetahui apakah schema diterima oleh pengguna atau
tidak. Lingkup *schema* bersifat relatif mengacu pada tujuan proyek. Pengukuran dilakukan menggunakan kepakaran untuk menentukan seluruh kebutuhan proyek dan membandingkannya dengan *schema*.

Schema database akademik ISTA memiliki jangkauan yang minim, sehingga akan menimbulkan kesulitan pemeliharaan jika terjadi penambahan kebutuhan baru, termasuk di saat ingin mengintegrasikan database sehingga menjadi SIPT yang terintegrasi.

### 5.2.4.5. Analisis/Pengujian Aspek Tingkat Detail (Level Of Detail)

Tingkat detail merupakan salah satu aspek teknik, yaitu apakah tingkat detail *schema* telah sesuai. Aspek ini penting untuk mengetahui apakah *schema* diterima oleh pengguna atau tidak. Pengukuran dilakukan menggunakan kepakaran teknik untuk menentukan tingkat detail yang diinginkan dan membandingkannya dengan *schema*.

Schema database akademik ISTA belum mampu memenuhi semua kebutuhan para penggunanya, kondisi akibat schema yang masih kurang mendetail.

### 5.2.4.6. Analisis/Pengujian Aspek Kelengkapan (Completeness)

Kelengkapan merupakan salah satu aspek teknik, yaitu apakah *schema* telah lengkap (dengan mengacu pada kebutuhan). Aspek ini penting untuk mengetahui apakah *schema* dapat diterima oleh pengguna atau tidak. Pengukuran dapat dilakukan dari aspek jangkauan dan tingkat detail (sebelumnya).

Schema database akademik ISTA memuat banyak schema yang tidak ada relevansinya dengan schema lainnya.

### 5.2.4.7. Analisis/Pengujian Aspek Minimalitas (Minimality)

Tingkat detail merupakan salah satu aspek teknik, yaitu apakah *schema* dimodelkan secara kompak dan tidak ada perulangan. Aspek ini penting karena *schema* konseptual harus tepat. Pengukuran dilakukan
menggunakan kepakaran teknik untuk mengecek apakah terdapat aspek-aspek yang dimodelkan secara berulang.

Sangat penting memiliki schema database yang presisi, tidak ada perulangan schema. Schema konseptual database seharusnya sudah bisa menangani permasalahan secara tepat. Pada database akademik ISTA terdapat beberapa pengulangan schema dengan berbagai alasan yang melatarbelakanginya.

Misalnya, relasi dkhs01, dkhs02, dkhs03, dkhs04, dkhs05, dkhs06 dan dkhs07, semua relasi tersebut memiliki schema yang sama yakni,

```
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>nomhs</td>
<td>character</td>
<td>not null</td>
</tr>
<tr>
<td>idmtk</td>
<td>integer</td>
<td>not null</td>
</tr>
<tr>
<td>nilai</td>
<td>character</td>
<td></td>
</tr>
</tbody>
</table>
```

Indexes:
"dkhs04_pkey" primary key, btree (nomhs, idmtk)

Penambahan field merupakan salah satu alternatif yang dapat dilakukan untuk menghindari berulangnya schema dalam database.

5.2.4.8. Analisis/Penguji Aspek Kemampuan Untuk Diintegrasikan (Ability Of Integration)

Kemampuan untuk diintegrasikan merupakan salah satu aspek teknis, yaitu apakah schema telah disesuaikan untuk standarisasi dalam organisasi secara menyeluruh atau pemodelan. Aspek ini tergantung pada konteks proyek atau organisasi, yaitu apakah lebih banyak terlibat pada asosiasi ekonomi atau berorientasi internasional, tetapi yang lebih relevan jika memiliki kemampuan untuk diiintegrasikan. Pengukuran dilakukan untuk menentukan apakah penggunaan nama-nama hanya dapat digunakan untuk cabang atau dapat diterima secara internasional.

Schema database akademik ISTA belum menerapkan standarisasi data dalam organisasi secara menyeluruh, sehingga akan menimbulkan
kesulitan pada saat akan mengintegrasikan database sehingga menjadi SIPT yang terintegrasi.

5.2.4.9. Analisis/Pengujuan Aspek Kemampuan Untuk Dibaca (Readability)

Kemampuan untuk dibaca merupakan salah satu aspek teknis, yaitu apakah semua istilah dalam schema dijelaskan atau tidak. Aspek ini penting, khususnya untuk dialog dengan staf teknik dan orang yang datang belakangan. Pengukuran dilakukan dengan meninjau dokumen untuk menentukan apakah setiap aspek mudah dipahami.

Schema yang ada pada database akademik ISTA memiliki tingkat kesulitan untuk dibaca yang tinggi, karena dalam schema tersebut tidak ada satupun keterangan yang melekat pada setiap schema. Pemberian keterangan pada schema akan memudahkan untuk memahami apa maksud dari schema tersebut. Informasi keterangan yang lebih detail akan berguna pada saat melakukan migrasi atau pembacaan schema oleh Database Administrator baru.

Dari database akademik ISTA terlihat adanya beberapa schema yang akan membingungkan. Salah satunya adalah sebagai berikut,

<table>
<thead>
<tr>
<th>List of relations</th>
<th>Name</th>
<th>Type</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>aa</td>
<td>table</td>
<td>wa2n</td>
</tr>
<tr>
<td>public</td>
<td>agama_agamaid_seq</td>
<td>sequence</td>
<td>wa2n</td>
</tr>
</tbody>
</table>

5.2.5. Rekapitulasi Hasil Analisis/Pengujuan Aspek-Aspek Kualitas Schema Database

Berdasarkan hasil analisis/pengujian pada bagian sebelumnya, secara ringkas dapat dibuat rekapitulasi hasil analisis/pengujian aspek-aspek kualitas schema database pada database akademik ISTA, secara ringkas ditampilkan pada Tabel 5.2.
<table>
<thead>
<tr>
<th>Kriteria</th>
<th>Deskripsi</th>
<th>Hasil Analisis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kebenaran (correctness)</td>
<td>Merupakan aspek teknik, apakah semua aspek diimplementasikan secara benar sesuai dengan kebutuhan dan batasan sistem. Aspek ini dapat digunakan untuk mengukur semua schema. Pengukuran dilakukan menggunakan kepakaran dengan meninjau tingkat kedalaman pengetahuan pada seluruh aspek teknik.</td>
<td><em>Schema database</em> akademik ISTA tidak memiliki suatu batasan (constraint) yang dapat mem-filter masuknya data ke dalam tabel sehingga tidak ada jaminan bahwa data yang dimasukkan merupakan data yang benar</td>
</tr>
<tr>
<td>Konsistensi (consistency)</td>
<td>Merupakan aspek teknik, apakah semua aspek dalam model terbebas dari kontradiksi. Aspek konsistensi dan kebenaran sangat penting untuk mengukur apakah schema diterima oleh pengguna atau tidak. Pengukuran dilakukan menggunakan kepakaran teknik dengan menganalisis konsistensi setiap aspek teknik pada model dan membandingkannya dengan aspek teknik berikutnya.</td>
<td><em>Schema database</em> akademik ISTA memuat banyak data redundancy, kondisi ini mengakibatkan potensi yang besar terhadap terjadinya data inconsistency</td>
</tr>
<tr>
<td>Relevansi (relevance)</td>
<td>Merupakan aspek teknik, apakah aspek-aspek teknik pada schema relevan digunakan. Aspek ini penting untuk mengetahui apakah schema diterima oleh pengguna atau tidak. Pengukuran dilakukan menggunakan kepakaran teknik untuk menentukan seluruh aspek yang relevan dan membandingkannya dengan schema.</td>
<td><em>Schema database</em> akademik ISTA memuat banyak schema yang tidak ada relevansinya dengan schema lainya</td>
</tr>
<tr>
<td>Tingkat detail (level of detail)</td>
<td>Apakah tingkat detail schema telah sesuai. Aspek ini penting untuk mengetahui apakah schema diterima oleh pengguna atau tidak. Pengukuran dilakukan menggunakan kepakaran teknik untuk menentukan tingkat detail yang diinginkan dan membandingkannya dengan schema.</td>
<td><em>Schema database</em> akademik ISTA belum mampu memenuhi semua kebutuhan para penggunanya, kondisi akibat schema yang masih kurang mendetail</td>
</tr>
<tr>
<td>Kelengkapan (completeness)</td>
<td>Apakah schema telah lengkap (dengan mengacu pada kebutuhan). Aspek ini penting untuk mengetahui apakah schema dapat diterima oleh pengguna</td>
<td><em>Schema database</em> akademik ISTA memuat banyak schema yang tidak ada relevansinya dengan</td>
</tr>
</tbody>
</table>
atau tidak. Pengukuran dapat dilakukan dari aspek jangkauan dan tingkat detail (sebelumnya).

<table>
<thead>
<tr>
<th>Minimalitas (minimality)</th>
<th>Apakah <em>schema</em> dimodelkan secara kompak dan tidak ada perulangan. Aspek ini penting karena <em>schema</em> konseptual harus tepat. Pengukuran dilakukan menggunakan kepakaran teknik untuk mengecek apakah terdapat aspek-aspek yang dimodelkan secara berulang.</th>
<th><em>Schema database</em> akademik ISTA memuat banyak pengulangan <em>schema</em>, dengan berbagai alasan yang melatarbelakanginya, utamanya kemudahan pada saat <em>programming</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Kemampuan untuk diintegrasikan (ability of integration)</td>
<td>Apakah <em>schema</em> telah disesuaikan untuk standarisasi dalam organisasi secara menyeluruh atau pemodelan. Aspek ini tergantung pada konteks proyek atau organisasi, yaitu apakah lebih banyak terlibat pada asosiasi ekonomi atau berorientasi internasional, tetapi yang lebih relevan jika memiliki kemampuan untuk diintegrasikan. Pengukuran dilakukan untuk menentukan apakah penggunaan nama-nama hanya dapat digunakan untuk cabang atau dapat diterima secara internasional.</td>
<td><em>Schema database</em> akademik ISTA belum menerapkan standarisasi data dalam organisasi secara menyeluruh, sehingga akan menimbulkan kesulitan pada saat akan mengintegrasikan database sehingga menjadi SIPT yang terintegri</td>
</tr>
<tr>
<td>Kemampuan untuk dibaca (readability)</td>
<td>Apakah semua istilah dalam <em>schema</em> dijelaskan atau tidak. Aspek ini penting, khususnya untuk dialog dengan staf teknik dan orang yang datang belakangan. Pengukuran dilakukan dengan meninjau dokumen untuk menentukan apakah setiap aspek mudah dipahami.</td>
<td><em>Schema</em> yang ada pada database akademik ISTA memiliki tingkat kesulitan untuk dibaca yang tinggi, karena <em>schema</em> tersebut tidak ada satupun keterangan yang melekat pada setiap <em>schema</em></td>
</tr>
</tbody>
</table>

BAB VI
KESIMPULAN DAN SARAN

6.1. Kesimpulan

Perkembangan teknologi informasi-komputer saat ini sudah mencapai pada tahap di mana ukurannya semakin kecil, kecepatannya semakin tinggi, namun harganya semakin murah dibandingkan dengan kemampuan kerjanya. Kondisi ini mendorong masyarakat berlomba-lomba memanfaatkan komputer sebagai alat bantu pengolahan data dengan cara membangun sistem pengolahan data terkomputerisasi untuk penyajian informasi, baik untuk keperluan pribadi maupun organisasinya. Sekalipun kurang tepat, sistem pengolahan data terkomputerisasi untuk penyajian informasi dalam suatu organisasi sering disebut sebagai Sistem Informasi Manajemen (SIM).

Kenyataan di lapangan menunjukkan, bahwa tidak semua organisasi berhasil mengimplementasikan SIM dengan baik. Eko Nugroho (1991) telah meneliti pengaruh struktur organisasi dan sumber daya manusia terhadap keberhasilan implementasi SIM dalam organisasi dalam penelitian tesisnya. Hasil penelitian tersebut membuktikan, bahwa rata-rata persentase tingkat keberhasilan implementasi SIM, khususnya di DIY relatif rendah, hanya sekitar 60%.


Kegagalan implementasi SIM dan fenomena “tambal sulam” aplikasi SIM ternyata juga dapat terjadi dalam SIAKAD. Hal ini dapat dibuktikan
dengan masih adanya PT yang telah melakukan pengembangan dan implementasi SIAKAD lebih dari satu dekade lamanya, namun hasilnya belum memuaskan hingga saat ini.

Dengan menggunakan meta model, penelitian Olaf Herden (2001) telah berhasil mendefinisikan dan mengukur aspek-aspek kualitas schema berorientasi obyek (object oriented) suatu database, dan mengusulkan sebuah proses untuk melakukan tinjauan dan pengukuran aspek kualitas schema suatu database. Aspek-aspek yang relevan untuk pengukuran kualitas schema suatu database yang dimaksud meliputi 9 (sembilan) kriteria, yaitu kebenaran (correctness), konsistensi (consistency), relevansi (relevance), jangkauan (scope), tingkat detail (level of detail), kelengkapan (completeness), minimalitas (minimality), kemampuan untuk diintegrasikan (ability of integration), serta kemampuan untuk dibaca (readability).

Penelitian ini berhasil melakukan analisis terhadap aspek-aspek kualitas schema database. Analisis dilakukan pada rancangan schema database yang digunakan pada database akademik yang digunakan di ISTA. Analisis dilakukan pada rancangan logikal, bukan implementasi pada DBMS, teknologi perangkat keras, maupun program aplikasi yang digunakan. Dan, analisis dilakukan dengan menggunakan model logika berupa business rule yang digunakan pada ISTA.

Penelitian dilakukan dengan menggunakan perangkat keras berupa seperangkat komputer dan perangkat lunak sistem operasi GNU/Linux Distribusi Slackware 9.1 Kernel 2.6.7 dan PostgreSQL 7.4.14.

Berdasarkan hasil analisis/penguji, secara umum dapat dinyatakan bahwa schema database akademik ISTA memiliki kualitas yang rendah.

6.2. Saran

Kondisi kualitas schema database akademik ISTA yang rendah akan menjadi potensi bagi timbulnya berbagai permasalahan yang serius di kelak kemudian hari. Dan oleh karenanya, maka perlu segera dilakukan upaya-upaya penyempurnaan untuk menghindari timbulnya permasalahan yang lebih kompleks dan semakin sulit ditangani.
DAFTAR PUSTAKA


AMULET Development Corp, 2004, How To Modify Database Structure, eCriteria, Web Hosted Database for Business


Badan Akreditasi Nasional Perguruan Tinggi Departemen Pendidikan Nasional, 2001, Pedoman Evaluasi Diri Program Studi, Jakarta

Badan Akreditasi Nasional Perguruan Tinggi Departemen Pendidikan Nasional, 2001, Pedoman Penyusunan Portofolio Akreditasi Program Studi, Jakarta

Badan Akreditasi Nasional Perguruan Tinggi Departemen Pendidikan Nasional, 2001, Panduan Pengisian Borang Akreditasi Program Studi S1, Jakarta


Departemen Pendidikan Nasional, 1999, Peraturan Pemerintah Nomor 60 Tahun 1999 Tentang Pendidikan Tinggi, Jakarta


Eppler, M.J. and Wittig, D., 2000, Conceptualizing Information Quality: A Review of Information Quality Frameworks from the Last Ten Years, presented at 5th International Conference on Information Quality (ICIQ 2000), Cambridge, MA


Indrajit, R.E., 2000, Manajemen Sistem Informasi dan Teknologi Informasi, PT. Elex Media Komputindo, Jakarta

ISTA, 2002, Portofolio IST AKPRIND, Yogyakarta


Loudon, 2004, Management Information Systems 8/e, Prentice Hall


ISTA, 2002, Evaluasi Diri Program Studi Teknik Informatika FTI, IST AKPRIND, Yogyakarta


Redman, T.C., 1996, Data Quality for the Information Age, Boston, MA: Artech House

References: OS/400 DB2/400 Database-An Overview (SC41-3700-00, CD-ROM QBKAUB01), OS/400 DDS Reference (SC41-3712-00, CD-ROM QBKAUI01)


Sumber Pustaka dari Internet:


Mullen, B., 2005, *Generalized Table Suggestions*, http://www2.cstudies.ubc.ca/~mullen/IEDBS.html#GeneralTables, diakses pada tanggal 11-08-2005


-- PostgreSQL database dump

SET client_encoding = 'LATIN1';
SET check_function_bodies = false;

SET SESSION AUTHORIZATION 'postgres';

--
-- TOC entry 4 (OID 2200)
-- Name: public; Type: ACL; Schema: -; Owner: postgres
--
REVOKE ALL ON SCHEMA public FROM PUBLIC;
GRANT ALL ON SCHEMA public TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

SET search_path = public, pg_catalog;

--
-- TOC entry 32 (OID 1218695)
-- Name: aa; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE aa (aa character(1) DEFAULT 'n'::bpchar);

--
-- TOC entry 5 (OID 1218700)
-- Name: agama_agamaid_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE agama_agamaid_seq
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;

--
-- TOC entry 6 (OID 1218700)
-- Name: agama_agamaid_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE agama_agamaid_seq FROM PUBLIC;
GRANT ALL ON TABLE agama_agamaid_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 33 (OID 1218702)
-- Name: beban; Type: TABLE; Schema: public; Owner: jack
CREATE TABLE beban (  mreg character varying(4),  kode character varying(6),  jkelas integer,  jsks integer,  jur character varying(2) );

REVOKE ALL ON TABLE beban FROM PUBLIC;  
GRANT ALL ON TABLE beban TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE biayapendek (  mreg character varying(4) NOT NULL,  harga integer NOT NULL,  beauji real,  batasklas smallint );

REVOKE ALL ON TABLE biayapendek FROM PUBLIC;  
GRANT ALL ON TABLE biayapendek TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE buuning (  tgl date,  "level" smallint,  jumlah integer );

-- TOC entry 38 (OID 1218706)  
-- Name: buuning; Type: ACL; Schema: public; Owner: jack

REVOKE ALL ON TABLE buuning FROM PUBLIC;
GRANT ALL ON TABLE buuning TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 39 (OID 1218708)
-- Name: detilkhs01; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE detilkhs01 (
  nomhs character varying(10),
  mreg character varying(4),
  idmtk integer,
  nilai character varying(1)
);

--
-- TOC entry 40 (OID 1218710)
-- Name: detkhs; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE detkhs (
  idkhs integer DEFAULT nextval('detkhs_idkhs_seq'::text) NOT NULL,
  nomhs character varying(10) NOT NULL,
  idmtk integer NOT NULL,
  n1 character(1),
  n2 character(1),
  nilai character(1),
  mutu smallint,
  kredit smallint
);

--
-- TOC entry 41 (OID 1218710)
-- Name: detkhs; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE detkhs FROM PUBLIC;
GRANT ALL ON TABLE detkhs TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 42 (OID 1218713)
-- Name: detkhs01; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE detkhs01 (
  iddetkhs integer DEFAULT nextval('"detkhs01_iddetkhs_seq"'::text)
  NOT NULL,
  idkhs integer,
  nomhs character varying(10),
  idmtk integer,
  n1 character(1),
  n2 character(1),
  nilai character(1),
  mutu smallint,
  kredit smallint
SET SESSION AUTHORIZATION 'jack';

-- TOC entry 43 (OID 1218713)
-- Name: detkhs01; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE detkhs01 FROM PUBLIC;
GRANT ALL ON TABLE detkhs01 TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 7 (OID 1218716)
-- Name: detkhs01_iddetkhs_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE detkhs01_iddetkhs_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

-- TOC entry 8 (OID 1218716)
-- Name: detkhs01_iddetkhs_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE detkhs01_iddetkhs_seq FROM PUBLIC;
GRANT ALL ON TABLE detkhs01_iddetkhs_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 9 (OID 1218718)
-- Name: detkhs_idkhs_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE detkhs_idkhs_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

-- TOC entry 10 (OID 1218718)
-- Name: detkhs_idkhs_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE detkhs_idkhs_seq FROM PUBLIC;
GRANT ALL ON TABLE detkhs_idkhs_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 44 (OID 1218720)
REVOKE ALL ON TABLE dkhs02 FROM PUBLIC;
GRANT ALL ON TABLE dkhs02 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 50 (OID 1218726)
-- Name: dkhs03; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE dkhs03 (nomhs character varying(10) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1))
;

--
-- TOC entry 51 (OID 1218726)
-- Name: dkhs03; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE dkhs03 FROM PUBLIC;
GRANT ALL ON TABLE dkhs03 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 52 (OID 1218728)
-- Name: dkhs04; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE dkhs04 (nomhs character varying(10) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1))
;

--
-- TOC entry 53 (OID 1218728)
-- Name: dkhs04; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE dkhs04 FROM PUBLIC;
GRANT ALL ON TABLE dkhs04 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 54 (OID 1218730)
-- Name: dkhs05; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE dkhs05 (nomhs character varying(10) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1))
;
-- TOC entry 55 (OID 1218732)
-- Name: dkhs06; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE dkhs06 (
    nomhs character varying(10) NOT NULL,
    idmtk integer NOT NULL,
    nilai character varying(1)
);

-- TOC entry 56 (OID 1218732)
-- Name: dkhs06; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE dkhs06 FROM PUBLIC;
GRANT ALL ON TABLE dkhs06 TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 57 (OID 1218734)
-- Name: dkhs07; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE dkhs07 (
    nomhs character varying(10) NOT NULL,
    idmtk integer NOT NULL,
    nilai character varying(1)
);

-- TOC entry 58 (OID 1218736)
-- Name: dosen; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE dosen (
    kode character varying(6) NOT NULL,
    nama character varying(50),
    gelar1 character varying(25),
    gelar2 character varying(25),
    nodos character varying(25),
    kodefak character varying(4),
    goldos character varying(10),
    jabdos character varying(25),
    jsks smallint,
    jklas smallint,
    jgabung smallint,
    status character varying(4),
    kodegol smallint,
    aktif character varying(25),
    urut integer
);

-- TOC entry 59 (OID 1218736)
-- Name: dosen; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE dosen FROM PUBLIC;
GRANT ALL ON TABLE dosen TO ista;

SET SESSION AUTHORIZATION 'jack';
--
-- TOC entry 60 (OID 1218738)
-- Name: dosenjur; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE dosenjur {
  kddosen character varying(6) NOT NULL,
  kdjur character varying(2)
};

--
-- TOC entry 61 (OID 1218738)
-- Name: dosenjur; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE dosenjur FROM PUBLIC;
GRANT ALL ON TABLE dosenjur TO ista;

SET SESSION AUTHORIZATION 'jack';
--
-- TOC entry 62 (OID 1218740)
-- Name: fakultas; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE fakultas {
  kodefak character varying(4) NOT NULL,
  fakultas character varying(50)
};

--
-- TOC entry 63 (OID 1218740)
-- Name: fakultas; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE fakultas FROM PUBLIC;
GRANT ALL ON TABLE fakultas TO ista;

SET SESSION AUTHORIZATION 'jack';
--
-- TOC entry 64 (OID 1218742)
-- Name: fasilitas; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE fasilitas {
  "level" smallint NOT NULL,
  ket text
};
-- TOC entry 65 (OID 1218742)
-- Name: fasilitas; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE fasilitas FROM PUBLIC;
GRANT ALL ON TABLE fasilitas TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 66 (OID 1218747)
-- Name: format; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE format (
    orientasi character(1),
    pgsetup character varying(15)
);

--
-- TOC entry 67 (OID 1218747)
-- Name: format; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE format FROM PUBLIC;
GRANT ALL ON TABLE format TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 68 (OID 1218749)
-- Name: hari; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE hari (
    idhari smallint NOT NULL,
    namahari character varying(6) NOT NULL
);

--
-- TOC entry 69 (OID 1218749)
-- Name: hari; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE hari FROM PUBLIC;
GRANT ALL ON TABLE hari TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 70 (OID 1218751)
-- Name: herdibaa; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE herdibaa (
    mreg character varying(4),
    nomhs character varying(10),
    --


```sql
-- TOC entry 71 (OID 1218751)
-- Name: herdibaa; Type: ACL; Schema: public; Owner: jack

REVOKE ALL ON TABLE herdibaa FROM PUBLIC;
GRANT ALL ON TABLE herdibaa TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 72 (OID 1218753)
-- Name: honor; Type: TABLE; Schema: public; Owner: jack

CREATE TABLE honor (
    mreg character varying(4) NOT NULL,
    uhadir real NOT NULL,
    ukoreksi real NOT NULL,
    unaskah real NOT NULL,
    ubonus real NOT NULL,
    uajar real NOT NULL
);

-- TOC entry 73 (OID 1218753)
-- Name: honor; Type: ACL; Schema: public; Owner: jack

REVOKE ALL ON TABLE honor FROM PUBLIC;
GRANT ALL ON TABLE honor TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 74 (OID 1218755)
-- Name: hutangnilai; Type: TABLE; Schema: public; Owner: jack

CREATE TABLE hutangnilai (
    kode character varying(6) NOT NULL,
    mreg character varying(4) NOT NULL,
    fileher character varying(8) NOT NULL,
    idkelas integer
);

-- TOC entry 75 (OID 1218755)
-- Name: hutangnilai; Type: ACL; Schema: public; Owner: jack

REVOKE ALL ON TABLE hutangnilai FROM PUBLIC;
GRANT ALL ON TABLE hutangnilai TO ista;
```
SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 76 (OID 1218757)
-- Name: identitas; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE identitas (  
id character varying(10) NOT NULL,
"password" character varying(10),
"level" smallint,
ket text,
identitas text,
aktif character varying(1)
);

--
-- TOC entry 77 (OID 1218757)
-- Name: identitas; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE identitas FROM PUBLIC;
GRANT ALL ON TABLE identitas TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 78 (OID 1218762)
-- Name: jalur; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE jalur (  
id integer DEFAULT nextval('jalur_id_seq'::text) NOT NULL,
jalurnya character varying(20) NOT NULL
);

--
-- TOC entry 79 (OID 1218762)
-- Name: jalur; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE jalur FROM PUBLIC;
GRANT ALL ON TABLE jalur TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 11 (OID 1218765)
-- Name: jalur_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE jalur_id_seq  
INCREMENT BY 1  
MAXVALUE 2147483647  
NO MINVALUE  
CACHE 1;

--
-- TOC entry 12 (OID 1218765)
-- Name: jalur_id_seq; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE jalur_id_seq FROM PUBLIC;
GRANT ALL ON TABLE jalur_id_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 80 (OID 1218767)
-- Name: jalurdipakai; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE jalurdipakai (
  id smallint,
  jalur character varying(20),
  kdj smallint,
  jalurnya character varying(20),
  angkatan character varying(4)
);

--
-- TOC entry 81 (OID 1218767)
-- Name: jalurdipakau; Type: TABLE; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE jalurdipakau FROM PUBLIC;
GRANT ALL ON TABLE jalurdipakau TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 82 (OID 1218769)
-- Name: jalurdipakau; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE jalurdipakau (
  id integer,
  jalur character varying(50),
  angkatan character varying(4),
  kdj integer,
  jalurnya character varying(50)
);

--
-- TOC entry 83 (OID 1218769)
-- Name: jalurdipakau; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE jalurdipakau FROM PUBLIC;
GRANT ALL ON TABLE jalurdipakau TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 84 (OID 1218771)
-- Name: jam; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE jam (  
jamke smallint NOT NULL,  
jam character varying(5) NOT NULL,  
isi boolean DEFAULT false
);

--
-- TOC entry 85 (OID 1218771)
-- Name: jam; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE jam FROM PUBLIC;
GRANT ALL ON TABLE jam TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 86 (OID 1218774)
-- Name: jenisher; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE jenisher (  
id integer DEFAULT nextval('"jenisher_id_seq"'::text) NOT NULL,  
jenis character varying(35)
);

--
-- TOC entry 87 (OID 1218774)
-- Name: jenisher; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE jenisher FROM PUBLIC;
GRANT ALL ON TABLE jenisher TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 13 (OID 1218777)
-- Name: jenisher_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--

CREATE SEQUENCE jenisher_id_seq  
INCREMENT BY 1  
MAXVALUE 2147483647  
NO MINVALUE  
CACHE 1;

--
-- TOC entry 14 (OID 1218777)
-- Name: jenisher_id_seq; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE jenisher_id_seq FROM PUBLIC;
GRANT ALL ON TABLE jenisher_id_seq TO ista;
SET SESSION AUTHORIZATION 'jack';

-- TOC entry 88 (OID 1218779)
-- Name: jeniskpta; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE jeniskpta (  
   kode character varying(4) NOT NULL,
   jenis character varying(25)
);

-- TOC entry 89 (OID 1218779)
-- Name: jeniskpta; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE jeniskpta FROM PUBLIC;
GRANT ALL ON TABLE jeniskpta TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 90 (OID 1218781)
-- Name: jsmta; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE jsmta (  
   kode character varying(10) NOT NULL,
   jurusan character varying(50) NOT NULL
);

-- TOC entry 91 (OID 1218781)
-- Name: jsmta; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE jsmta FROM PUBLIC;
GRANT ALL ON TABLE jsmta TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 92 (OID 1218783)
-- Name: khs; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khs (  
   nomhs character varying(10) NOT NULL,
   ipk real,
   jsks smallint,
   jmtk smallint,
   jmutu smallint
);

-- TOC entry 93 (OID 1218783)
-- Name: khs; Type: ACL; Schema: public; Owner: jack
REVOKE ALL ON TABLE khs FROM PUBLIC;
GRANT ALL ON TABLE khs TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 94 (OID 1218785)
-- Name: khs01; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khs01 (  
idkhs integer DEFAULT nextval("khs01_idkhs_seq"::text) NOT NULL,  
nomhs character varying(10) NOT NULL,  
jmtk smallint,  
jsks smallint,  
jmutu smallint,  
ipk real  
);

--
-- TOC entry 95 (OID 1218785)
-- Name: khs01; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khs01 FROM PUBLIC;
GRANT ALL ON TABLE khs01 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 15 (OID 1218788)
-- Name: khs01_idkhs_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE khs01_idkhs_seq  
INCREMENT BY 1  
MAXVALUE 2147483647  
NO MINVALUE  
CACHE 1;

--
-- TOC entry 16 (OID 1218788)
-- Name: khs01_idkhs_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khs01_idkhs_seq FROM PUBLIC;
GRANT ALL ON TABLE khs01_idkhs_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 96 (OID 1218790)
-- Name: khsdetil01; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil01 (  
...
nomhs character varying(10),
mreg character varying(4),
idmtk integer,
nilai character varying(1)
);

--
-- TOC entry 97 (OID 1218790)
-- Name: khsdetil01; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil01 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil01 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 98 (OID 1218792)
-- Name: khsdetil02; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil02 (nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1))
;

--
-- TOC entry 99 (OID 1218792)
-- Name: khsdetil02; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil02 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil02 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 100 (OID 1218794)
-- Name: khsdetil03; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil03 (nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1))
;

--
-- TOC entry 101 (OID 1218794)
-- Name: khsdetil03; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil03 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil03 TO ista;
SET SESSION AUTHORIZATION 'jack';

-- TOC entry 102 (OID 1218796)
-- Name: khsdetil04; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil04 (  
nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1)
);

-- TOC entry 103 (OID 1218796)
-- Name: khsdetil04; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil04 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil04 TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 104 (OID 1218798)
-- Name: khsdetil05; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil05 (  
nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1)
);

-- TOC entry 105 (OID 1218798)
-- Name: khsdetil05; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil05 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil05 TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 106 (OID 1218800)
-- Name: khsdetil06; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil06 (  
nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1)
);
-- TOC entry 107 (OID 1218800)
-- Name: khsdetil06; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil06 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil06 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 108 (OID 1218802)
-- Name: khsdetil07; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil07 (  
nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1)
);

--
-- TOC entry 109 (OID 1218802)
-- Name: khsdetil07; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil07 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil07 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 110 (OID 1218804)
-- Name: khsdetil08; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil08 (  
nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1)
);

--
-- TOC entry 111 (OID 1218804)
-- Name: khsdetil08; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil08 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil08 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 112 (OID 1218806)
-- Name: khsdetil09; Type: TABLE; Schema: public; Owner: jack
CREATE TABLE khsdetil09 (nomhs character varying(10) NOT NULL, mreg character varying(4) NOT NULL, idmtk integer NOT NULL, nilai character varying(1));

-- TOC entry 113 (OID 1218806)
-- Name: khsdetil09; Type: ACL; Schema: public; Owner: jack

REVOKE ALL ON TABLE khsdetil09 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil09 TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 114 (OID 1218808)
-- Name: khsdetil10; Type: TABLE; Schema: public; Owner: jack

CREATE TABLE khsdetil10 (nomhs character varying(10) NOT NULL, mreg character varying(4) NOT NULL, idmtk integer NOT NULL, nilai character varying(1));

-- TOC entry 115 (OID 1218808)
-- Name: khsdetil10; Type: ACL; Schema: public; Owner: jack

REVOKE ALL ON TABLE khsdetil10 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil10 TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 116 (OID 1218810)
-- Name: khsdetil11; Type: TABLE; Schema: public; Owner: jack

CREATE TABLE khsdetil11 (nomhs character varying(10) NOT NULL, mreg character varying(4) NOT NULL, idmtk integer NOT NULL, nilai character varying(1));

-- TOC entry 117 (OID 1218810)
-- Name: khsdetil11; Type: ACL; Schema: public; Owner: jack
REVOKE ALL ON TABLE khsdetil11 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil11 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 118 (OID 1218812)
-- Name: khsdetil31; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil31 (  
nomhs character varying(10) NOT NULL,  
mreg character varying(4) NOT NULL,  
idmtk integer NOT NULL,  
nilai character varying(1)  );

--
-- TOC entry 119 (OID 1218812)
-- Name: khsdetil31; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil31 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil31 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 120 (OID 1218814)
-- Name: khsdetil32; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil32 (  
nomhs character varying(10) NOT NULL,  
mreg character varying(4) NOT NULL,  
idmtk integer NOT NULL,  
nilai character varying(1)  );

--
-- TOC entry 121 (OID 1218814)
-- Name: khsdetil32; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil32 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil32 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 122 (OID 1218816)
-- Name: khsdetil33; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil33 (  
nomhs character varying(10) NOT NULL,  
mreg character varying(4) NOT NULL,  
idmtk integer NOT NULL,  
nilai character varying(1)  );
null character varying(1)
);

--
-- TOC entry 123 (OID 1218816)
-- Name: khsdetil33; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil33 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil33 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 124 (OID 1218818)
-- Name: khsdetil34; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil34 (nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1));

--
-- TOC entry 125 (OID 1218818)
-- Name: khsdetil34; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil34 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil34 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 126 (OID 1218820)
-- Name: khsdetil35; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE khsdetil35 (nomhs character varying(10) NOT NULL,
mreg character varying(4) NOT NULL,
idmtk integer NOT NULL,
nilai character varying(1));

--
-- TOC entry 127 (OID 1218820)
-- Name: khsdetil35; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE khsdetil35 FROM PUBLIC;
GRANT ALL ON TABLE khsdetil35 TO ista;

SET SESSION AUTHORIZATION 'jack';
CREATE TABLE kknkpta {
    jur character varying(2),
    idmtk integer,
    mtk character varying(3)
};

REVOKE ALL ON TABLE kknkpta FROM PUBLIC;
GRANT ALL ON TABLE kknkpta TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE kodeprop {
    kp character varying(2) NOT NULL,
    kode character varying(5) NOT NULL,
    arti character varying(30)
};

REVOKE ALL ON TABLE kodeprop FROM PUBLIC;
GRANT ALL ON TABLE kodeprop TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE konsenmhs {
    id integer DEFAULT nextval('"konsenmhs_id_seq"'::text) NOT NULL,
    nomhs character varying(15),
    idks integer
};

REVOKE ALL ON TABLE konsenmhs FROM PUBLIC;
GRANT ALL ON TABLE konsenmhs TO ista;

SET SESSION AUTHORIZATION 'jack';
REVOKE ALL ON TABLE konsenmhs FROM PUBLIC;
GRANT ALL ON TABLE konsenmhs TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 17 (OID 1218829)
-- Name: konsenmhs_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE konsenmhs_id_seq
   START WITH 1
   INCREMENT BY 1
   NO MAXVALUE
   NO MINVALUE
   CACHE 1;

-- TOC entry 18 (OID 1218829)
-- Name: konsenmhs_id_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE konsenmhs_id_seq FROM PUBLIC;
GRANT ALL ON TABLE konsenmhs_id_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 134 (OID 1218831)
-- Name: konsenstudi; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE konsenstudi (    id integer DEFAULT nextval('"konsenstudi_id_seq"'::text) NOT NULL,
   kode character varying(2),
   konsentrasi character varying(75)
);

-- TOC entry 135 (OID 1218831)
-- Name: konsenstudi; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE konsenstudi FROM PUBLIC;
GRANT ALL ON TABLE konsenstudi TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 19 (OID 1218834)
-- Name: konsenstudi_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE konsenstudi_id_seq
   INCREMENT BY 1
   NO MAXVALUE
   NO MINVALUE
CACHE 1;

-- TOC entry 20 (OID 1218834)
-- Name: konsenstudi_id_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE konsenstudi_id_seq FROM PUBLIC;
GRANT ALL ON TABLE konsenstudi_id_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 136 (OID 1218836)
-- Name: konsentrasi; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE konsentrasi (  
id integer DEFAULT nextval('konsentrasi_id_seq'::text) NOT NULL,
jur character varying(2) NOT NULL,
konsentrasi character varying(50) NOT NULL,
ket character varying(50),
sks integer,
urut integer
);

-- TOC entry 137 (OID 1218836)
-- Name: konsentrasi; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE konsentrasi FROM PUBLIC;
GRANT ALL ON TABLE konsentrasi TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 21 (OID 1218839)
-- Name: konsentrasi_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE konsentrasi_id_seq
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;

-- TOC entry 138 (OID 1218841)
-- Name: konversi; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE konversi (  
idkonversi integer DEFAULT nextval('konversi_idkonversi_seq'::text) NOT NULL,
idmtklama smallint,
idmtkbaru smallint,
kodemtklama character varying(10),
kodemtkbaru character varying(10)
);

--
-- TOC entry 139 (OID 1218841)
-- Name: konversi; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE konversi FROM PUBLIC;
GRANT ALL ON TABLE konversi TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 22 (OID 1218844)
-- Name: konversi_idkonversi_seq; Type: SEQUENCE; Schema: public;
Owner: jack
--
CREATE SEQUENCE konversi_idkonversi_seq
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;

--
-- TOC entry 23 (OID 1218844)
-- Name: konversi_idkonversi_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE konversi_idkonversi_seq FROM PUBLIC;
GRANT ALL ON TABLE konversi_idkonversi_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 140 (OID 1218846)
-- Name: konvmtk; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE konvmtk (id integer DEFAULT nextval('konvmtk_id_seq'::text) NOT NULL, idmtklama smallint, idmtkbaru text);

--
-- TOC entry 141 (OID 1218846)
-- Name: konvmtk; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE konvmtk FROM PUBLIC;
GRANT ALL ON TABLE konvmtk TO ista;

SET SESSION AUTHORIZATION 'jack';
CREATE SEQUENCE konvmtk_id_seq
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;

CREATE TABLE kpta (
  idkpta integer DEFAULT nextval('kpta_idkpta_seq'::text) NOT NULL,
  nomhs character varying(10) NOT NULL,
  judul text,
  pembimbing text,
  penguji text,
  tglsk date,
  nosk text,
  tglujian date,
  idmtk integer,
  mreg character varying(4),
  skripsi boolean DEFAULT false
);

CREATE SEQUENCE kpta_idkpta_seq
  START WITH 1
  INCREMENT BY 1
MAXVALUE 2147483647
NO MINVALUE
CACHE 1;

-- TOC entry 27 (OID 1218861)
-- Name: kpta_idkpta_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE kpta_idkpta_seq FROM PUBLIC;
GRANT ALL ON TABLE kpta_idkpta_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 144 (OID 1218863)
-- Name: kurikulum; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE kurikulum (  
idmtk integer DEFAULT nextval('kurikulum_idmtk_seq'::text) NOT NULL,  
jur character varying(6) NOT NULL,  
thkurik character varying(4) NOT NULL,  
kodemtk character varying(10) NOT NULL,  
matakuliah character varying(50),  
kredit smallint,  
smt smallint,  
sifat character varying(10),  
prasarat text,  
pengampu text,  
asing text,  
u integer,  
konsentrasi integer,  
kelompok character varying(4)  
);

--
-- TOC entry 145 (OID 1218863)
-- Name: kurikulum; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE kurikulum FROM PUBLIC;
GRANT ALL ON TABLE kurikulum TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 28 (OID 1218869)
-- Name: kurikulum_idmtk_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE kurikulum_idmtk_seq  
INCREMENT BY 1  
MAXVALUE 2147483647  
NO MINVALUE  
CACHE 1;
REVOKE ALL ON TABLE kurikulum_idmtk_seq FROM PUBLIC;
GRANT ALL ON TABLE kurikulum_idmtk_seq TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE maha (nomhs character varying(10), nirm character varying(25), nama character varying(50), kdosen character varying(6), jalur smallint, nouji character varying(20), jur character varying(6), ang character varying(4), konsentrasi integer, lulus timestamp with time zone);

CREATE TABLE mahasiswa (nomhs character varying(10) NOT NULL, nama character varying(50), tgllahir date, lelaki boolean, agama smallint, warga smallint, goldar character varying(2), kdosen character varying(10), nikah smallint, jalur smallint, kotalahir character varying(50), jur character varying(2), ang character varying(4), alamatyk character varying(100), kdpos character varying(10));

REVOKE ALL ON TABLE mahasiswa FROM PUBLIC;
GRANT ALL ON TABLE mahasiswa TO ista;
CREATE TABLE manytoone (  
idmtk integer,  
smt integer)
);

CREATE TABLE masaher (  
mreg character varying(4) NOT NULL,  
fileher text NOT NULL,  
aktive boolean DEFAULT false,  
tahun integer,  
dipakai character varying DEFAULT 'n'::character varying
);

CREATE TABLE mhs (  
nomhs character varying(10) NOT NULL,  
nirm character varying(25),  
nama character varying(50),  
kdosen character varying(6),  
jalur smallint,  
nouji character varying(20),  
jur character varying(6) NOT NULL,
ang character varying(4) NOT NULL,
konsentras integer,
lulus date,
tutupteori date,
status character varying(10)
);

--
-- TOC entry 154 (OID 1218884)
-- Name: mhs; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE mhs FROM PUBLIC;
GRANT ALL ON TABLE mhs TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 155 (OID 1218886)
-- Name: mhsdetil; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE mhsdetil (nomhs character varying(10) NOT NULL,
nsmta character varying(8),
jsmta character varying(10),
nosttb character varying(50),
tglsttb date,
mpsttb smallint,
jjminilll real,
mpnem smallint,
nem real,
namaortu character varying(70),
alamatortu character varying(100),
pekortu smallint,
penghasilan real,
pendidikanortu character varying(5),
telponortu character varying(15),
kabortu character varying(5),
proportu character varying(5),
kotasttb character varying(50),
kodeposortu character varying(10)
);

--
-- TOC entry 156 (OID 1218886)
-- Name: mhsdetil; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE mhsdetil FROM PUBLIC;
GRANT ALL ON TABLE mhsdetil TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 157 (OID 1218888)
-- Name: mhsdetil1; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE mhsdetil1 (  nomhs character varying(10) NOT NULL,  nsmta character varying(8),  jsmta character varying(10),  nosttb character varying(50),  tglsttb date,  mpsttb smallint,  jmlnilai real,  mpnem smallint,  nem real,  namaortu character varying(70),  alamatortu character varying(100),  pekortu smallint,  penghasilan real,  pendidikanortu character varying(5),  telponortu character varying(15),  kabortu character varying(5),  proortu character varying(5),  kotasttb character varying(50),  kodeposortu character varying(10) ) ;

--
-- TOC entry 158 (OID 1218888)
-- Name: mhsdetil1; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE mhsdetil1 FROM PUBLIC;
GRANT ALL ON TABLE mhsdetil1 TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 159 (OID 1218890)
-- Name: monitorkhs; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE monitorkhs (  id integer DEFAULT nextval('monitorkhs_id_seq'::text) NOT NULL,  nomhs character varying(15) NOT NULL,  idmtk integer,  iddetkrs integer,  nilai character varying(1),  mreg character varying(4) ) ;

--
-- TOC entry 160 (OID 1218890)
-- Name: monitorkhs; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE monitorkhs FROM PUBLIC;
GRANT ALL ON TABLE monitorkhs TO ista;
GRANT ALL ON TABLE monitorkhs TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 30 (OID 1218893)
CREATE SEQUENCE monitorkhs_id_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

CREATE TABLE mtkherta (
    jur character varying(2),
    idmtk integer
);

CREATE TABLE nomhsbaru (
    nomhs character varying(10) NOT NULL,
    nirm character varying(25),
    nama character varying(50),
    kdosen character varying(6),
    jalur smallint,
    nouji character varying(20),
    jur character varying(6) NOT NULL,
    ang character varying(4) NOT NULL,
    nobar character varying(10)
);
-- Name: nomhsbaru; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE nomhsbaru FROM PUBLIC;
GRANT ALL ON TABLE nomhsbaru TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 165 (OID 1218899)
-- Name: nomhsterakhir; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE nomhsterakhir (
   jurang character varying(4),
   maksi character varying(4)
);

--
-- TOC entry 166 (OID 1218901)
-- Name: pagesetup; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE pagesetup (
   sethal character varying(20) NOT NULL
);

--
-- TOC entry 167 (OID 1218901)
-- Name: pagesetup; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE pagesetup FROM PUBLIC;
GRANT ALL ON TABLE pagesetup TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 168 (OID 1218903)
-- Name: pegawai; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE pegawai (
   nik character varying(15) NOT NULL,
   nama character varying(50),
   tglmasuk date,
   statgol integer,
   pangkatid integer,
   pangkatgol character varying(10),
   edukatif character varying(3),
   pangkat character varying(30),
   unit character varying(50),
   pensiun integer
);

--
-- TOC entry 169 (OID 1218903)
-- Name: pegawai; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE pegawai FROM PUBLIC;
GRANT ALL ON TABLE pegawai TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE pengampu (  
jur character varying(2),
kode character varying(6),
idmtk integer
);

CREATE TABLE pga_forms (  
formname character varying(64),
formsource text
);

CREATE TABLE pga_layout (  
tablename character varying(64),
nrcols smallint,
colnames text,
colwidth text
);
REVOKE ALL ON TABLE pga_layout FROM PUBLIC;
GRANT ALL ON TABLE pga_layout TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE pga_queries (  
    queryname character varying(64),  
    querytype character(1),  
    querycommand text,  
    querytables text,  
    querylinks text,  
    queryresults text,  
    querycomments text
);

REVOKE ALL ON TABLE pga_queries FROM PUBLIC;
GRANT ALL ON TABLE pga_queries TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE pga_reports (  
    reportname character varying(64),  
    reportsource text,  
    reportbody text,  
    reportprocs text,  
    reportoptions text
);
SET SESSION AUTHORIZATION 'jack';

-- TOC entry 180 (OID 1218927)
-- Name: pga_schema; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE pga_schema (  schemaname character varying(64),  schematables text,  schemalinks text );

-- TOC entry 181 (OID 1218927)
-- Name: pga_schema; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE pga_schema FROM PUBLIC;
GRANT ALL ON TABLE pga_schema TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 182 (OID 1218932)
-- Name: pga_scripts; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE pga_scripts (  scriptname character varying(64),  scriptsource text );

-- TOC entry 183 (OID 1218932)
-- Name: pga_scripts; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE pga_scripts FROM PUBLIC;
GRANT ALL ON TABLE pga_scripts TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 184 (OID 1218937)
-- Name: pgset; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE pgset (  sethal character varying(20),  form character(1),  jbar smallint );

-- TOC entry 185 (OID 1218937)
-- Name: pgset; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE pgset FROM PUBLIC;
GRANT ALL ON TABLE pgset TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 186 (OID 1218939)
-- Name: praktikum; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE praktikum (  
idmtk integer NOT NULL  );

--
-- TOC entry 187 (OID 1218939)
-- Name: praktikum; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE praktikum FROM PUBLIC;
GRANT ALL ON TABLE praktikum TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 188 (OID 1218941)
-- Name: prodi; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE prodi (  
kode character varying(6) NOT NULL,  
fakultas character varying(50) NOT NULL,  
jurusan character varying(50) NOT NULL,  
pstudi character varying(50) NOT NULL,  
status character varying(15) NOT NULL,  
jenjang character varying(10) NOT NULL,  
kodefak character varying(4) NOT NULL,  
kajur character varying(50),  
slh character varying[],  
peringkat character varying(2),  
sk character varying(50),  
tglsk date,  
dekan character varying(6),  
kjur character varying(6),  
kprodi character varying(6),  
asalsk text,  
fac text,  
nip character varying(25),  
dep text,  
prodi text  );

--
-- TOC entry 189 (OID 1218941)
-- Name: prodi; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE prodi FROM PUBLIC;
GRANT ALL ON TABLE prodi TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 190 (OID 1218946)
-- Name: propinsi; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE propinsi(
    kode character varying(5) NOT NULL,
    arti character varying(30)
);

--
-- TOC entry 191 (OID 1218946)
-- Name: propinsi; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE propinsi FROM PUBLIC;
GRANT ALL ON TABLE propinsi TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 192 (OID 1218948)
-- Name: pt; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE pt(
    nama text NOT NULL,
    alamat text,
    kota text,
    email text,
    phone text,
    website text
);

--
-- TOC entry 193 (OID 1218948)
-- Name: pt; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE pt FROM PUBLIC;
GRANT ALL ON TABLE pt TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 194 (OID 1218953)
-- Name: ruang; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE ruang(
    kode character varying(5) NOT NULL,
    dtp smallint,
    ket text
);
REVOKE ALL ON TABLE ruang FROM PUBLIC;
GRANT ALL ON TABLE ruang TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE smta (
kode character varying(8),
nama text,
alamat text,
jnsmta character varying(1)
);

REVOKE ALL ON TABLE smta FROM PUBLIC;
GRANT ALL ON TABLE smta TO ista;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE statdos (
kode character varying(6),
kg smallint,
goldos character varying(10),
pangkat character varying(50)
);

REVOKE ALL ON TABLE statdos FROM PUBLIC;
GRANT ALL ON TABLE statdos TO ista;

SET SESSION AUTHORIZATION 'jack';
nomhs character varying(10) NOT NULL,
nama character varying(50),
tgllahir date,
lelaki boolean,
agama smallint,
warga smallint,
goldar character varying(2),
kdosen character varying(10),
nikah smallint,
jalur smallint,
kotalahir character varying(50),
jur character varying(2),
ang character varying(4),
alamatyk character varying(100),
kpos character varying(10)
);

--
-- TOC entry 206 (OID 1218971)
-- Name: tempmahasiswa; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE tempmahasiswa FROM PUBLIC;
GRANT ALL ON TABLE tempmahasiswa TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 207 (OID 1218973)
-- Name: tempnomhs; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE tempnomhs ( nomhs character varying(10), jur character varying(2) );

--
-- TOC entry 208 (OID 1218973)
-- Name: tempnomhs; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE tempnomhs FROM PUBLIC;
GRANT ALL ON TABLE tempnomhs TO ista;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 209 (OID 1218975)
-- Name: unit; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE unit ( namaunit character varying(50) NOT NULL, ka character varying(6) );

--
-- TOC entry 210 (OID 1218975)
-- Name: unit; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE unit FROM PUBLIC;
GRANT ALL ON TABLE unit TO ista;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 211 (OID 1218977)
-- Name: utangnil; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE utangnil {
  mreg character varying(4),
  kode character varying(6),
  idklas integer
};

-- TOC entry 212 (OID 1218979)
-- Name: xx; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE xx {
  "iddetkrs " integer,
  idkrs integer,
  idkelas integer
};

-- TOC entry 237 (OID 1976634)
-- Name: khsdetil_nomhs_mreg_idmtk; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX khsdetil_nomhs_mreg_idmtk ON khsdetil01 USING btree
  (nomhs, mreg, idmtk);

-- TOC entry 254 (OID 1976635)
-- Name: konsenstudi_id_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX konsenstudi_id_key ON konsenstudi USING btree
  (id);

-- TOC entry 255 (OID 1976636)
-- Name: konversi_idkonversi_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX konversi_idkonversi_key ON konversi USING btree
  (idkonversi);
CREATE UNIQUE INDEX konvmtk_id_key ON konvmtk USING btree (id);

CREATE UNIQUE INDEX kurikulum_idmtk_key ON kurikulum USING btree (idmtk);

ALTER TABLE ONLY biayapendek
    ADD CONSTRAINT biayapendek_pkey PRIMARY KEY (mreg);

ALTER TABLE ONLY detkhs01
    ADD CONSTRAINT detkhs01_pkey PRIMARY KEY (iddetkhs);

ALTER TABLE ONLY detkhs
    ADD CONSTRAINT detkhs_pkey PRIMARY KEY (idkhs);

ALTER TABLE ONLY dkhs01
    ADD CONSTRAINT dkhs01_pkey PRIMARY KEY (nomhs, idmtk);

ALTER TABLE ONLY dkhs02
    ADD CONSTRAINT dkhs02_pkey PRIMARY KEY (nomhs, idmtk);
-- TOC entry 218 (OID 1976534)
-- Name: dkhs03_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY dkhs03
  ADD CONSTRAINT dkhs03_pkey PRIMARY KEY (nomhs, idmtk);

--
-- TOC entry 219 (OID 1976536)
-- Name: dkhs04_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY dkhs04
  ADD CONSTRAINT dkhs04_pkey PRIMARY KEY (nomhs, idmtk);

--
-- TOC entry 220 (OID 1976538)
-- Name: dkhs05_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY dkhs05
  ADD CONSTRAINT dkhs05_pkey PRIMARY KEY (nomhs, idmtk);

--
-- TOC entry 221 (OID 1976540)
-- Name: dkhs06_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY dkhs06
  ADD CONSTRAINT dkhs06_pkey PRIMARY KEY (nomhs, idmtk);

--
-- TOC entry 222 (OID 1976542)
-- Name: dkhs07_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY dkhs07
  ADD CONSTRAINT dkhs07_pkey PRIMARY KEY (nomhs, idmtk);

--
-- TOC entry 223 (OID 1976544)
-- Name: dosen_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY dosen
  ADD CONSTRAINT dosen_pkey PRIMARY KEY (kode);

--
-- TOC entry 224 (OID 1976546)
-- Name: dosenjur_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY dosenjur
  ADD CONSTRAINT dosenjur_pkey PRIMARY KEY (kddosen);
ALTER TABLE ONLY fakultas
    ADD CONSTRAINT fakultas_pkey PRIMARY KEY (kodefak);

ALTER TABLE ONLY fasilitas
    ADD CONSTRAINT fasilitas_pkey PRIMARY KEY ("level");

ALTER TABLE ONLY hari
    ADD CONSTRAINT hari_pkey PRIMARY KEY (idhari);

ALTER TABLE ONLY honor
    ADD CONSTRAINT honor_pkey PRIMARY KEY (mreg);

ALTER TABLE ONLY identitas
    ADD CONSTRAINT identitas_pkey PRIMARY KEY (id);

ALTER TABLE ONLY jalur
    ADD CONSTRAINT jalur_pkey PRIMARY KEY (id);

ALTER TABLE ONLY jam
    ADD CONSTRAINT jam_pkey PRIMARY KEY (jamke);
-- TOC entry 232 (OID 1976562)
-- Name: jenisher_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY jenisher
    ADD CONSTRAINT jenisher_pkey PRIMARY KEY (id);

--
-- TOC entry 233 (OID 1976564)
-- Name: jeniskpta_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY jeniskpta
    ADD CONSTRAINT jeniskpta_pkey PRIMARY KEY (kode);

--
-- TOC entry 234 (OID 1976566)
-- Name: jsmta_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY jsmta
    ADD CONSTRAINT jsmta_pkey PRIMARY KEY (kode);

--
-- TOC entry 236 (OID 1976568)
-- Name: khs01_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY khs01
    ADD CONSTRAINT khs01_pkey PRIMARY KEY (idkhs);

--
-- TOC entry 235 (OID 1976570)
-- Name: khs_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY khs
    ADD CONSTRAINT khs_pkey PRIMARY KEY (nomhs);

--
-- TOC entry 238 (OID 1976572)
-- Name: khsdetil102_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY khsdetil102
    ADD CONSTRAINT khsdetil102_pkey PRIMARY KEY (nomhs, mreg, idmtk);

--
-- TOC entry 239 (OID 1976574)
-- Name: khsdetil103_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY khsdetil103
    ADD CONSTRAINT khsdetil103_pkey PRIMARY KEY (nomhs, mreg, idmtk);

--
ALTER TABLE ONLY khsdetil04
    ADD CONSTRAINT khsdetil04_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil05
    ADD CONSTRAINT khsdetil05_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil06
    ADD CONSTRAINT khsdetil06_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil07
    ADD CONSTRAINT khsdetil07_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil08
    ADD CONSTRAINT khsdetil08_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil09
    ADD CONSTRAINT khsdetil09_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil10
    ADD CONSTRAINT khsdetil10_pkey PRIMARY KEY (nomhs, mreg, idmtk);
ALTER TABLE ONLY khsdetil11
  ADD CONSTRAINT khsdetil11_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil31
  ADD CONSTRAINT khsdetil31_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil32
  ADD CONSTRAINT khsdetil32_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil33
  ADD CONSTRAINT khsdetil33_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil34
  ADD CONSTRAINT khsdetil34_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY khsdetil35
  ADD CONSTRAINT khsdetil35_pkey PRIMARY KEY (nomhs, mreg, idmtk);

ALTER TABLE ONLY kodeprop
  ADD CONSTRAINT kodeprop_pkey PRIMARY KEY (kode);
-- TOC entry 257 (OID 1976604)
-- Name: kpta_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY kpta
  ADD CONSTRAINT kpta_pkey PRIMARY KEY (idkpta);

-- TOC entry 259 (OID 1976606)
-- Name: mahasiswa_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY mahasiswa
  ADD CONSTRAINT mahasiswa_pkey PRIMARY KEY (nomhs);

-- TOC entry 260 (OID 1976608)
-- Name: masaher_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY masaher
  ADD CONSTRAINT masaher_pkey PRIMARY KEY (mreg);

-- TOC entry 261 (OID 1976610)
-- Name: mhs_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY mhs
  ADD CONSTRAINT mhs_pkey PRIMARY KEY (nomhs);

-- TOC entry 262 (OID 1976612)
-- Name: mhsdetil_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY mhsdetil
  ADD CONSTRAINT mhsdetil_pkey PRIMARY KEY (nomhs);

-- TOC entry 263 (OID 1976614)
-- Name: nomhsbaru_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY nomhsbaru
  ADD CONSTRAINT nomhsbaru_pkey PRIMARY KEY (nomhs);

-- TOC entry 264 (OID 1976616)
-- Name: pagesetup_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY pagesetup
  ADD CONSTRAINT pagesetup_pkey PRIMARY KEY (sethal);
ALTER TABLE ONLY pegawai
    ADD CONSTRAINT pegawai_pkey PRIMARY KEY (nik);

ALTER TABLE ONLY praktikum
    ADD CONSTRAINT praktikum_pkey PRIMARY KEY (idmtk);

ALTER TABLE ONLY prodi
    ADD CONSTRAINT prodi_pkey PRIMARY KEY (kode);

ALTER TABLE ONLY propinsi
    ADD CONSTRAINT propinsi_pkey PRIMARY KEY (kode);

ALTER TABLE ONLY ruang
    ADD CONSTRAINT ruang_pkey PRIMARY KEY (kode);

ALTER TABLE ONLY takp
    ADD CONSTRAINT takp_pkey PRIMARY KEY (kode);

ALTER TABLE ONLY tempmahasiswa
    ADD CONSTRAINT tempmahasiswa_pkey PRIMARY KEY (nomhs);
ALTER TABLE ONLY unit
    ADD CONSTRAINT unit_pkey PRIMARY KEY (namaunit);

SET SESSION AUTHORIZATION 'postgres';

COMMENT ON SCHEMA public IS 'Standard public schema';

Plain Text Attachment [ Scan and Save to Computer ]

SET client_encoding = 'LATIN1';
SET check_function_bodies = false;

SET SESSION AUTHORIZATION 'postgres';

REVOKE ALL ON SCHEMA public FROM PUBLIC;
GRANT ALL ON SCHEMA public TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
SET search_path = public, pg_catalog;

CREATE TABLE pga_queries (
    queryname character varying(64),
    querytype character(1),
    querycommand text,
    querytables text,
    querylinks text,
    queryresults text,
    querycomments text
);

CREATE TABLE pga_queries (queryname character varying(64), querytype character(1), querycommand text, querytables text, querylinks text, queryresults text, querycomments text);
REVOKE ALL ON TABLE pga_queries FROM PUBLIC;
GRANT ALL ON TABLE pga_queries TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';  

--  
-- TOC entry 63 (OID 4888017)  
-- Name: pga_forms; Type: TABLE; Schema: public; Owner: jack  
--  
CREATE TABLE pga_forms (
    formname character varying(64),
    formsource text
);  

--  
-- TOC entry 64 (OID 4888017)  
-- Name: pga_forms; Type: ACL; Schema: public; Owner: jack  
--  
REVOKE ALL ON TABLE pga_forms FROM PUBLIC;
GRANT ALL ON TABLE pga_forms TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';  

--  
-- TOC entry 65 (OID 4888022)  
-- Name: pga_scripts; Type: TABLE; Schema: public; Owner: jack  
--  
CREATE TABLE pga_scripts (
    scriptname character varying(64),
    scriptsource text
);  

--  
-- TOC entry 66 (OID 4888022)  
-- Name: pga_scripts; Type: ACL; Schema: public; Owner: jack  
--  
REVOKE ALL ON TABLE pga_scripts FROM PUBLIC;
GRANT ALL ON TABLE pga_scripts TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';  

--  
-- TOC entry 67 (OID 4888027)  
-- Name: pga_reports; Type: TABLE; Schema: public; Owner: jack  
--  
CREATE TABLE pga_reports (
    reportname character varying(64),
    reportsource text,
    reportbody text,
    reportprocs text,
    reportoptions text
);
REVOKE ALL ON TABLE pga_reports FROM PUBLIC;
GRANT ALL ON TABLE pga_reports TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE pga_schema (
    schemaname character varying(64),
    schematables text,
    schemalinks text
);

REVOKE ALL ON TABLE pga_schema FROM PUBLIC;
GRANT ALL ON TABLE pga_schema TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE pga_layout (
    tablename character varying(64),
    nrcols smallint,
    colnames text,
    colwidth text
);

REVOKE ALL ON TABLE pga_layout FROM PUBLIC;
GRANT ALL ON TABLE pga_layout TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 73 (OID 4888042)
-- Name: her; Type: TABLE; Schema: public; Owner: jack
CREATE TABLE her (  idher integer DEFAULT nextval('her_idher_seq'::text) NOT NULL,  mreg character varying(4) NOT NULL,  nomhs character varying(10) NOT NULL,  tglher date,  jher character varying(1),  tglherbaa date,  fotoktm character(1),  ktmjadi date,  ktambil date);

--
-- TOC entry 74 (OID 4888042)
-- Name: her; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE her FROM PUBLIC;
GRANT ALL ON TABLE her TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 75 (OID 4888045)
-- Name: tawar; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tawar (  idtawar integer DEFAULT nextval('tawar_idtawar_seq'::text) NOT NULL,  mreg character varying(4) NOT NULL,  idmtk integer NOT NULL,  nklas smallint DEFAULT 1 NOT NULL,  peserta integer,  setuju boolean,  x integer);

--
-- TOC entry 76 (OID 4888045)
-- Name: tawar; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tawar FROM PUBLIC;
GRANT ALL ON TABLE tawar TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 77 (OID 4888049)
-- Name: klas; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE klas (  idklas integer DEFAULT nextval('klas_idklas_seq'::text) NOT NULL,  idtawar integer NOT NULL,  klas character(1) NOT NULL,  ruang character varying(10),


dtp smallint,
jml smallint,
kdosen character varying(24),
kasisten character varying(24),
isi boolean,
tampil boolean
);

--
-- TOC entry 78 (OID 4888049)
-- Name: klas; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE klas FROM PUBLIC;
GRANT ALL ON TABLE klas TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 79 (OID 4888052)
-- Name: detkrs; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE detkrs (  
  iddetkrs integer DEFAULT nextval('detkrs_iddetkrs_seq '::text) NOT NULL,
  idkrs integer NOT NULL,
  idkelas integer NOT NULL,
  tugas character varying(4),
  mid character varying(4),
  sem character varying(4),
  akhir character(1),
  iduji smallint,
  idumid smallint
);

--
-- TOC entry 80 (OID 4888052)
-- Name: detkrs; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE detkrs FROM PUBLIC;
GRANT ALL ON TABLE detkrs TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 81 (OID 4888055)
-- Name: krs; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE krs (  
  idkrs integer DEFAULT nextval('krs_idkrs_seq '::text) NOT NULL,
  idher integer NOT NULL,
  jsks smallint,
  jmtk smallint,
  ip real,
  lunas boolean DEFAULT false,
  posted date,
acc boolean DEFAULT false,
tglacc date,
ketacc text,
tgldaf date,
bsks smallint,
kdosen character varying(6),
smt integer,
jmutu smallint
);

--
-- TOC entry 82 (OID 4888055)
-- Name: krs; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE krs FROM PUBLIC;
GRANT ALL ON TABLE krs TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 83 (OID 4888063)
-- Name: prodi; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE prodi (
    kode character varying(6) NOT NULL,
fakultas character varying(50) NOT NULL,
jurusan character varying(50) NOT NULL,
pstudi character varying(50) NOT NULL,
status character varying(15) NOT NULL,
jenjang character varying(10) NOT NULL,
kodefak character varying(4) NOT NULL
);

--
-- TOC entry 84 (OID 4888063)
-- Name: prodi; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE prodi FROM PUBLIC;
GRANT ALL ON TABLE prodi TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 85 (OID 4888065)
-- Name: dosen; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE dosen (
    kode character varying(6) NOT NULL,
nama character varying(50),
gelar1 character varying(25),
gelar2 character varying(25),
nodos character varying(25),
kodefak character varying(4),
goldos character varying(10),
jabdos character varying(25),
)
jsks smallint,  
jklas smallint,  
jgabung smallint,  
status character varying(4)  
);

--  
-- TOC entry 86 (OID 4888065)  
-- Name: dosen; Type: ACL; Schema: public; Owner: jack  
--  
REVOKE ALL ON TABLE dosen FROM PUBLIC;  
GRANT ALL ON TABLE dosen TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';  

--  
-- TOC entry 87 (OID 4888067)  
-- Name: ajar; Type: TABLE; Schema: public; Owner: jack  
--  
CREATE TABLE ajar (  
idajar integer DEFAULT nextval('ajar_idajar_seq'::text) NOT NULL,  
idklas integer NOT NULL,  
kode character varying(6) NOT NULL,  
dosen boolean DEFAULT true,  
gabung text,  
setuju boolean,  
ket character varying(15)  
);

--  
-- TOC entry 88 (OID 4888067)  
-- Name: ajar; Type: ACL; Schema: public; Owner: jack  
--  
REVOKE ALL ON TABLE ajar FROM PUBLIC;  
GRANT ALL ON TABLE ajar TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';  

--  
-- TOC entry 89 (OID 4888074)  
-- Name: jam; Type: TABLE; Schema: public; Owner: jack  
--  
CREATE TABLE jam (  
jamke smallint NOT NULL,  
jam character varying(5) NOT NULL,  
isi boolean DEFAULT false  
);  

--  
-- TOC entry 90 (OID 4888074)  
-- Name: jam; Type: ACL; Schema: public; Owner: jack  
--  
REVOKE ALL ON TABLE jam FROM PUBLIC;
GRANT ALL ON TABLE jam TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 91 (OID 4888077)
-- Name: hari; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE hari (
    idhari smallint NOT NULL,
    namahari character varying(15) NOT NULL
);

--
-- TOC entry 92 (OID 4888077)
-- Name: hari; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE hari FROM PUBLIC;
GRANT ALL ON TABLE hari TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 93 (OID 4888079)
-- Name: kuliah; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE kuliah (
    idkuliah integer DEFAULT nextval('kuliah_idkuliah_seq'::text) NOT NULL,
    idkelas integer NOT NULL,
    idharijam integer NOT NULL,
    ket text,
    idajar integer
);

--
-- TOC entry 94 (OID 4888079)
-- Name: kuliah; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE kuliah FROM PUBLIC;
GRANT ALL ON TABLE kuliah TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 95 (OID 4888085)
-- Name: harijamruang; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE harijamruang (
    idharijam integer DEFAULT nextval('harijamruang_idharijam_seq'::text) NOT NULL,
    idhari smallint NOT NULL,
    namahari character varying(15) NOT NULL,
BEGIN;

CREATE TABLE harijamruang (
    jamke smallint NOT NULL,
    jam character varying(5) NOT NULL,
    isi boolean DEFAULT false,
    kode character varying(5) NOT NULL,
    dtp smallint NOT NULL,
    ket text
);

--
-- TOC entry 96 (OID 4888085)
-- Name: harijamruang; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE harijamruang FROM PUBLIC;
GRANT ALL ON TABLE harijamruang TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 97 (OID 4888092)
-- Name: pt; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE pt (
    nama text NOT NULL,
    alamat text,
    kota text,
    email text,
    phone text,
    website text
);

--
-- TOC entry 98 (OID 4888092)
-- Name: pt; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE pt FROM PUBLIC;
GRANT ALL ON TABLE pt TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 99 (OID 4888097)
-- Name: fakultas; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE fakultas (
    kodefak character varying(4) NOT NULL,
    fakultas text NOT NULL
);

--
-- TOC entry 100 (OID 4888097)
-- Name: fakultas; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE fakultas FROM PUBLIC;
GRANT ALL ON TABLE fakultas TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 101 (OID 4888102)
-- Name: mahasisw; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE mahasisw (nomhs character varying(10) NOT NULL, nirm character varying(25), nama character varying(50), kdosen character varying(6), jalur smallint, nouji character varying(20), jur character varying(6) NOT NULL, ang character varying(4) NOT NULL);

--
-- TOC entry 102 (OID 4888102)
-- Name: mahasisw; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE mahasisw FROM PUBLIC;
GRANT ALL ON TABLE mahasisw TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 103 (OID 4888104)
-- Name: matrikujian; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE matrikujian (idtgljamruang integer DEFAULT nextval('matrikujian_idtgljamruang_seq'::text) NOT NULL, kode character varying(5), tglke smallint, hari character varying(6), tanggal character varying(20), dtpu smallint, isi boolean, jamke smallint, jam character varying(5))
;

--
-- TOC entry 104 (OID 4888104)
-- Name: matrikujian; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE matrikujian FROM PUBLIC;
GRANT ALL ON TABLE matrikujian TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
CREATE TABLE klsuji (  iduji integer DEFAULT nextval('klsuji_iduji_seq':text) NOT NULL,  idtawar integer NOT NULL,  idtgljamruang integer NOT NULL,  kode character varying(5) NOT NULL,  hari character varying(6) NOT NULL,  tanggal character varying(20) NOT NULL,  jam character varying(5) NOT NULL,  dtpu smallint NOT NULL,  dosen character varying(5),  jpes smallint)

CREATE TABLE tanggalujian (  tglke smallint NOT NULL,  tglujian character varying(17),  hari character varying(6))

CREATE TABLE ngajar (  idajar integer NOT NULL,  idtawar integer,  idklas integer NOT NULL,  kode character varying(5) NOT NULL,
dosen boolean,
   gabung text,
   setuju boolean,
   ket text
);

--
-- TOC entry 110 (OID 4888112)
-- Name: ngajar; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE ngajar FROM PUBLIC;
GRANT ALL ON TABLE ngajar TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 111 (OID 4888117)
-- Name: tempuji01; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji01 (
   iddetkrs integer,
   idkrs integer
);

--
-- TOC entry 112 (OID 4888117)
-- Name: tempuji01; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji01 FROM PUBLIC;
GRANT ALL ON TABLE tempuji01 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 113 (OID 4888119)
-- Name: tempuji02; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji02 (
   iddetkrs integer,
   idkrs integer
);

--
-- TOC entry 114 (OID 4888119)
-- Name: tempuji02; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji02 FROM PUBLIC;
GRANT ALL ON TABLE tempuji02 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
CREATE TABLE tempuji03 (  iddetkrs integer,  idkrs integer);

REVOKE ALL ON TABLE tempuji03 FROM PUBLIC;
GRANT ALL ON TABLE tempuji03 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE tempuji04 (  iddetkrs integer,  idkrs integer);

REVOKE ALL ON TABLE tempuji04 FROM PUBLIC;
GRANT ALL ON TABLE tempuji04 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE tempuji05 (  iddetkrs integer,  idkrs integer);

REVOKE ALL ON TABLE tempuji05 FROM PUBLIC;
GRANT ALL ON TABLE tempuji05 TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 121 (OID 4888127)
-- Name: tempuji06; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji06
  (iddetkrs integer,
   idkrs integer);

--
-- TOC entry 122 (OID 4888127)
-- Name: tempuji06; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji06 FROM PUBLIC;
GRANT ALL ON TABLE tempuji06 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 123 (OID 4888129)
-- Name: tempuji07; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji07
  (iddetkrs integer,
   idkrs integer);

--
-- TOC entry 124 (OID 4888129)
-- Name: tempuji07; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji07 FROM PUBLIC;
GRANT ALL ON TABLE tempuji07 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 125 (OID 4888131)
-- Name: tempuji08; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji08
  (iddetkrs integer,
   idkrs integer);

--
-- TOC entry 126 (OID 4888131)
-- Name: tempuji08; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji08 FROM PUBLIC;
GRANT ALL ON TABLE tempuji08 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 127 (OID 4888133)
-- Name: tempuji09; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji09 {
    iddetkrs integer,
    idkrs integer
};

--
-- TOC entry 128 (OID 4888133)
-- Name: tempuji09; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji09 FROM PUBLIC;
GRANT ALL ON TABLE tempuji09 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 129 (OID 4888135)
-- Name: tempuji10; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji10 {
    iddetkrs integer,
    idkrs integer
};

--
-- TOC entry 130 (OID 4888135)
-- Name: tempuji10; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji10 FROM PUBLIC;
GRANT ALL ON TABLE tempuji10 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 131 (OID 4888137)
-- Name: tempuji31; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji31 {
    iddetkrs integer,
    idkrs integer
};

--
-- TOC entry 132 (OID 4888137)
-- Name: tempuji31; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE tempuji31 FROM PUBLIC;
GRANT ALL ON TABLE tempuji31 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 133 (OID 4888139)
-- Name: tempuji32; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE tempuji32 (  
  iddetkrs integer,  
  idkrs integer
);

--
-- TOC entry 134 (OID 4888139)
-- Name: tempuji32; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE tempuji32 FROM PUBLIC;
GRANT ALL ON TABLE tempuji32 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 135 (OID 4888141)
-- Name: tempuji33; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE tempuji33 (  
  iddetkrs integer,  
  idkrs integer
);

--
-- TOC entry 136 (OID 4888141)
-- Name: tempuji33; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE tempuji33 FROM PUBLIC;
GRANT ALL ON TABLE tempuji33 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 137 (OID 4888143)
-- Name: tempuji34; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE tempuji34 (  
  iddetkrs integer,  
  idkrs integer
);
-- TOC entry 138 (OID 4888143)
-- Name: tempuji34; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji34 FROM PUBLIC;
GRANT ALL ON TABLE tempuji34 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 139 (OID 4888145)
-- Name: tempuji35; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempuji35 (
    iddetkrs integer,
    idkrs integer
);

-- TOC entry 140 (OID 4888145)
-- Name: tempuji35; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempuji35 FROM PUBLIC;
GRANT ALL ON TABLE tempuji35 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 141 (OID 4888147)
-- Name: jamujian; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE jamujian (
    jamke smallint NOT NULL,
    jam character varying(5) NOT NULL
);

-- TOC entry 142 (OID 4888147)
-- Name: jamujian; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE jamujian FROM PUBLIC;
GRANT ALL ON TABLE jamujian TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 143 (OID 4888149)
-- Name: interip; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE interip (
    ips real NOT NULL,
    sksmaks smallint NOT NULL
)
REVOKE ALL ON TABLE interip FROM PUBLIC;
GRANT ALL ON TABLE interip TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 145 (OID 4888151)
-- Name: kurikulum; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE kurikulum (idmtk integer DEFAULT nextval('kurikulum_idmtk_seq'::text) NOT NULL,
jur character varying(6) NOT NULL,
thurik character varying(4) NOT NULL,
kodemtk character varying(10) NOT NULL,
matakuliah character varying(50),
kredit smallint,
smt smallint,
sifat character varying(10),
prasarat text,
pengampu text);

--
-- TOC entry 146 (OID 4888151)
-- Name: kurikulum; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE kurikulum FROM PUBLIC;
GRANT ALL ON TABLE kurikulum TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 147 (OID 4888157)
-- Name: urutmtk; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE urutmtk (idurutmtk integer DEFAULT nextval('urutmtk_idurutmtk_seq'::text) NOT NULL,
u integer,
idmtk integer,
matakuliah character varying(50),
jur character varying(10),
sks integer)

--
-- TOC entry 148 (OID 4888157)
-- Name: urutmtk; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE urutmtk FROM PUBLIC;
GRANT ALL ON TABLE urutmtk TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
--
-- TOC entry 149 (OID 4888160)
-- Name: jumlahsks; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE jumlahsks (  
nomhs character varying(20) NOT NULL,  
nama character varying(50) NOT NULL,  
keterangan text,  
jsks integer,  
jmtk integer,  
jur character varying(10)
);

--
-- TOC entry 150 (OID 4888160)
-- Name: jumlahsks; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE jumlahsks FROM PUBLIC;
GRANT ALL ON TABLE jumlahsks TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
--
-- TOC entry 151 (OID 4888165)
-- Name: xx; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE xx (  
iddetkrs character varying(20),  
idkrs character varying(10),  
idkelas character varying(20),  
klas character varying(1)
);

--
-- TOC entry 152 (OID 4888165)
-- Name: xx; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE xx FROM PUBLIC;
GRANT ALL ON TABLE xx TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
--
-- TOC entry 153 (OID 4888167)
-- Name: nilaibelum; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE nilaibelum (  
idtawar smallint,  
idkelas smallint
);

--  
-- TOC entry 154 (OID 4888167)  
-- Name: nilaibelum; Type: ACL; Schema: public; Owner: jack  
--

REVOKE ALL ON TABLE nilaibelum FROM PUBLIC;  
GRANT ALL ON TABLE nilaibelum TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--  
-- TOC entry 155 (OID 4888169)  
-- Name: jadwal; Type: TABLE; Schema: public; Owner: jack  
--

CREATE TABLE jadwal (  
kode character varying(10),  
matakuliah character varying(50),  
sks smallint,  
kelas character varying(1),  
hari character varying(15),  
awal character varying(5),  
akhir character varying(5),  
ruang character varying(5),  
dosen character varying(60)
);

--  
-- TOC entry 156 (OID 4888169)  
-- Name: jadwal; Type: ACL; Schema: public; Owner: jack  
--

REVOKE ALL ON TABLE jadwal FROM PUBLIC;  
GRANT ALL ON TABLE jadwal TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--  
-- TOC entry 157 (OID 4888171)  
-- Name: jamkuliah; Type: TABLE; Schema: public; Owner: jack  
--

CREATE TABLE jamkuliah (  
awal character varying(5),  
akhir character varying(5),  
sks smallint
);

--  
-- TOC entry 158 (OID 4888171)  
-- Name: jamkuliah; Type: ACL; Schema: public; Owner: jack  
--
REVOKE ALL ON TABLE jamkuliah FROM PUBLIC;
GRANT ALL ON TABLE jamkuliah TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 159 (OID 4888173)
-- Name: abdos; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE abdos (  
  kode character varying(6),
  idkuliah integer,
  tgl character varying(10),
  hadir boolean,
  tglisi date,
  materi text
);

--
-- TOC entry 160 (OID 4888173)
-- Name: abdos; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE abdos FROM PUBLIC;
GRANT ALL ON TABLE abdos TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 161 (OID 4888178)
-- Name: matdos2; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE matdos2 (  
  kode character varying(4),
  idkuliah integer,
  tgl date,
  tglisi date,
  materi text,
  pertemuanke integer,
  judul text,
  jmlmhs integer,
  jam character varying(11)
);

--
-- TOC entry 162 (OID 4888178)
-- Name: matdos2; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE matdos2 FROM PUBLIC;
GRANT ALL ON TABLE matdos2 TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
CREATE TABLE yangher (  nomhs character varying(10),  mreg character varying(4),  nama character varying(50),  jur character varying(2),  ang character varying(4),  jher smallint,  jenis character varying(30) );

REVOKE ALL ON TABLE yangher FROM PUBLIC;
GRANT ALL ON TABLE yangher TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE abmhs (  idkuliah integer,  nomhs character varying(10),  tanggal date,  idtawar integer  );

REVOKE ALL ON TABLE abmhs FROM PUBLIC;
GRANT ALL ON TABLE abmhs TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE dether (  iddether integer DEFAULT nextval('"dether_iddether_seq"'::text) NOT NULL,  nomhs character varying(10),  jher smallint,  nominal real,  tglher date  );
CREATE TABLE rekapnilai (  
mreg character varying(4) NOT NULL,  
kodej character varying(6) NOT NULL,  
tahunke smallint,  
kodemtk character varying(10) NOT NULL,  
matakuliah character varying(50) NOT NULL,  
kredit smallint,  
ja smallint,  
jb smallint,  
jc smallint,  
jd smallint,  
je smallint,  
jn smallint,  
jumlah smallint,  
dosen character varying(50)  
);
REVOKE ALL ON TABLE ijinpakai FROM PUBLIC;
GRANT ALL ON TABLE ijinpakai TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 5 (OID 4888194)
-- Name: harijamruang_idharijam_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE harijamruang_idharijam_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

-- TOC entry 6 (OID 4888194)
-- Name: harijamruang_idharijam_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE harijamruang_idharijam_seq FROM PUBLIC;
GRANT ALL ON TABLE harijamruang_idharijam_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 173 (OID 4888196)
-- Name: tempujill; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempujill (
    iddetkrs integer NOT NULL,
    idkrs integer NOT NULL
);

-- TOC entry 174 (OID 4888196)
-- Name: tempujill; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempujill FROM PUBLIC;
GRANT ALL ON TABLE tempujill TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 175 (OID 4888198)
-- Name: ruang; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE ruang (
    kode character varying(5) NOT NULL,
    dtp smallint,
CREATE TABLE tempbeban (
    kode character varying(6) NOT NULL,
    nama character varying(50),
    gelar1 character varying(25),
    gelar2 character varying(25),
    jsks smallint,
    jklas smallint,
    jgabung smallint,
    jbina smallint,
    beban smallint
);

--
-- TOC entry 177 (OID 4888203)
-- Name: tempbeban; Type: TABLE; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempbeban FROM PUBLIC;
GRANT ALL ON TABLE tempbeban TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 7 (OID 4888205)
-- Name: ajar_idajar_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE ajar_idajar_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

--
-- TOC entry 8 (OID 4888205)
-- Name: ajar_idajar_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE ajar_idajar_seq FROM PUBLIC;
GRANT ALL ON TABLE ajar_idajar_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 9 (OID 4888207)
-- Name: her_idher_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE her_idher_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

--
-- TOC entry 10 (OID 4888207)
-- Name: her_idher_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE her_idher_seq FROM PUBLIC;
GRANT ALL ON TABLE her_idher_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 11 (OID 4888209)
-- Name: krs_idkrs_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE krs_idkrs_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

--
-- TOC entry 12 (OID 4888209)
-- Name: krs_idkrs_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE krs_idkrs_seq FROM PUBLIC;
GRANT ALL ON TABLE krs_idkrs_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 13 (OID 4888211)
-- Name: detkrs_iddetkrs_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE detkrs_iddetkrs_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;
-- TOC entry 14 (OID 4888211)
-- Name: detkrs_iddetkrs_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE detkrs_iddetkrs_seq FROM PUBLIC;
GRANT ALL ON TABLE detkrs_iddetkrs_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 15 (OID 4888213)
-- Name: klas_idklas_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE klas_idklas_seq
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;

-- TOC entry 16 (OID 4888213)
-- Name: klas_idklas_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE klas_idklas_seq FROM PUBLIC;
GRANT ALL ON TABLE klas_idklas_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 17 (OID 4888215)
-- Name: kuliah_idkuliah_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE kuliah_idkuliah_seq
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;

-- TOC entry 18 (OID 4888215)
-- Name: kuliah_idkuliah_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE kuliah_idkuliah_seq FROM PUBLIC;
GRANT ALL ON TABLE kuliah_idkuliah_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 19 (OID 4888217)
-- Name: kurikulum_idmtk_seq; Type: SEQUENCE; Schema: public; Owner: jack

CREATE SEQUENCE kurikulum_idmtk_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

REVOKE ALL ON TABLE kurikulum_idmtk_seq FROM PUBLIC;
GRANT ALL ON TABLE kurikulum_idmtk_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE SEQUENCE tawar_idtawar_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

REVOKE ALL ON TABLE tawar_idtawar_seq FROM PUBLIC;
GRANT ALL ON TABLE tawar_idtawar_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE SEQUENCE urutmtk_idurutmtk_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

REVOKE ALL ON TABLE urutmtk_idurutmtk_seq FROM PUBLIC;
GRANT ALL ON TABLE urutmtk_idurutmtk_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 179 (OID 4888225)
-- Name: cj; Type: VIEW; Schema: public; Owner: jack
--
CREATE VIEW cj AS
    SELECT t6.jur, t6.nomhs, t6.nama, t1.idkrs, t1.bsks, t1.jsks,
           t1.jmtk, sum(t5.kredit) AS sum, count(t2.iddetkrs) AS count
    FROM her t0, krs t1, detkrs t2, klas t3, tawar t4, kurikulum t5, mahasisw t6
    WHERE ((t0.idher = t1.idher) AND ((t0.nomhs)::text = (t6.nomhs)::text))
          AND (t1.idkrs = t2.idkrs) AND (t2.idkelas = t3.idklas)
          AND (t3.idtawar = t4.idtawar) AND (t4.idmtk = t5.idmtk)
    GROUP BY t6.jur, t6.nomhs, t6.nama, t1.idkrs, t1.bsks, t1.jsks, t1.jmtk;

-- TOC entry 180 (OID 4888228)
-- Name: Kurang; Type: VIEW; Schema: public; Owner: jack
--
CREATE VIEW "Kurang" AS
    SELECT krs.idher, krs.idkrs, krs.bsks, krs.jsks, krs.jmtk
    FROM krs
    WHERE (krs.bsks < krs.jsks);

-- TOC entry 181 (OID 4888231)
-- Name: bskskurang; Type: VIEW; Schema: public; Owner: jack
--
CREATE VIEW bskskurang AS
    SELECT krs.idher, krs.idkrs, krs.bsks, krs.jsks, krs.jmtk
    FROM krs
    WHERE (krs.bsks < krs.jsks);

-- TOC entry 25 (OID 4888232)
-- Name: setorbaa_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE setorbaa_id_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

-- TOC entry 26 (OID 4888232)
-- Name: setorbaa_id_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE setorbaa_id_seq FROM PUBLIC;
GRANT ALL ON TABLE setorbaa_id_seq TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 182 (OID 4888234)
-- Name: setorbaa; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE setorbaa (  
id integer DEFAULT nextval('"setorbaa_id_seq"'::text) NOT NULL,  
nomhs character varying(15),  
jher smallint,  
idbank smallint,  
tglher date,  
tglbank date,  
nominal real  
);

--
-- TOC entry 183 (OID 4888234)
-- Name: setorbaa; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE setorbaa FROM PUBLIC;
GRANT ALL ON TABLE setorbaa TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 184 (OID 4888237)
-- Name: bank; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE bank (  
id smallint NOT NULL,  
bank character varying(20),  
alamat character varying(30),  
rekening character varying(25),  
an character varying(30)  
);

--
-- TOC entry 185 (OID 4888237)
-- Name: bank; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE bank FROM PUBLIC;
GRANT ALL ON TABLE bank TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 27 (OID 4888239)
-- Name: jenisher_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE jenisher_id_seq  
INCREMENT BY 1  
MAXVALUE 2147483647  
NO MINVALUE
CACHE 1;

-- TOC entry 28 (OID 4888239)
-- Name: jenisher_id_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE jenisher_id_seq FROM PUBLIC;
GRANT ALL ON TABLE jenisher_id_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 186 (OID 4888241)
-- Name: jenisher; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE jenisher (
    id integer DEFAULT nextval('"jenisher_id_seq"'::text) NOT NULL,
    jenis character varying(35)
);

-- TOC entry 187 (OID 4888241)
-- Name: jenisher; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE jenisher FROM PUBLIC;
GRANT ALL ON TABLE jenisher TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 188 (OID 4888244)
-- Name: tglserah; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tglserah (
    id integer DEFAULT nextval('"tglserah_id_seq"'::text) NOT NULL,
    idtawar integer NOT NULL,
    kdsn character varying(6) NOT NULL,
    tglsum date,
    tglsus date,
    catatanumid text,
    catatansem text,
    tglium date,
    tglius date
);

-- TOC entry 189 (OID 4888244)
-- Name: tglserah; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tglserah FROM PUBLIC;
GRANT ALL ON TABLE tglserah TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 190 (OID 4888250)
-- Name: km; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE km (id integer DEFAULT nextval('"km_id_seq"::text') NOT NULL, idklas integer, kode character varying(6), nilai real);

--
-- TOC entry 191 (OID 4888250)
-- Name: km; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE km FROM PUBLIC;
GRANT ALL ON TABLE km TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 29 (OID 4888253)
-- Name: km_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE km_id_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

--
-- TOC entry 30 (OID 4888253)
-- Name: km_id_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE km_id_seq FROM PUBLIC;
GRANT ALL ON TABLE km_id_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 192 (OID 4888255)
-- Name: mques; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE mques (item smallint NOT NULL, itemnya text NOT NULL, penilaian text, kelompok character varying(2));
REVOKE ALL ON TABLE mques FROM PUBLIC;
GRANT ALL ON TABLE mques TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE ques (    idq integer DEFAULT nextval('"ques_idq_seq"'::text) NOT NULL,    id integer,    item integer NOT NULL,    nilai smallint NOT NULL    );

REVOKE ALL ON TABLE ques FROM PUBLIC;
GRANT ALL ON TABLE ques TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE SEQUENCE ques_idq_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

REVOKE ALL ON TABLE ques_idq_seq FROM PUBLIC;
GRANT ALL ON TABLE ques_idq_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

REVOKE ALL ON TABLE pques FROM PUBLIC;
GRANT ALL ON TABLE pques TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
CREATE TABLE pques (
  id integer DEFAULT nextval('"pques_id_seq"'::text) NOT NULL,
  jk character(1),
  ipk real,
  status smallint,
  idklas integer,
  saran text,
  kdosen character varying(6)
);

-- TOC entry 197 (OID 4888265)
-- Name: pques; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE pques FROM PUBLIC;
GRANT ALL ON TABLE pques TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 33 (OID 4888271)
-- Name: pques_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE pques_id_seq
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;

-- TOC entry 34 (OID 4888271)
-- Name: pques_id_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE pques_id_seq FROM PUBLIC;
GRANT ALL ON TABLE pques_id_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 198 (OID 4888273)
-- Name: tgumid; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tgumid (
  tglke smallint NOT NULL,
  tglujian character varying(17),
  hari character varying(6)
);

-- TOC entry 199 (OID 4888273)
-- Name: tgumid; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tgumid FROM PUBLIC;
GRANT ALL ON TABLE tgumid TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 200 (OID 4888275)
-- Name: klsumid; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE klsumid (  
iduji integer DEFAULT nextval('klsumid_iduji_seq''::text) NOT NULL,  
idtawar integer NOT NULL,  
itgjamruang integer NOT NULL,  
kode character varying(5) NOT NULL,  
hari character varying(6) NOT NULL,  
tanggal character varying(20) NOT NULL,  
jam character varying(5) NOT NULL,  
dtpu smallint NOT NULL,  
dosen character varying(5),  
jpes smallint
);

-- TOC entry 201 (OID 4888275)
-- Name: klsumid; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE klsumid FROM PUBLIC;
GRANT ALL ON TABLE klsumid TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 35 (OID 4888278)
-- Name: klsumid_iduji_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE klsumid_iduji_seq  
INCREMENT BY 1  
MAXVALUE 2147483647  
NO MINVALUE  
CACHE 1;

-- TOC entry 36 (OID 4888278)
-- Name: klsumid_iduji_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE klsumid_iduji_seq FROM PUBLIC;
GRANT ALL ON TABLE klsumid_iduji_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 202 (OID 4888280)
-- Name: matumid; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE matumid (
    idtggljamruang integer DEFAULT nextval('matumid_idtggljamruang_seq'::text) NOT NULL,
    kode character varying(5),
    tglke smallint,
    hari character varying(6),
    tanggal character varying(20),
    dtpu smallint,
    isi boolean,
    jamke smallint,
    jam character varying(5)
);

-- TOC entry 203 (OID 4888280)
-- Name: matumid; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE matumid FROM PUBLIC;
GRANT ALL ON TABLE matumid TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 37 (OID 4888283)
-- Name: matumid_idtggljamruang_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE matumid_idtggljamruang_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

-- TOC entry 38 (OID 4888283)
-- Name: matumid_idtggljamruang_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE matumid_idtggljamruang_seq FROM PUBLIC;
GRANT ALL ON TABLE matumid_idtggljamruang_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 39 (OID 4888285)
-- Name: klsuji_iduji_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE klsuji_iduji_seq
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;
REVOKE ALL ON TABLE klsuji_iduji_seq FROM PUBLIC;
GRANT ALL ON TABLE klsuji_iduji_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE SEQUENCE matrikujian_idtgljamruang_seq
INCREMENT BY 1
MAXVALUE 2147483647
NO MINVALUE
CACHE 1;

REVOKE ALL ON TABLE matrikujian_idtgljamruang_seq FROM PUBLIC;
GRANT ALL ON TABLE matrikujian_idtgljamruang_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE ipqpmtk (
  id integer DEFAULT nextval('"ipqpmtk_id_seq"'::text) NOT NULL,
  kode character varying(6),
  idtawar integer,
  idklas integer,
  a1 real,
  a2 real,
  a3 real,
  a4 real,
  a5 real,
  tsc real,
  jmlres integer,
  jmlpes integer,
  ipkrt2 real,
  nilai real,
  kjur character varying(2)
);

REVOKE ALL ON TABLE ipqpmtk FROM PUBLIC;
GRANT ALL ON TABLE ipqpmtk TO PUBLIC;
REVOKE ALL ON TABLE ipqpmtk FROM PUBLIC;
GRANT ALL ON TABLE ipqpmtk TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE SEQUENCE ipqpmtk_id_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

REVOKE ALL ON TABLE ipqpmtk_id_seq FROM PUBLIC;
GRANT ALL ON TABLE ipqpmtk_id_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE ipques (
    id integer DEFAULT nextval('"ipques_id_seq"'::text) NOT NULL,
    kode character varying(6),
    idtawar integer,
    a1 real,
    a2 real,
    a3 real,
    a4 real,
    a5 real,
    tsc real,
    jmlres integer,
    jmlpes integer,
    ipkrt2 real,
    nilai real
);

REVOKE ALL ON TABLE ipques FROM PUBLIC;
GRANT ALL ON TABLE ipques TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';

CREATE SEQUENCE ipques_id_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

REVOKE ALL ON TABLE ipques_id_seq FROM PUBLIC;
GRANT ALL ON TABLE ipques_id_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

CREATE TABLE ipq (  
id integer DEFAULT nextval('"ipq_id_seq"'::text) NOT NULL,  
kode character varying(6),  
a1 real,  
a2 real,  
a3 real,  
a4 real,  
a5 real,  
tsc real,  
jmlres integer,  
jmlpes integer,  
ipkrt2 real,  
nilai real,  
jmtk integer
);

REVOKE ALL ON TABLE ipq FROM PUBLIC;
GRANT ALL ON TABLE ipq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';
CREATE SEQUENCE ipq_id_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 2147483647
    NO MINVALUE
    CACHE 1;

--
-- TOC entry 48 (OID 4888302)
-- Name: ipq_id_seq; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE ipq_id_seq FROM PUBLIC;
GRANT ALL ON TABLE ipq_id_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 210 (OID 4888304)
-- Name: pstkul; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE pstkul (    
    kode character varying(6),
    idtawar integer,
    jpes integer
);

--
-- TOC entry 211 (OID 4888304)
-- Name: pstkul; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE pstkul FROM PUBLIC;
GRANT ALL ON TABLE pstkul TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 212 (OID 4888306)
-- Name: pstkls; Type: TABLE; Schema: public; Owner: jack
--

CREATE TABLE pstkls (    
    kode character varying(6),
    idtawar integer,
    idklas integer,
    jpes integer
);

--
-- TOC entry 213 (OID 4888306)
-- Name: pstkls; Type: ACL; Schema: public; Owner: jack
--

REVOKE ALL ON TABLE pstkls FROM PUBLIC;
GRANT ALL ON TABLE pstkls TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 214 (OID 4888308)
-- Name: quesx; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE quesx ( idq integer DEFAULT nextval('"quesx_idq_seq"'::text) NOT NULL, id integer, nilai1 smallint, nilai2 smallint, nilai3 smallint, nilai4 smallint, nilai5 smallint, nilai6 smallint, nilai7 smallint, nilai8 smallint, nilai9 smallint, nilai10 smallint, nilai11 smallint, nilai12 smallint, nilai13 smallint, nilai14 smallint, nilai15 smallint, nilai16 smallint )

--
-- TOC entry 215 (OID 4888308)
-- Name: quesx; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE quesx FROM PUBLIC;
GRANT ALL ON TABLE quesx TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 49 (OID 4888311)
-- Name: quesx_idq_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE quesx_idq_seq
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;

--
-- TOC entry 50 (OID 4888311)
-- Name: quesx_idq_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE quesx_idq_seq FROM PUBLIC;
GRANT ALL ON TABLE quesx_idq_seq TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 216 (OID 4888313)
-- Name: ipqj; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE ipqj (  
id integer DEFAULT nextval('"ipqj_id_seq"'::text) NOT NULL,
kode character varying(6),
a1 real,
a2 real,
a3 real,
a4 real,
a5 real,
tsc real,
jmlres integer,
jmlpes integer,
ipkrt2 real,
nilai real,
kjur character varying(2),
jmtk integer
);

--
-- TOC entry 217 (OID 4888313)
-- Name: ipqj; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE ipqj FROM PUBLIC;
GRANT ALL ON TABLE ipqj TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 51 (OID 4888316)
-- Name: ipqj_id_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE ipqj_id_seq
START WITH 1
INCREMENT BY 1
MAXVALUE 2147483647
NO MINVALUE
CACHE 1;

--
-- TOC entry 52 (OID 4888316)
-- Name: ipqj_id_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE ipqj_id_seq FROM PUBLIC;
GRANT ALL ON TABLE ipqj_id_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 218 (OID 4888318)
-- Name: matdos; Type: TABLE; Schema: public; Owner: jack
CREATE TABLE matdos (
    kode character varying(6),
    idkuliah integer,
    tgl date,
    tglisi date,
    materi text,
    pertemuanke integer,
    judul text,
    jmlmhs integer,
    jam character varying(11)
);

--
-- TOC entry 219 (OID 4888318)
-- Name: matdos; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE matdos FROM PUBLIC;
GRANT ALL ON TABLE matdos TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 220 (OID 4888323)
-- Name: registernilai; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE registernilai (
    idregisternilai integer DEFAULT nextval('registernilai_idregisternilai_seq'::text) NOT NULL,
    dosen character varying(6),
    tglreg date,
    tglbatasisi date,
    iduji integer,
    idkelas integer,
    idtawar integer,
    sql text,
    isi date,
    jur character varying(2),
    cek date,
    posting date,
    absen integer,
    peserta integer
);

--
-- TOC entry 221 (OID 4888323)
-- Name: registernilai; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE registernilai FROM PUBLIC;
GRANT ALL ON TABLE registernilai TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 53 (OID 4888329)
CREATE SEQUENCE registernilai_idregisternilai_seq
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

CREATE TABLE tempnilai (
    iddetkrs integer,
    idregisternilai integer,
    akhir character(1),
    tugas character varying(4),
    mid character varying(4),
    sem character varying(4),
    nomhs character varying(10),
    baru character(1)
);

CREATE TABLE tempubahnilai (
    iddetkrs integer,
    idubahnilai integer,
    akhir character(1),
    tugas character varying(4),
    mid character varying(4),
    sem character varying(4),
    nomhs character varying(10),
    baru character(1)
);
nomhs character varying(10),
baru character(1)
);

-- TOC entry 225 (OID 4888333)
-- Name: tempubahnilai; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempubahnilai FROM PUBLIC;
GRANT ALL ON TABLE tempubahnilai TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 226 (OID 4888335)
-- Name: ubahnilai; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE ubahnilai (  
idubahnilai integer DEFAULT nextval('ubahnilai_idubahnilai_seq'::text) NOT NULL,  
tgl date,  
isi date,  
posting date,  
keterangan text,  
idreg character varying(50),  
cek date)
);

--
-- TOC entry 227 (OID 4888335)
-- Name: ubahnilai; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE ubahnilai FROM PUBLIC;
GRANT ALL ON TABLE ubahnilai TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 55 (OID 4888341)
-- Name: ubahnilai_idubahnilai_seq; Type: SEQUENCE; Schema: public;
Owner: jack
--
CREATE SEQUENCE ubahnilai_idubahnilai_seq  
INCREMENT BY 1  
NO MAXVALUE  
NO MINVALUE  
CACHE 1;

--
-- TOC entry 56 (OID 4888341)
-- Name: ubahnilai_idubahnilai_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE ubahnilai_idubahnilai_seq FROM PUBLIC;
GRANT ALL ON TABLE ubahnilai_idubahnilai_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 228 (OID 4888343)
-- Name: registernilaimid; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE registernilaimid (  idregisternilai integer DEFAULT nextval('registernilaimid_idregisternilai_seq'::text) NOT NULL,  dosen character varying(6),  tglreg date,  tglbatasisi date,  iduji integer,  idkelas integer,  idtawar integer,  sql text,  isi date,  jur character varying(2),  cek date,  posting date,  absen integer,  peserta integer );

--
-- TOC entry 229 (OID 4888343)
-- Name: registernilaimid; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE registernilaimid FROM PUBLIC;
GRANT ALL ON TABLE registernilaimid TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

--
-- TOC entry 57 (OID 4888349)
-- Name: registernilaimid_idregisternilai_seq; Type: SEQUENCE; Schema: public; Owner: jack
--
CREATE SEQUENCE registernilaimid_idregisternilai_seq  INCREMENT BY 1  NO MAXVALUE  NO MINVALUE  CACHE 1;

--
-- TOC entry 58 (OID 4888349)
-- Name: registernilaimid_idregisternilai_seq; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE registernilaimid_idregisternilai_seq FROM PUBLIC;
GRANT ALL ON TABLE registernilaimid_idregisternilai_seq TO PUBLIC;
SET SESSION AUTHORIZATION 'jack';

-- TOC entry 230 (OID 4888351)
-- Name: tempnilaimid; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempnilaimid (    iddetkrs integer,    idregisternilai integer,    akhir character(1),    tugas character varying(4),    mid character varying(4),    sem character varying(4),    nomhs character varying(10),    baru character(1) );

-- TOC entry 231 (OID 4888351)
-- Name: tempnilaimid; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempnilaimid FROM PUBLIC; GRANT ALL ON TABLE tempnilaimid TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 232 (OID 4888353)
-- Name: tempubahnilaimid; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE tempubahnilaimid (    iddetkrs integer,    idubahnilai integer,    akhir character(1),    tugas character varying(4),    mid character varying(4),    sem character varying(4),    nomhs character varying(10),    baru character(1) );

-- TOC entry 233 (OID 4888353)
-- Name: tempubahnilaimid; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE tempubahnilaimid FROM PUBLIC; GRANT ALL ON TABLE tempubahnilaimid TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 234 (OID 4888355)
-- Name: ubahnilaimid; Type: TABLE; Schema: public; Owner: jack
--
CREATE TABLE ubahnilaimid (
    idubahnilai integer DEFAULT nextval('ubahnilaimid_idubahnilai_seq'::text) NOT NULL,
    tgl date,
    isi date,
    posting date,
    keterangan text,
    idreg character varying(50),
    cek date
);

-- TOC entry 235 (OID 4888355)
-- Name: ubahnilaimid; Type: ACL; Schema: public; Owner: jack
--
REVOKE ALL ON TABLE ubahnilaimid FROM PUBLIC;
GRANT ALL ON TABLE ubahnilaimid TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 59 (OID 4888361)
-- Name: ubahnilaimid_idubahnilai_seq; Type: SEQUENCE; Schema: public;
-- Owner: jack
--
CREATE SEQUENCE ubahnilaimid_idubahnilai_seq
    START WITH 100
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

-- TOC entry 60 (OID 4888361)
-- Name: ubahnilaimid_idubahnilai_seq; Type: ACL; Schema: public;
-- Owner: jack
--
REVOKE ALL ON TABLE ubahnilaimid_idubahnilai_seq FROM PUBLIC;
GRANT ALL ON TABLE ubahnilaimid_idubahnilai_seq TO PUBLIC;

SET SESSION AUTHORIZATION 'jack';

-- TOC entry 263 (OID 5081256)
-- Name: tglserah_id_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX tglserah_id_key ON tglserah USING btree (id);

-- TOC entry 264 (OID 5081257)
-- Name: km_id_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX km_id_key ON km USING btree (id);

-- TOC entry 266 (OID 5081258)
-- Name: ques_idq_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX ques_idq_key ON ques USING btree (id);

-- TOC entry 271 (OID 5081259)
-- Name: ipqpmtk_id_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX ipqpmtk_id_key ON ipqpmtk USING btree (id);

-- TOC entry 272 (OID 5081260)
-- Name: ipques_id_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX ipques_id_key ON ipques USING btree (id);

-- TOC entry 273 (OID 5081261)
-- Name: ipq_id_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX ipq_id_key ON ipq USING btree (id);

-- TOC entry 274 (OID 5081262)
-- Name: quesx_idq_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX quesx_idq_key ON quesx USING btree (id);

-- TOC entry 275 (OID 5081263)
-- Name: ipqj_id_key; Type: INDEX; Schema: public; Owner: jack
--
CREATE UNIQUE INDEX ipqj_id_key ON ipqj USING btree (id);

-- TOC entry 236 (OID 5081264)
-- Name: tawar_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY tawar
    ADD CONSTRAINT tawar_pkey PRIMARY KEY (idtawar);

-- TOC entry 237 (OID 5081266)
-- Name: klas_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
ALTER TABLE ONLY klas
    ADD CONSTRAINT klas_pkey PRIMARY KEY (idklas);

-- TOC entry 238 (OID 5081268)
-- Name: detkrs_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY detkrs
    ADD CONSTRAINT detkrs_pkey PRIMARY KEY (iddetkrs);

-- TOC entry 239 (OID 5081270)
-- Name: krs_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY krs
    ADD CONSTRAINT krs_pkey PRIMARY KEY (idkrs);

-- TOC entry 240 (OID 5081272)
-- Name: prodi_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY prodi
    ADD CONSTRAINT prodi_pkey PRIMARY KEY (kode);

-- TOC entry 241 (OID 5081274)
-- Name: dosen_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY dosen
    ADD CONSTRAINT dosen_pkey PRIMARY KEY (kode);

-- TOC entry 242 (OID 5081276)
-- Name: ajar_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY ajar
    ADD CONSTRAINT ajar_pkey PRIMARY KEY (idajar);

-- TOC entry 243 (OID 5081278)
-- Name: jam_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY jam
    ADD CONSTRAINT jam_pkey PRIMARY KEY (jamke);

-- TOC entry 244 (OID 5081280)
-- Name: hari_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
ALTER TABLE ONLY hari
  ADD CONSTRAINT hari_pkey PRIMARY KEY (idhari);

--
-- TOC entry 245 (OID 5081282)
-- Name: kuliah_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY kuliah
  ADD CONSTRAINT kuliah_pkey PRIMARY KEY (idkuliah);

--
-- TOC entry 246 (OID 5081284)
-- Name: harijamruang_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY harijamruang
  ADD CONSTRAINT harijamruang_pkey PRIMARY KEY (idharijam);

--
-- TOC entry 247 (OID 5081286)
-- Name: fakultas_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY fakultas
  ADD CONSTRAINT fakultas_pkey PRIMARY KEY (kodefak);

--
-- TOC entry 248 (OID 5081288)
-- Name: mahasiswa_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY mahasiswa
  ADD CONSTRAINT mahasiswa_pkey PRIMARY KEY (nomhs);

--
-- TOC entry 249 (OID 5081290)
-- Name: matrikualijian_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY matrikualijian
  ADD CONSTRAINT matrikualijian_pkey PRIMARY KEY (idtgljamruang);

--
-- TOC entry 250 (OID 5081292)
-- Name: klsuji_pkey; Type: CONSTRAINT; Schema: public; Owner: jack
--
ALTER TABLE ONLY klsuji
  ADD CONSTRAINT klsuji_pkey PRIMARY KEY (iduji);
-- TOC entry 251 (OID 5081294)
-- Name: tanggalujian_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY tanggalujian
    ADD CONSTRAINT tanggalujian_pkey PRIMARY KEY (tglke);

--
-- TOC entry 252 (OID 5081296)
-- Name: ngajar_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY ngajar
    ADD CONSTRAINT ngajar_pkey PRIMARY KEY (idajar);

--
-- TOC entry 253 (OID 5081298)
-- Name: jamujian_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY jamujian
    ADD CONSTRAINT jamujian_pkey PRIMARY KEY (jamke);

--
-- TOC entry 254 (OID 5081300)
-- Name: interip_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY interip
    ADD CONSTRAINT interip_pkey PRIMARY KEY (ips);

--
-- TOC entry 255 (OID 5081302)
-- Name: jumlahsks_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY jumlahsks
    ADD CONSTRAINT jumlahsks_pkey PRIMARY KEY (nomhs, nama);

--
-- TOC entry 256 (OID 5081304)
-- Name: dether_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY dether
    ADD CONSTRAINT dether_pkey PRIMARY KEY (iddether);

--
-- TOC entry 257 (OID 5081306)
-- Name: ijinpakai_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY ijinpakai
    ADD CONSTRAINT ijinpakai_pkey PRIMARY KEY ("level");
-- TOC entry 258 (OID 5081308)
-- Name: ruang_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY ruang
    ADD CONSTRAINT ruang_pkey PRIMARY KEY (kode);

-- TOC entry 259 (OID 5081310)
-- Name: tempbeban_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY tempbeban
    ADD CONSTRAINT tempbeban_pkey PRIMARY KEY (kode);

-- TOC entry 260 (OID 5081312)
-- Name: setorbaa_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY setorbaa
    ADD CONSTRAINT setorbaa_pkey PRIMARY KEY (id);

-- TOC entry 261 (OID 5081314)
-- Name: bank_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY bank
    ADD CONSTRAINT bank_pkey PRIMARY KEY (id);

-- TOC entry 262 (OID 5081316)
-- Name: jenisher_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY jenisher
    ADD CONSTRAINT jenisher_pkey PRIMARY KEY (id);

-- TOC entry 265 (OID 5081318)
-- Name: mques_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY mques
    ADD CONSTRAINT mques_pkey PRIMARY KEY (item);

-- TOC entry 267 (OID 5081320)
-- Name: pques_pkey; Type: CONSTRAINT; Schema: public; Owner: jack

ALTER TABLE ONLY pques
    ADD CONSTRAINT pques_pkey PRIMARY KEY (id);
ALTER TABLE ONLY tgu_field
  ADD CONSTRAINT tgu_field_pkey PRIMARY KEY (tgu_field);

ALTER TABLE ONLY klsum
  ADD CONSTRAINT klsum_pkey PRIMARY KEY (iduji);

ALTER TABLE ONLY matum
  ADD CONSTRAINT matum_pkey PRIMARY KEY (idtgljamruang);

SET SESSION AUTHORIZATION 'postgres';

COMMENT ON SCHEMA public IS 'Standard public schema';
BAGIAN B.
DRAFT ARTIKEL ILMIAH
ANALISIS ASPEK-ASPEK KUALITAS SCHEMA DATABASE
(Studi Kasus Pada Database Akademik ISTA Yogyakarta)

Oleh:
Suwanto Raharjo dan Edhy Sutanta
Jurusan Teknik Informatika, FTI, ISTA Yogyakarta

Intisari
Kenyataan di lapangan selama ini menunjukkan bahwa tidak semua organisasi berhasil mengimplementasikan SIM dengan baik. Salah satu hasil audit pada implementasi SIM di Indonesia menunjukkan bahwa yang paling sering dijumpai adalah terjadinya fenomena tambal sulam aplikasi SIM, termasuk di dalamnya adalah kegagalan implementasi dan fenomena “tambal sulam” pada SIAKAD. Hal ini terbukti dengan masih banyaknya PT yang telah melakukan pengembangan dan implementasi SIAKAD, namun hasilnya belum memuaskan hingga saat ini. Salah satu peran database, yaitu sebagai sumber informasi bagi SIM, mengimplikasikan bahwa database yang digunakan harus mampu memenuhi kebutuhan berbagai informasi dan data bagi para penggunanya, baik untuk saat ini maupun mendatang. Rancangan schema database seharusnya dirancang sedemikian rupa sehingga mempunyai kualitas yang tinggi. Terkait dengan hal tersebut, terdapat banyak aspek yang mempengaruhi keberhasilan implementasi SIM dalam organisasi, diantaranya adalah aspek organisasi, manajemen, perilaku, atau teknologi yang meliputi perangkat keras, perangkat lunak, rekayasa sistem, database.

Dengan mengacu pada hasil penelitian Olaf Herden (2001), penelitian ini melakukan analisis tentang aspek-aspek kualitas schema database pada database akademik yang digunakan di ISTA. Analisis dilakukan pada rancangan logikal dengan menggunakan model logika berupa business rule yang digunakan pada ISTA. Aspek-aspek kualitas schema yang dianalisis meliputi 9 (sembilan) kriteria, yaitu correctness, consistency, relevance, scope, level of detail, completeness, minimality, ability of integration, serta readability. Hasil penelitian ini diharapkan dapat memberikan rekomendasi langkah yang mungkin dilakukan oleh pengelola/penanggungjawab database akademik di ISTA.

Berdasarkan hasil analisis, secara umum dapat dinyatakan bahwa schema database akademik ISTA memiliki kualitas yang rendah. Kondisi ini akan menjadi potensi bagi timbulnya berbagai permasalahan yang serius di kelak kemudian hari. Dan oleh karenanya, maka perlu segera dilakukan upaya-upaya penyempurnaan untuk menghindari timbulnya permasalahan yang lebih kompleks dan semakin sulit ditangani.

Kata-kata kunci : schema, database, kualitas shema database
PENDAHULUAN
Perkembangan teknologi informasi-komputer saat ini sudah mencapai pada tahap di mana ukurannya semakin kecil, kecepatannya semakin tinggi, namun harganya semakin murah dibandingkan dengan kemampuan kerjanya. Kondisi ini mendorong masyarakat berlomba-lomba memanfaatkan komputer sebagai alat bantu pengolahan data dengan cara membangun sistem pengolahan data terkomputerisasi untuk penyajian informasi, baik untuk keperluan pribadi maupun organisasinya. Sekalipun kurang tepat, sistem pengolahan data terkomputerisasi untuk penyajian informasi dalam suatu organisasi sering disebut sebagai Sistem Informasi Manajemen (SIM).


Kenyataan di lapangan menunjukkan, bahwa tidak semua organisasi berhasil mengimplementasikan SIM dengan baik. Eko Nugroho (1991) telah meneliti pengaruh struktur organisasi dan sumber daya manusia terhadap keberhasilan implementasi SIM dalam organisasi dalam penelitian tesisnya. Hasil penelitian tersebut membuktikan, bahwa rata-rata persentase tingkat keberhasilan implementasi SIM, khususnya di DIY relatif rendah, hanya sekitar 60%. Penelitian tersebut juga telah membuktikan bahwa sumberdaya manusia (para teknisi sistem informasi, misalnya operator, analis sistem komputer, manajer unit pemrosesan data dan lain-lain, dan para pengguna sistem informasi) merupakan faktor penting yang mempengaruhi keberhasilan implementasi SIM. Selanjutnya, penelitian tersebut menyarankan adanya penelitian lebih lanjut pada aspek-aspek lain yang mempengaruhi keberhasilan implementasi SIM dalam organisasi, yaiku aspek organisasi, manajemen, perilaku, atau teknologi yang meliputi perangkat keras, perangkat lunak, rekayasa sistem, database.


Organisasi yang mengimplementasikan SIM dapat bergerak dalam bidang apa saja, baik yang bergerak dalam bidang produksi barang maupun jasa, profit maupun non-profit, berskala kecil, menengah,
maupun besar. Salah satu contoh organisasi yang telah mengimplementasikan SIM adalah Perguruan Tinggi (PT). Implementasi SIM dalam lingkungan PT utamanya digunakan untuk pengolahan data akademik yang sering dikenal dengan sebutan Sistem Informasi Akademik (SIAKAD). Kegagalan implementasi SIM dan fenomena “tambal sulam” aplikasi SIM ternyata juga dapat terjadi dalam SIAKAD. Hal ini dapat dibuktikan dengan masih adanya PT yang telah melakukan pengembangan dan implementasi SIAKAD lebih dari satu dekade lamanya, namun hasilnya belum memuaskan hingga saat ini.

Salah satu peran database, yaitu sebagai sumber informasi bagi SIM, mengimplikasikan bahwa database yang digunakan harus mampu memenuhi kebutuhan berbagai informasi (dan data) bagi para penggunanya, baik untuk saat ini maupun mendatang. Tanpa mengabaikan sumber daya informasi yang lainnya, aspek rekayasa sistem, khususnya rancangan schema database akan sangat mempengaruhi keberhasilan implementasi SIM dalam organisasi. Rancangan schema database yang berkualitas akan meningkatkan kualitas SIM, sebaliknya rancangan schema database yang kurang berkualitas akan berpotensi menimbulkan permasalahan-permasalahan dan bahkan keagakan implementasi SIM pada suatu organisasi, termasuk Perguruan Tinggi. Dengan demikian, schema database seharusnya dirancang sedemikian rupa sehingga mempunyai kualitas yang baik/tinggi.

Dengan menggunakan meta model, penelitian Olaf Herden (2001) telah berhasil mendefinisikan dan mengukur aspek-aspek kualitas schema berorientasi obyek (object oriented) suatu database, dan mengusulkan sebuah proses untuk melakukan tinjauan dan pengukuran aspek kualitas schema suatu database. Aspek-aspek yang relevan untuk pengukuran kualitas schema suatu database yang dimaksud meliputi 9 (sembilan) kriteria, yaitu kebenaran (correctness), konsistensi (consistency), relevansi (relevance), jangkauan (scope), tingkat detail (level of detail), kelengkapan (completeness), minimalitas (minimality), kemampuan untuk diintegrasikan (ability of integration), serta kemampuan untuk dibaca (readability) (Olaf Herden, 2001, http://www.iro.umontreal.ca/~sahraouh/qaoose01/Herden.PDF).

Penelitian ini akan melakukan analisis/pengujian tentang aspek-aspek kualitas schema database. Analisis dilakukan pada rancangan schema database yang digunakan pada database akademik yang digunakan di ISTA. Analisis dilakukan pada rancangan logikal, bukan implementasi pada DBMS, teknologi perangkat keras, maupun program aplikasi yang digunakan. Dan, analisis dilakukan dengan menggunakan model logika berupa business rule yang digunakan pada ISTA.

Penelitian ini dilakukan dengan tujuan untuk mengetahui kualitas schema database yang digunakan pada database akademik di ISTA. Hasil dari penelitian ini diharapkan dapat memberikan manfaat berupa rekomendasi alternatif-alternatif solusi yang mungkin dilakukan oleh pengelola/penanggungjawab database akademik di ISTA, terkait dengan

**PEMBAHASAN**

**Database dan SIM**


Sementara itu, Raymond McLeod Jr. dan George Shell (2001) menyatakan bahwa salah satu sumber daya konseptual informasi dalam organisasi adalah berupa *database*. Dapat dipahami bahwa *database* merupakan sumber daya penting dalam SIM.

**Perancangan Database Untuk SIM**


Menurut Jan L. Harrington (2004), suatu rancangan *database* yang buruk akan mengakibatkan efek negatif, antara lain:

1. Munculnya kerangkapan data
2. Munculnya inkonsistensi data
3. Munculnya permasalahan pada penyisipan data
4. Munculnya permasalahan pada penghapusan data

Dalam referensi yang lain, Abraham Silberschatz, Henry F. Korth, dan S. Sudarshan (2001) menyatakan bahwa perancangan *database* relasional diperlukan untuk mendapatkan sekumpulan *schema* relasi yang baik. Rancangan yang buruk akan mengakibatkan perulangan informasi
1. Menghindari kerangkapan data
2. Menjamin bahwa kerelasian antar atribut dapat direpresentasikan
3. Memberikan fasilitas pengecekan batasan integritas pada proses update

Sementara itu, J. L. Whitten dan L. D. Bentley (1998), menyatakan bahwa tujuan dan prasyarat perancangan database adalah:
1. Database harus memberikan efisiensi media penyimpan (storage), update, dan penampilan kembali data-data
2. Database harus andal, yaitu memiliki integritas tinggi yang memberikan kepercayaan bagi para pengguna terhadap data
3. Database harus dapat beradaptasi (adaptable) dan dapat berkembang (scaleable) untuk memenuhi kebutuhan dan aplikasi baru

Untuk memperoleh SIM yang baik, maka pengembang SIM perlu memahami metode perancangan database yang baik, yaitu(http://www2.cstudies.ubc.ca/~mullen/IEDBS.html):
1. Mengidentifikasi kebutuhan informasi pada sebuah bisnis
2. Merancang database relasional yang didasarkan pada kebutuhan bisnis
3. Membuat dokumenasi database
4. Meningkatkan fleksibilitas database untuk mengantisipasi perubahan
5. Menyederhanakan database dengan jumlah tabel
6. Membangun database relasional dan menyesuaikannya untuk memperoleh kinerja yang optimal
7. Meningkatkan kinerja database dengan indexing dan mengontrol kerangkapan
8. Mengurangi beaya pengembangan software dengan generalisasi

1. Kamus data (Data Dictionary/DD)
2. Kebutuhan pengguna
3. Diagram Aliran Data/DAD (Data Flow Diagram/DFD)
4. Bagan struktur (structure chart)
5. Model data (data model)
6. Prototipe tampilan layar
7. Model keadaan (state model)
8. Diagram tugas (task diagram)
9. Diagram kelas (class diagram)
10. Diagram objek (object diagram)

Dalam referensi yang lain, Raymond McLeod dan George Schell (2001) menyatakan bahwa penyusunan database dapat dilakukan dengan menggunakan dua pendekatan, yaitu:
1. Pendekatan berorientasi proses (pendekatan pemecahan permasalahan)
2. Pendekatan model perusahaan. Selanjutnya, penyusunan database dilakukan melalui tiga langkah utama, yaitu:
   a. Mendeskripsikan data
   b. Memasukkan data
   c. Menggunakan database (menggunakan bahasa query, QBE, DML)


Menurut Paul Litwin (2003), perancangan database lebih merupakan suatu seni daripada suatu ilmu pengetahuan. Sekalipun tidak mengungkap seluruh tahapan dalam proses perancangan, Paul Litwin memberikan kerangka kerja yang sesuai digunakan dalam perancangan database, yaitu sebagai berikut (http://r937.com/relational.html):
1. Pelajari proses bisnis dan proses lain dalam organisasi untuk mencoba membuat model.
2. Tulislah pernyataan mendasar atau misi yang harus dilakukan oleh sistem. Kemudian dilanjutkan dengan membuat daftar kebutuhan pada sistem.


5. Perhatikan seluruh laporan yang tercetak pada kertas atau komputer.

6. Pastikan bahwa setiap data dalam laporan tersedia di dalam tabel. Jika data belum tersedia, tambahkan data tersebut dalam tabel-tabel yang ada atau buatlah tabel baru.

7. Tambahkan dan isikan beberapa baris data pada setiap tabel.


10. Pastikan bahwa tabel memenuhi 1NF, yaitu seluruh atribut bernilai atomik dan tidak memuat grup perulangan. Jika diperlukan, pecahlah tabel untuk memperoleh 1NF.

11. Pastikan bahwa tabel memenuhi 2NF, yaitu setiap tabel hanya mendeskripsikan sebuah entitas dan semua atribut non-key bergantung sepenuhnya (FFD) pada PK. Jika diperlukan, pecahlah tabel untuk memperoleh 2NF. Jika tabel memiliki PK berupa kunci komposit, maka pecahlah PK menjadi atribut-atribut yang seluruhnya ditempatkan pada tabel itu juga.

12. Pastikan bahwa tabel memenuhi 3NF, yaitu hilangkan atribut yang nilainya dapat dihitung dan hilangkan atribut yang bersifat mutual dependent dengan cara membuat tabel lookup.


17. Kembali ke rancangan tabel dan aturan bisnis.

Paul Litwin (2003) juga menyatakan bahwa penggunaan model RDBM dalam perancangan database akan memberikan beberapa keuntungan, antara lain:
1. Entri data, update, dan penghapusan data akan lebih efisien.
2. Penampilan kembali, peringkasan, dan pelaporan data akan lebih efisien.
3. Jika database dirancang dengan menggunakan model yang diformulasikan dengan baik, maka perilakunya akan dapat diprediksi.
4. Jika informasi disimpan dalam database (bukan dalam program aplikasi) maka database memiliki dokumentasi tersendiri
5. Perubahan pada schema database dapat dilakukan dengan lebih mudah.

Dalam referensi yang lain, Jun Yang (1999) menyatakan bahwa penyusunan database dapat dilakukan melalui empat tahapan, yaitu:
1. Memahami domain dunia nyata yang akan ditangkap
2. Menspesifikasikan domain dunia nyata tersebut dengan menggunakan model data (menggunakan ER_M atau Object Definition Language/ODL)
3. Menterjemahkan spesifikasi ke model database (relasional, ODL)


Pendekatan Dalam Pengukuran Kualitas Informasi
235

Wittig (2000)). Dalam hal ini terdapat tiga macam pendekatan yang berbeda yang dapat digunakan untuk menspesifikasikan kualitas informasi, yaitu (Klesse, Herrmann, Brändli, Mügeli, Maier, http://wotan.liu.edu/docis/dbl/iqiqiq/2004_418_CIPACS.htm):

1. Pendekatan intuitif (intuitive approach), pendekatan ini mengusulkan atribut atribut kualitas informasi yang didasarkan pada wawasan/pengetahuan subyektif seseorang.


3. Pendekatan teoritis (theoretical approach), pendekatan ini berusaha memunculkan usulan teori baru didasarkan pada teori-teori yang sudah ada.

Kelemahan utama pendekatan intuitif dan empiris adalah pengaruh tingkat pengetahuan peneliti terhadap penilaian yang diberikan. Sedangkan kelemahan pendekatan teoritis adalah kemungkinan terjadinya kesalahan pada teori baru yang diusulkan (Klesse, Herrmann, Brändli, Mügeli, Maier, http://wotan.liu.edu/docis/dbl/iqiqiq/2004_418_CIPACS.htm).

Implementasi SIAKAD di ISTA


Hingga saat ini, UPT PUSKOM telah berhasil mengembangkan aplikasi-aplikasi berbasis komputer yang digunakan untuk menangani proses pengolahan data di lingkungan ISTA, yaitu:

1. Sub Sistem Informasi Penerimaan Mahasiswa Baru
2. Sub Sistem Informasi Akademik
3. Sub Sistem Informasi Perpustakaan
4. Sub Sistem Informasi Keuangan Mahasiswa
5. Sub Sistem Informasi Kepegawaian
6. Sub Sistem Informasi Inventaris
Selain itu, aplikasi-aplikasi berukuran kecil dengan karakteristik yang spesifik juga telah dikembangkan dan diimplementasikan sebagai modul-modul yang terpisah, yaitu:

1. Aplikasi untuk pengolahan data Penggajian dosen-karyawan
2. Aplikasi untuk pengolahan data wisuda
3. Aplikasi untuk pengolahan data Praktikum
4. Aplikasi untuk pengolahan data Kuliah Kerja Nyata
5. Aplikasi untuk pengolahan data PKN-KP-Merencana-TA-Skripsi
6. Aplikasi untuk pengolahan data Evaluasi Kinerja Dosen Mengajar
7. Aplikasi untuk pengolahan data Realisasi dan Pencairan Anggaran

Pengembangan dan penerapan sistem informasi berbasis komputer di lingkungan ISTA memiliki 2 (dua) macam tujuan, yaitu:

1. Untuk memenuhi kebutuhan administrasi umum, keuangan, dan administrasi akademik.
2. Untuk memenuhi kebutuhan data statistik yang diperlukan dalam pengelolaan setiap Jurusan/Program Studi.

Secara umum, setiap sub sistem informasi dan aplikasi untuk pengolahan data yang telah dikembangkan oleh UPT PUSKOM dan diterapkan di lingkungan ISTA, mampu memberikan output berupa informasi (dan data) dalam format terinci, teringkas/rekapitulasi, maupun spesifik; baik yang bersifat insidental maupun rutin; dalam bentuk teks, tabel, maupun grafis; serta dapat ditampilkan pada media softcopy maupun hardcopy. Dengan tampilan dialog berbasis visual yang user friendly, maka informasi (dan data) yang dihasilkan dapat diakses secara mudah dan cepat. Output informasi (dan data) tersebut dapat diakses/digunakan oleh para pengguna di lingkungan ISTA maupun pihak luar, yang sesuai dengan kewenangannya, dan dapat dikelompokkan menjadi 4 (empat), yaitu:

1. Pengguna pada tingkat manajemen tertinggi (perencanaan strategis).
2. Para pengguna pada tingkat manajemen menengah (perencanaan taktis dan pengendalian manajemen).
3. Para pengguna pada tingkat manajemen terendah (perencanaan dan pengendalian operasional).
4. Personil atau institusi pengguna lainnya, yaitu dosen, mahasiswa, alumni, masyarakat pengguna lulusan, pemerintah (Depdiknas, Dikti, BAN-PT, Kopertis), penyandang dana (Yayasan Pembina Potensi Pembangunan/YPPP), orang tua/wali mahasiswa, serta masyarakat umum.

Hasil Analisis/Pengujuan

Penerapan SIAKAD di ISTA saat ini didukung oleh database akademik yang diimplementasikan dengan menggunakan database server model RDBMS, yaitu PostgreSQL. Schema database akademik ISTA tersusun atas 181 tabel. Secara teringkas, hasil analisis pada aspek-
aspek kualitas *schema* database akademik ISTA yang dianalisis ditampilkan pada Tabel 1.

**Tabel 1: Hasil analisis aspek kualitas *schema* database akademik ISTA**

<table>
<thead>
<tr>
<th>Kriteria</th>
<th>Deskripsi</th>
<th>Hasil Analisis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kebenaran (correctness)</td>
<td>Merupakan aspek teknik, apakah semua aspek dimodelkan secara benar sesuai dengan kebutuhan dan batasan sistem. Aspek ini dapat digunakan untuk mengukur semua <em>schema</em>. Pengukuran dilakukan menggunakan kepakanan dengan meningkat tingkat kedalaman pengetahuan pada seluruh aspek teknik.</td>
<td><em>Schema database</em> akademik ISTA tidak memiliki suatu batasan (constraint) yang dapat mem-filter masuknya data ke dalam tabel sehingga tidak ada jaminan bahwa data yang dimasukan merupakan data yang benar</td>
</tr>
<tr>
<td>Pendekatan</td>
<td>Deskripsi</td>
<td>Keterangan</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>Kemampuan untuk diintegrasikan (ability of integration)</td>
<td>Apakah schema telah disesuaikan untuk standarisasi dalam organisasi secara menyeluruh atau pemodelan. Aspek ini tergantung pada konteks proyek atau organisasi, yaitu apakah lebih banyak terlibat pada asosiasi ekonomi atau berorientasi internasional, tetapi yang lebih relevan jika memiliki kemampuan untuk diintegrasikan. Pengukuran dilakukan untuk menentukan apakah pengunaan nama-nama hanya dapat digunakan untuk cabang atau dapat diterima secara internasional.</td>
<td>Schema database akademik ISTA belum menerapkan standarisasi data dalam organisasi secara menyeluruh, sehingga akan menimbulkan kesulitan pada saat akan mengintegrasikan database sehingga menjadi SRIPT yang terintegrasi.</td>
</tr>
</tbody>
</table>

**KESIMPULAN**

DAFTAR PUSTAKA


AMULET Development Corp, 2004, How To Modify Database Structure, eCriteria, Web Hosted Database for Business


Eppler, M.J. and Wittig, D., 2000, Conceptualizing Information Quality: A Review of Information Quality Frameworks from the Last Ten Years, presented at 5th International Conference on Information Quality (ICIQ 2000), Cambridge, MA


Indrajit, R.E., 2000, Manajemen Sistem Informasi dan Teknologi Informasi, PT. Elex Media Komputindo, Jakarta


Loudon Loudon, 2004, Management Information Systems 8/e, Prentice Hall


Nugroho, E., 1991, Pengaruh Struktur Organisasi dan Sumber Daya Manusia Terhadap Keberhasilan Implementasi Sistem Informasi
Manajemen dalam Organisasi, Tesis, Program Magister Sains, Prodi Akuntansi, Fakultas Ekonomi, UGM, Yogyakarta


Redman, T.C., 1996, Data Quality for the Information Age, Boston, MA: Artech House


Mullen, B., 2005, *Generalized Table Suggestions*, http://www2.cstudies.ubc.ca/~mullen/IEDBS.html#GeneralTables, diakses pada tanggal 11-08-2005


BAGIAN C. SINOPSIS PENELITIAN LANJUTAN

Untuk penelitian selanjutnya, dapat dilakukan untuk melakukan analisis pada aspek-aspek lain yang mempengaruhi keberhasilan implementasi SIM dalam organisasi, yaitu aspek organisasi, manajemen, perilaku, atau teknologi yang meliputi perangkat keras, perangkat lunak, rekayasa sistem. Berdasarkan hasil-hasil analisis tersebut, selanjutnya dapat digunakan untuk mengembangkan model sistem, database, serta aplikasi sistem akademik berbasis komputer yang terintegrasi.