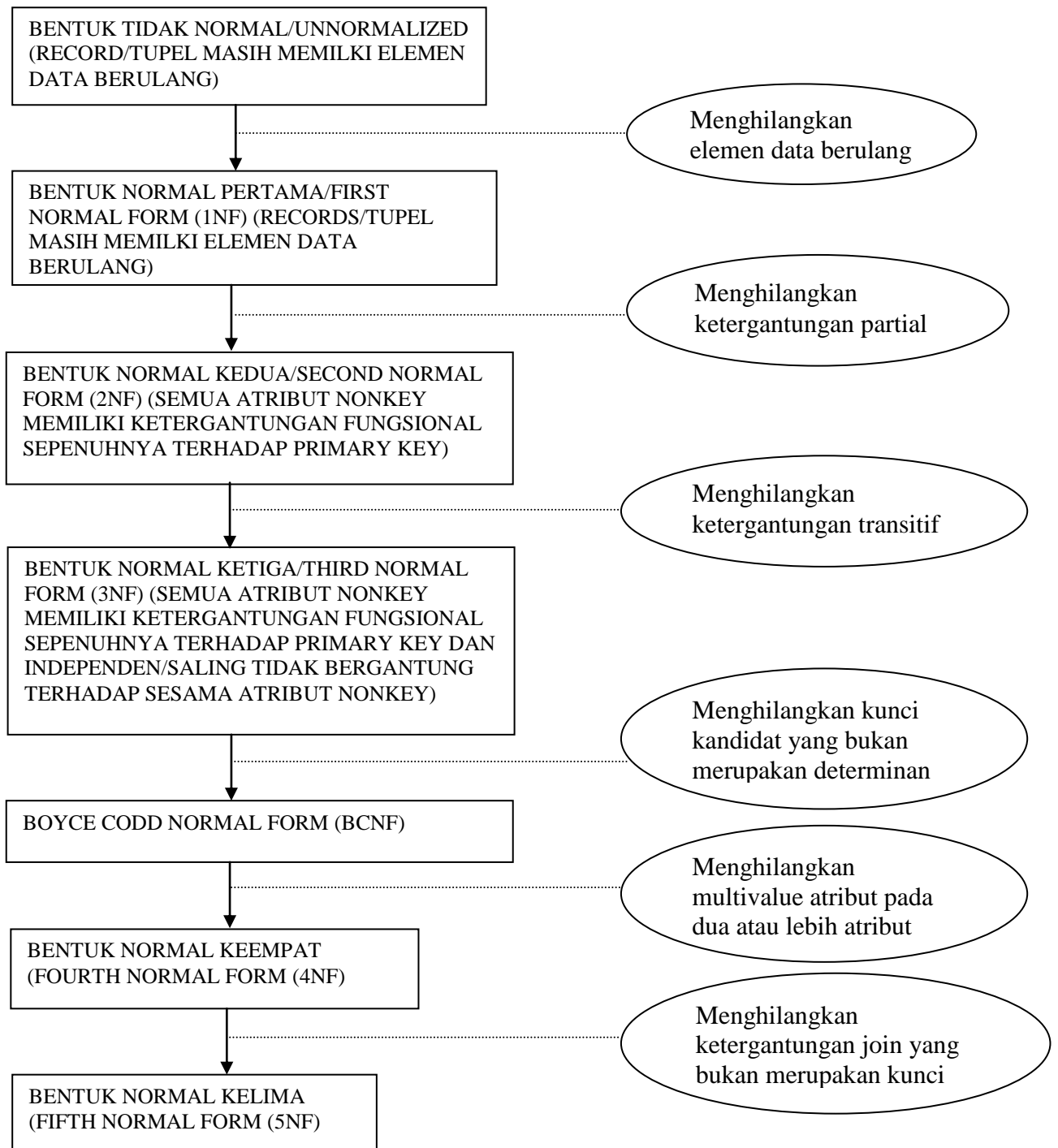


HAL YANG PENTING DIINGAT DI DALAM TAHAPAN NORMALISASI



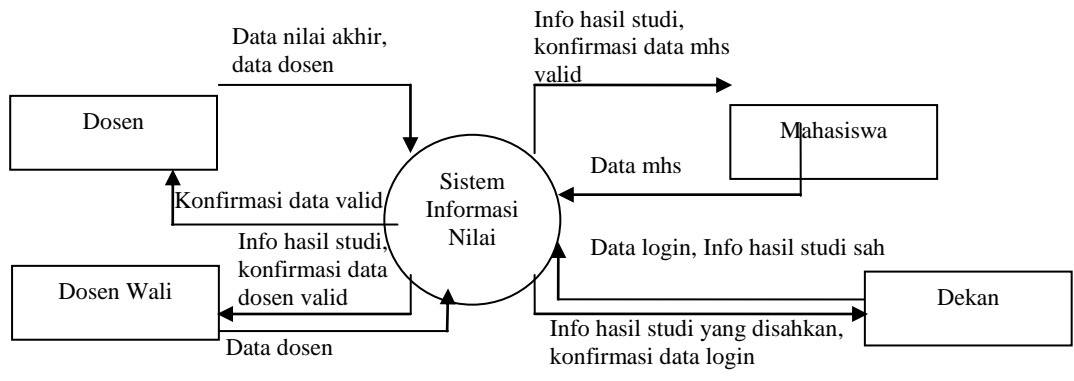
Sistem Pengolahan Data Nilai Mahasiswa Berbasis WEB

Contoh kasus pengolahan data nilai mahasiswa MI yang telah dibahas pada pertemuan sebelumnya (contoh kasus 2) dimana sistem informasi pengolahan data nilai mahasiswa yang diusulkan berbasis desktop dengan konsep stand alone atau single user, maka untuk analisis sistem informasi yang berjalan (analisis dokumen, analisis prosedur) tidak dijelaskan kembali, silahkan lihat kembali di contoh kasus 2. Dalam pembahasan ini sistem informasi yang diusulkan untuk contoh kasus tersebut diubah berbasis web. Dalam sistem informasi berbasis web tidak terdapat aliran dokumen, maka tidak perlu digambarkan aliran dokumen menggunakan flowmap. Ingat flowmap digunakan untuk menggambarkan aliran dokumen yang ada di sistem. Dokumen yang dimaksud disini berbentuk dokumen fisik, sedangkan di web semua data dalam bentuk digital.

Dalam sistem informasi yang diusulkan ini sekretariat berfungsi sebagai administrator/admin yang bertugas sebagai pengelola sistem yang dapat melakukan pengelolaan dan pengaturan sistem. Input awal ke sistem informasi terdiri dari 4 sumber yaitu dosen, dosen wali, mahasiswa dan dekan yang selanjutnya disebut dengan user. Dikarenakan terdapat dua pengguna yaitu user dan administrator dengan hak akses yang berbeda, maka antarmuka/interface pertama berupa halaman web untuk administrator dan antarmuka yang kedua yaitu halaman web untuk user.

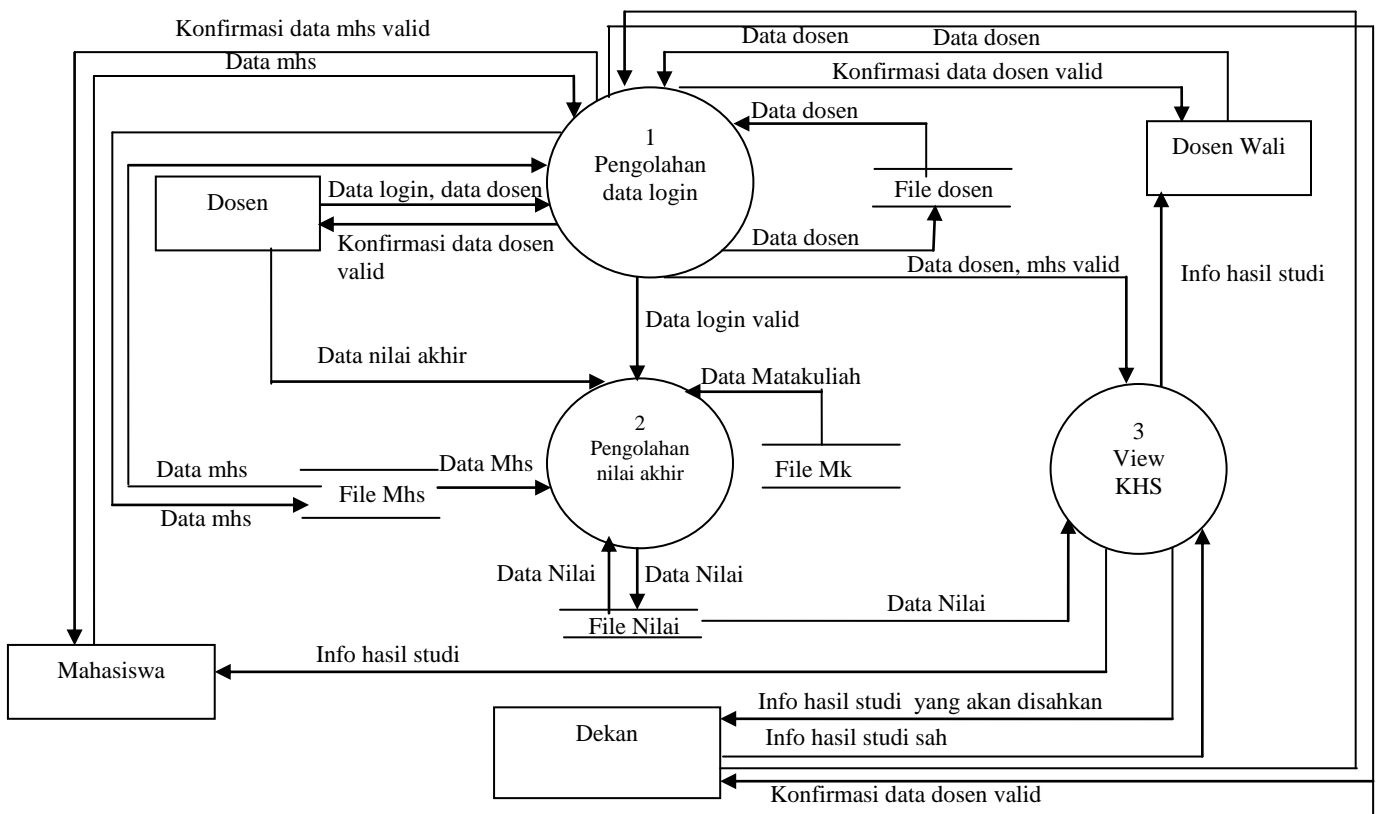
Diagram Konteks yang diusulkan

Sebagai user dari sistem informasi yang diusulkan masih sama dengan sistem informasi yang berjalan yaitu dosen, dosen wali, mahasiswa, dekan. Setiap user dibedakan dengan adanya pembacaan login yang dimiliki oleh masing-masing user, dimana login ini dibaca dari NIP untuk dosen, dekan. Sedangkan mahasiswa dibaca dari NIM.



Data Flow Diagram (DFD) Level 1

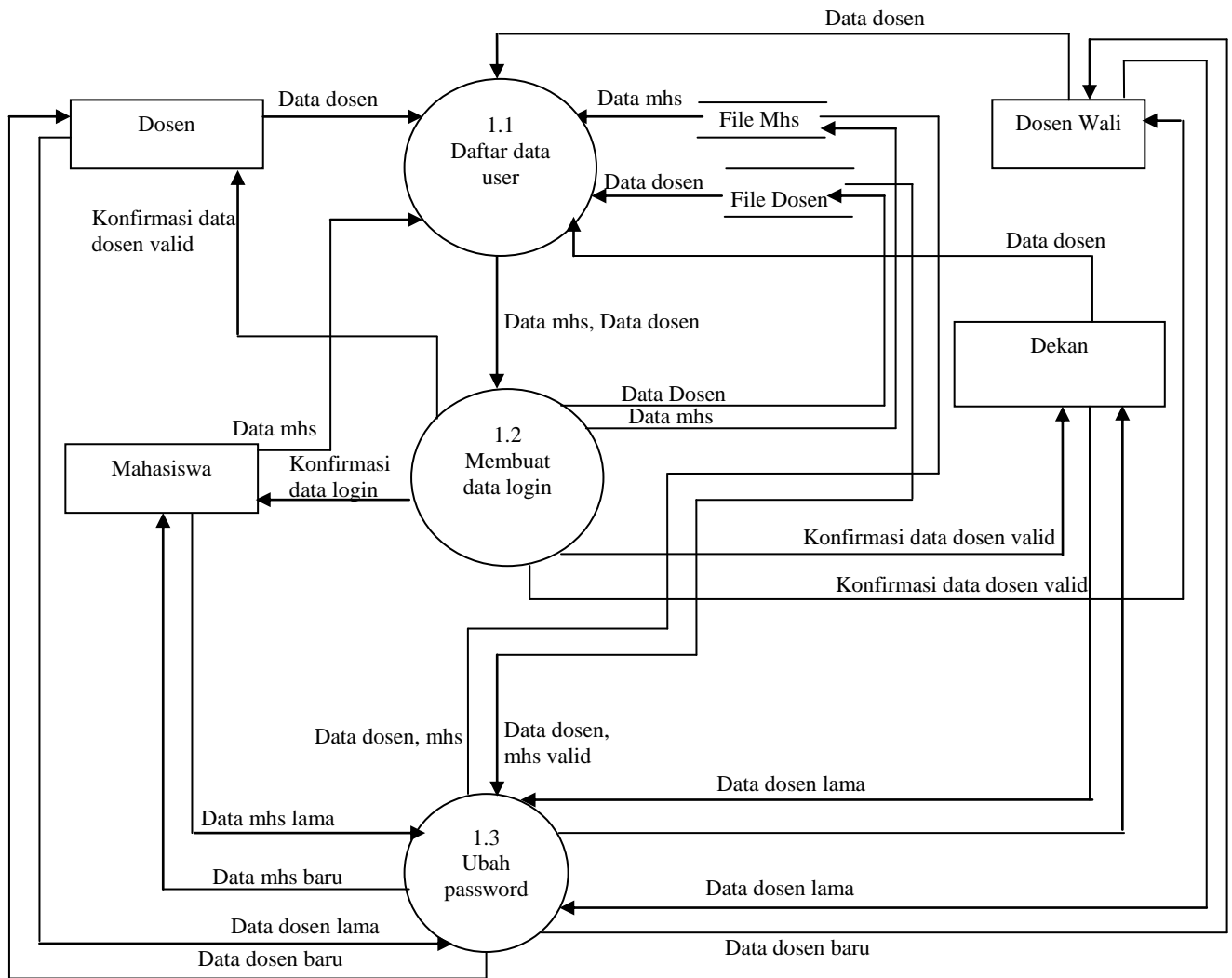
DFD Level 1 terdiri dari 3 proses pengolahan login, pengolahan data nilai akhir, view KHS dimana mahasiswa dan dosen wali hanya dapat read kemajuan studi dari mahasiswa dan dekan hanya melakukan perubahan status nilai menjadi sah/valid.



Data Flow Diagram (DFD) Level 2 Proses 1

Di DFD level 2 proses 1 yaitu pengolahan data login dijelaskan fasilitas di program aplikasi terkait proses tersebut yaitu proses daftar data user merupakan proses dimana user mendaftarkan

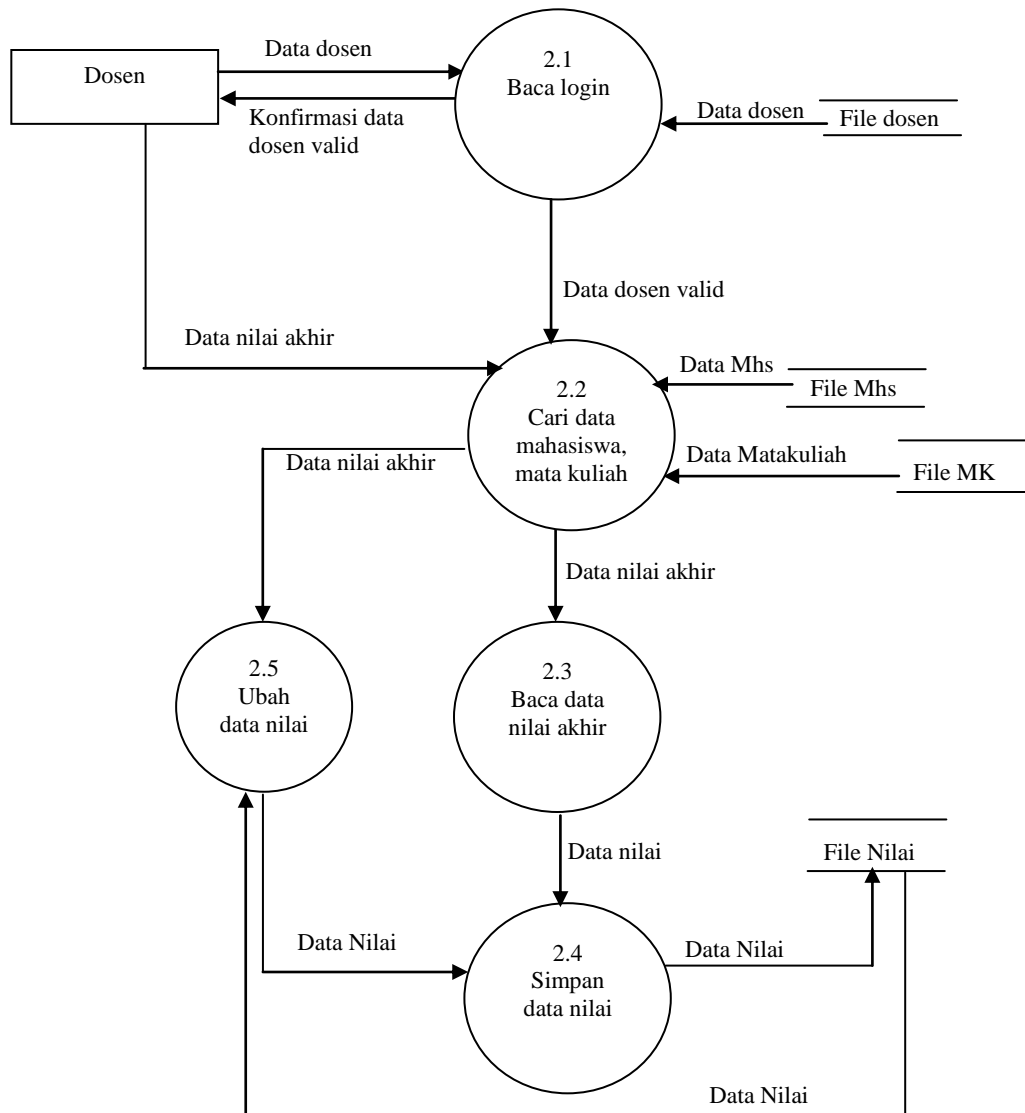
identitasnya apakah dosen atau mahasiswa untuk mendapatkan data login masing-masing, proses membuat login yaitu pembuatan data login untuk masing-masing user yang dilakukan jika data dari proses daftar user valid, proses ubah data login yaitu mengubah data login sesuai dengan keinginan masing-masing user.



Data Flow Diagram (DFD) Level 2 Proses 2

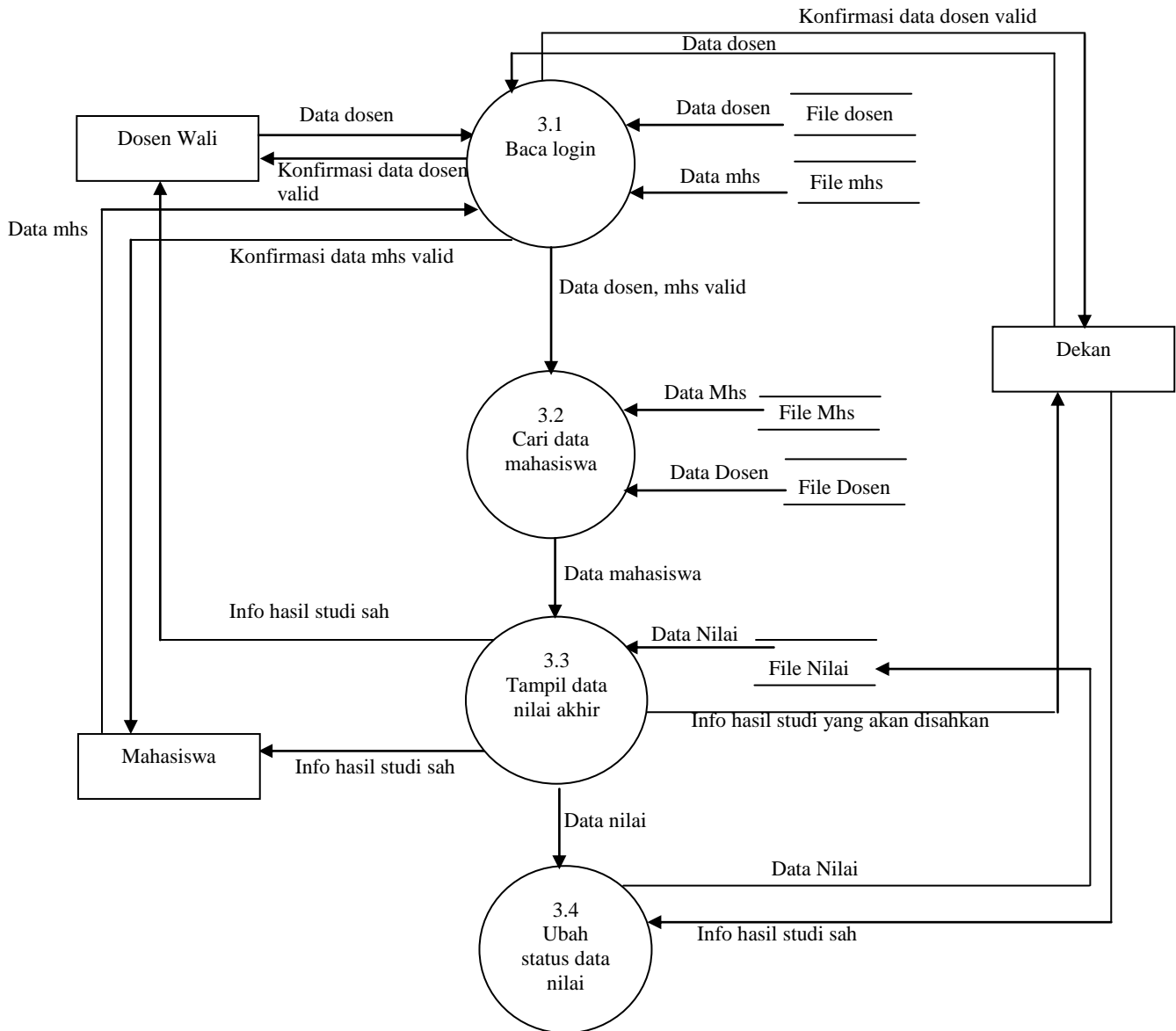
Di DFD level 2 yaitu pengolahan data nilai akhir yang terdiri dari proses pembacaan data nilai mahasiswa, proses ubah data nilai mahasiswa yang semua proses tersebut dilakukan oleh dosen.

Disini tidak diberikan fasilitas hapus data nilai karena data nilai sebelumnya disimpan sebagai data history.



Data Flow Diagram (DFD) Level 2 Proses 3

Di DFD level 2 yaitu view KHS yang terdiri dari proses pembacaan data login masing-masing user, proses cari data nilai mahasiswa berdasarkan dosen wali, tampil data nilai akhir masing-masing mahasiswa, validasi data nilai akhir oleh dekan dimana disini hanya mengubah status nilai mahasiswa menjadi valid atau sah.



KAMUS DATA (diturunkan dari aliran data dari tiap proses di level terakhir, dalam contoh ini DFD level terakhir berada di level 2, ingat struktur data yang ada dikamus data merupakan calon atribut yang nantinya akan ada di data base/masuk ke proses normalisasi. Acuannya dari item dianalisis dokumen sistem yang berjalan)

1. Nama Arus Data : Data Nilai

Alias : Data nilai akhir

Aliran Data : Dosen-P2, P2-F. Nilai, F. Nilai-P2, Dosen-P2.2, P2.2-P2.3, P2.3-P2.4, P2.22-P2.5, P2.4-F.Nilai, P2.4-P2.5, F.Nilai-P2.5, F.Nilai-P3.3, P3.4-F.Nilai, P3.3-P3.4

Struktur Data : kode_mk, nama_mk, sks, nip, nama_dosen, jur, prodi, kelas, fak, smt, ta, nim, nama_mhs, indeks

2. Nama Arus Data : Data MHS

Alias :Konfirmasi data mhs valid, data mhs valid, data mhs lama, data mhs baru

Aliran Data : Mahasiswa-P1.1, P1.2-Mahasiswa, P1.1-P1.2, Mahasiswa-P1.3, P1.3-F.Mahasiswa, F. Mahasiswa-P2.2, Mahasiswa-P3.1, P3.1-Mahasiswa, F. Mahasiswa-P3.2, P3.2-P3.3

Struktur Data : nim, nama_mhs, fak, jur, prodi, kelas, nip, **password_mhs (ini tidak ada di item analisis dokumen, jadi dapat ditambahkan atribut sesuai kebutuhan di program nantinya)**

3. Nama Arus Data : Data Mata Kuliah

Alias : -

Aliran Data : F.Mk-P2.2

Struktur Data : kode_mk, nama_mk, sks

4. Nama Arus Data : Data dosen

Alias : Konfirmasi data dosen valid, data dosen valid, data dosen lama, data dosen baru

Aliran Data : Dosen-P1.1, P1.1-P1.2, P1.2- Dosen, Dosen Wali-P1.1 , P1.2-Dosen Wali, Dekan-P1.1, P1.2-Dekan, P1.2-F.Dosen, F.Dosen-P1.1, P1.3-F.Dosen, F.Dosen-P1.3, P1.3-Dosen wali, P1.3-Dosen, P1.3-Dekan, F.Dosen-P2.1, P2.1-Dosen, Dosen Wali-P3.1, P3.1-Dosen Wali, P3.1-P3.2, F.Dosen-P3.2, P3.2-P3.3

Struktur Data : nip, nama_dosen, **password_dosen**

5. Nama Arus Data : Info hasil studi sah

Alias : Info hasil studi yang disahkan

Aliran Data : P3.3-Dosen Wali, P3.3-Mahasiswa, P3.3-Dekan, Dekan-P3.4

Struktur Data : nim, nama_mhs, nama_dosen, smt, ta, fak, jur, prodi, kelas, kode_mk,
nama_mk, sks, indeks, tot_sks, ipk, status

NORMALISASI

1. Langkah pertama dalam merancang basis data dengan sumber kamus data adalah membentuk tabel/skema tidak normal yaitu menggabungkan semua atribut yang ada pada kamus data dalam satu tabel/skema.

Bentuk UnNormal

MHS = { kode_mk, nama_mk, sks, nip, nama_dosen, jur, prodi, kelas, fak, smt, ta, nim, nama_mhs, indeks, nim, nama_mhs, fak, jur, prodi, kelas, nip, password_mhs, kode_mk, nama_mk, sks, nip, nama_dosen, password_dosen, nim, nama_mhs, nama_dosen, smt, ta, fak, jur, prodi, kelas, kode_mk, nama_mk, sks, indeks, tot_sks, ipk, status }

2. Langkah ke dua membentuk tabel normal/skema 1 dengan syarat menghilangkan semua atribut yang redundansi dari tabel yang belum normal.

Bentuk UnNormal

MHS = { kode_mk, nama_mk, sks, nip, nama_dosen, jur, prodi, kelas, fak, smt, ta, nim, nama_mhs, indeks, ~~nim, nama_mhs, fak, jur, prodi, kelas, nip,~~ password_mhs, ~~kode_mk, nama_mk, sks, nip, nama_dosen,~~ password_dosen, ~~nim, nama_mhs,~~ ~~nama_dosen, smt, ta, fak, jur, prodi, kelas, kode_mk, nama_mk, sks, indeks,~~ tot_sks, ipk, status }

Jadi bentuk normal yang pertama :

Bentuk Normal I

MHS = { kode_mk, nama_mk, sks, nip, nama_dosen, jur, prodi, kelas, fak, smt, ta, nim, nama_mhs, indeks, password_mhs, password_dosen, tot_sks, ipk, status }

3. Langkah ke 3 adalah membentuk tabel normal ke II, dengan syarat semua atribut bukan kunci harus bergantung sepenuhnya ke atribut kunci.

a. Menentukan atribut kunci dari tabel/skema bentuk normal 1

- b. Membagi tabel normal ke I menjadi beberapa tabel sesuai dengan banyaknya atribut kunci
- c. Menggabungkan atribut bukan kunci dengan kunci primer dengan syarat atribut bukan kunci bergantung sepenuhnya ke atribut kunci.

MHS = { nim*, nama_mhs, jur, prodi, kelas, fak, password_mhs }

DOSEN = { nip*, nama_dosen, password_dosen }

MATAKULIAH = {kode_mk*, nama_mk, sks }

- d. Sisa atribut digabungkan dengan salah satu tabel dengan syarat : sifat dari tabel tersebut merupakan tabel transaksi, tapi jika tidak ada yang bersifat transaksi, maka sisa atribut tersebut digabungkan berdasarkan kedekatan antar atribut.

Dari ketiga tabel diatas bersifat master, maka atributnya kita gabungkan berdasarkan kedekatan antar atribut yaitu tabel MATAKULIAH

MATAKULIAH = {kode_mk*, nama_mk, sks, smt, ta, indeks, tot_sks, ipk, status(ini atribut transitif yang tidak tergantung pada primary key tabel manapun) }

- e. Relasikan kedua tabel tersebut yaitu atribut kunci primer pada tabel yang sudah normal digabungkan ke tabel yang belum normal.
- f. Tabel yang sudah normal adalah tabel MHS dan tabel DOSEN.

BENTUK NORMAL II

MHS = { nim*, nama_mhs, jur, prodi, kelas, fak, password_mhs }

DOSEN = { nip*, nama_dosen, password_dosen }

MATAKULIAH = {kode_mk*, nama_mk, sks, smt, ta, indeks, tot_sks, ipk, status, nip**, nim** }

- 4. Membentuk normal ke 3 yaitu menghilangkan atribut yang transitif terhadap atribut kunci primer.

MATAKULIAH = {kode_mk*, nama_mk, sks }

KHS = { smt, ta, indeks, tot_sks, ipk, status, nip**, nim**, kode_mk** }

BENTUK NORMAL III

MHS = { nim*, nama_mhs, jur, prodi, kelas, fak, password_mhs }

DOSEN = { nip*, nama_dosen, password_dosen }

MATAKULIAH = {kode_mk*, nama_mk, sks }

KHS = { smt, ta, indeks, tot_sks, ipk, status, nip**, nim**, kode_mk** }

5. Suatu relasi disebut memenuhi bentuk normal Boyce Codd jika dan hanya jika suatu penentu (determinan) adalah kunci kandidat (atribut yang bersifat unik, ambil saja foreign key sebagai kunci kandidat).

NIP →	TA (x)	NIM →	TA (x)	kode_mk →	TA (x)
	index (x)		index (x)		index (x)
	tot_sks (x)		tot_sks (x)		tot_sks (x)
	IPK (x)		IPK (x)		IPK (x)
	NIM (x)		NIP (v)		NIM (x)
	kode_mk (x) status (x)		kode_mk (x) status (x)		NIP (x) status (x)

BENTUK NORMAL BOYCE Codd

MHS = { nim*, nama_mhs, jur, prodi, kelas, fak, password_mhs }

DOSEN = { nip*, nama_dosen, password_dosen }

MATAKULIAH = { kode_mk*, nama_mk, sks }

KHS = { smt, ta, tot_sks, index, ipk, status, nim**, kode_mk** }

WALI = { NIM**, NIP** }

6. Langkah selanjutnya, menentukan apakah pada tabel KHS masih ada atribut yang bernilai banyak lebih dari satu atribut, kalau masih ada maka rubah ke bentuk normal ke 4.

KHS = { smt, ta, tot_sks, ipk, nim** }

NILAI = { index, nim**, kode_mk** }

BENTUK NORMAL 4

MHS = { nim*, nama_mhs, jur, prodi, kelas, fak, password_mhs }

DOSEN = { nip*, nama_dosen, password_dosen }

MATAKULIAH = { kode_mk*, nama_mk, sks }

KHS = { smt, ta, tot_sks, ipk, nim** }

NILAI = { index, nim**, status, kode_mk** }

WALI = { NIM**, NIP** }

Ket : * = primay key

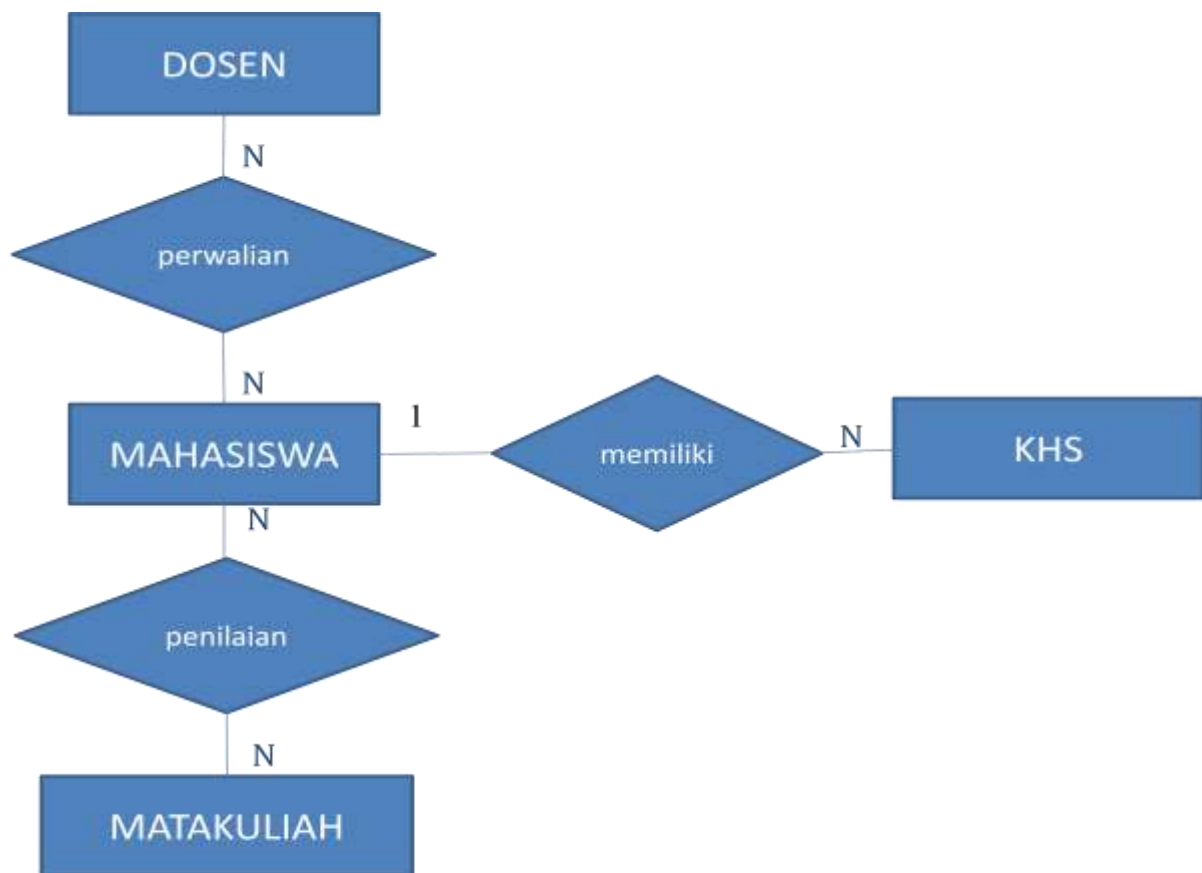
**** = foreign key**

Entity Relationship Diagram (ERD)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data karena hal ini relatif kompleks. Dengan ERD kita mencoba menjawab pertanyaan seperti :

- Data apa yang diperlukan ?
- Bagaimana data yang satu berhubungan dengan yang lain ?

Gambar 7 berikut ini merupakan ERD berdasarkan hasil normalisasi yang dilakukan sebelumnya.



Gambar 7. Entity Relationship Diagram

Kamus Data :

MHS = { nim*, nama_mhs, jur, prodi, kelas, fak, password_mhs }

DOSEN = { nip*, nama_dosen, password_dosen }

MATAKULIAH = { kode_mk*, nama_mk, sks }

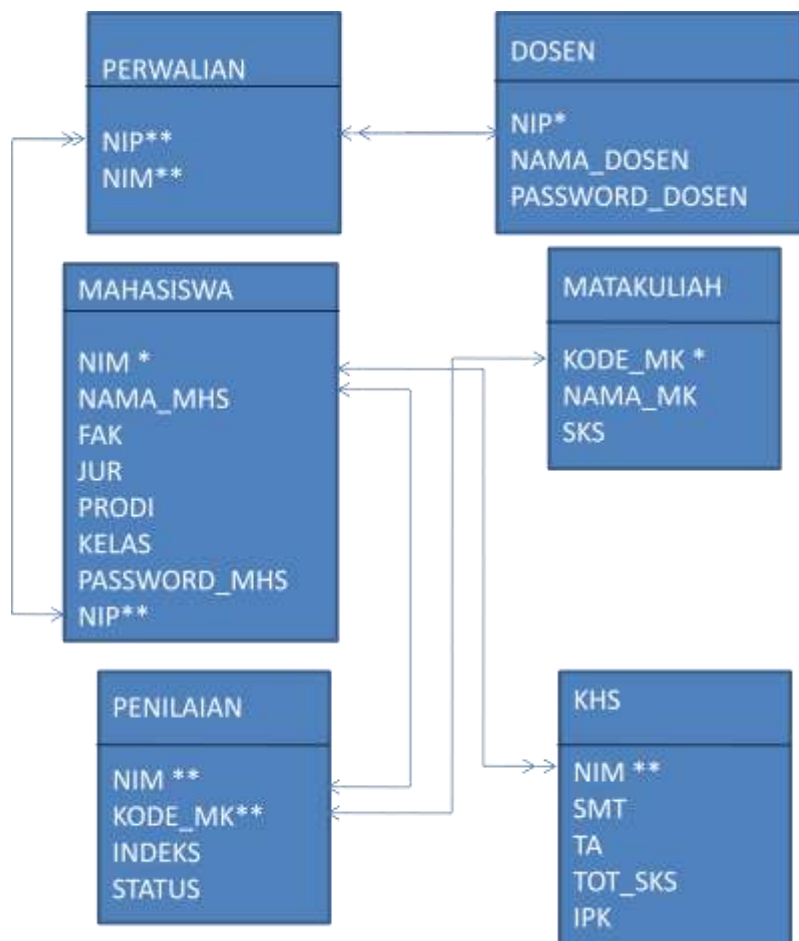
KHS = { smt, ta, tot_sks, ipk, nim** }

NILAI = { index, nim**, status, kode_mk** }

WALI = { NIM**, NIP** }

Relasi Tabel

Berdasarkan ERD yang dibuat, maka dibuat rancangan relasi table dari kasus tersebut yang terdapat pada gambar 8.



Gambar 8. Relasi Tabel

Spesifikasi Basis Data

Spesifikasi basis data dari kelima relasi yang telah dilakukan normalisasi tersebut adalah :

1. Nama File : Mahasiswa
 Media : Harddisk
 Isi : Data induk mahasiswa
 Organisasi : Index
 File
 Primary key : NIM
 Tipe file : File induk
 Panjang : 94 byte
 record
 Jumlah : 94000 (diasumsikan ada 1000 record yang akan berinteraksi di dalam file record tersebut, sehingga jumlah recordnya adalah 94 x1000)

Struktur File

No	Nama Field	Jenis	Lebar	Desimal	Keterangan
1	NIM	Character	10	0	Nomor Induk Mahasiswa, sebagai primary key
2	Nama_mhs	Character	20	0	
3	Fak	Character	30	0	Fakultas
4	Jur	Character	20	0	Jurusan
5	Prodi	Character	2	0	
6	Kelas	Numeric	4	0	
7	Password_mhs	Character	8	0	Password login mahasiswa ke aplikasi

Catatan : panjang record diperoleh dari jumlah total lebar, dimana 1 digit memakan size penyimpanan 1 byte.

jumlah record diambil dari jumlah data yang ada di dokumen, namun pada contoh ini jumlah record diasumsikan.

Organisasi file index analoginya sama seperti anda membaca buku literature atau kamus bahasa. Jika mencari satu istilah anda tidak perlu membaca dari halaman pertama sampai dengan halaman terakhir, namun cukup buka halaman table of indexnya otomatis akan diarahkan ke halaman istilah yang dimaksud berada. Kalau sequential, anda harus baca dari halaman pertama sama halaman terakhir. Oleh karena itu sequential tidak disarankan untuk tabel dengan jumlah record data yang relative banyak. Sebaliknya index tidak disarankan untuk tabel dengan ukuran jumlah record data yang sedikit, karena justru menyebabkan pencarian data sangat lama.

2. Nama File : Dosen
 Media : Harddisk
 Isi : Data induk dosen
 Organisasi : Index
 File
 Primary key : NIP
 Tipe file : File induk
 Panjang : 40byte
 record
 Jumlah : 800 (diasumsikan ada 20 record yang akan berinteraksi di dalam file record tersebut, sehingga jumlah recordnya adalah 40 x20)

Struktur File

No	Nama Field	Jenis	Lebar	Desimal	Keterangan
1	NIP	Character	7	0	Nomor Induk Pegawai dosen yang menjadi dosen wali, sebagai primary key
2	Nama_dosen	Character	25	0	golongan
3	Password_dosen	Character	8	0	Password login dosen ke aplikasi

3. Nama File : Matakuliah
Media : Harddisk
Isi : Daftar matakuliah yang ada di jurusan manajemen informatika di setiap prodi
Organisasi : Index
File
Primary key : Kode_mk
Tipe file : File induk
Panjang : 31 byte
record
Jumlah : 3100 (diasumsikan ada 100 record yang akan berinteraksi di dalam file record tersebut, sehingga jumlah recordnya adalah 31 x100)

Struktur File

No	Nama Field	Jenis	Lebar	Desimal	Keterangan
1	Kode_mk	Character	10	0	Kode matakuliah, sebagai primary key
2	Nama_mk	Character	20	0	Nama matakuliah
3	SKS	Numeric	1	0	Beban matakuliah

4. Nama File : Perwalian
Media : Harddisk
Isi : Data dosen yang menjadi dosen awali mahasiswa
Organisasi : Sequential
File
Primary key : -
Tipe file : File transaksi
Panjang : 17 byte
record
Jumlah : 340 (diasumsikan ada 20 record yang akan berinteraksi di dalam file record tersebut, sehingga jumlah recordnya adalah 20 x 17)

Struktur File

No	Nama Field	Jenis	Lebar	Desimal	Keterangan
1	NIP	Character	7	0	Nomor Induk Pegawai dosen yang menjadi dosen wali, sebagai foreign key
2	NIM	Character	10	0	Nomor Induk Mahasiswa, sebagai foreign key

5. Nama File : KHS
 Media : Harddisk
 Isi : Data hasil studi mahasiswa setiap semester tiap tahun ajaran
 Organisasi : Sequential
 File
 Primary key : -
 Tipe file : File transaksi
 Panjang : 27 byte
 record
 Jumlah : 27000 (diasumsikan ada 1000 record yang akan berinteraksi di dalam file record tersebut, sehingga jumlah recordnya adalah 27 x1000)

Struktur File

No	Nama Field	Jenis	Lebar	Desimal	Keterangan
1	NIM	Character	10	0	Nomor Induk Mahasiswa, sebagai foreign key
2	SMT	Numeric	2	0	Semester pengambilan matakuliah
3	TA	Character	9	0	Tahun ajaran di semester pengambilan matakuliah
4	Tot_SKS	Numeric	2	0	Jumlah keseluruhan pengambilan SKS di suatu semester
5	IPK	Numeric	4	2	Indeks Prestasi Kumulatif

6. Nama File : Penilaian
 Media : Harddisk
 Isi : Data nilai mahasiswa setiap mata kuliah yang ditempuh di suatu semester pada tiap tahun ajaran berupa huruf mutu
 Organisasi : Sequential
 File
 Primary key : -
 Tipe file : File transaksi
 Panjang : 26 byte
 record
 Jumlah : 26000 (diasumsikan ada 100 record yang akan berinteraksi di dalam file record tersebut, sehingga jumlah recordnya adalah 26x1000)

Struktur File

No	Nama Field	Jenis	Lebar	Desimal	Keterangan
1	NIM	Character	10	0	Nomor Induk Mahasiswa, sebagai foreign key
2	Kode_mk	Character	10	0	Kode matakuliah, sebagai foreign key
3	Index	Character	1	0	Index setiap mata kuliah yang ditempuh di suatu semester pada tahun ajaran berupa huruf mutu
4	Status	Character	5	0	Status yang menyatakan nilai valid/sah atau tidak

Enam struktur file dalam spesifikasi basis data tersebut yang akan digunakan oleh seorang programmer untuk merancang atau membuat file dengan menggunakan salah satu software berbasis database yang tersedia. Jadi jenis data yang ada di perancangan ini mengikuti aturan tipe data dari DBMS yang dipilih, apakah Oracle, MY SQL, Access.