BAB II

TEORI PENUNJANG

2.1. Definisi dan Komponen Sistem Informasi

Sistem terdiri dari komponen-komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan. Informasi merupakan proses lebih lanjut dari data dan memiliki nilai tambah, yang memenuhi syarat-syarat : lengkap, akurat, relevan, dan tepat waktu. Dengan demikian, sistem informasi dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. (Leman, 1998:3)

Menurut Leman (1998), komponen sistem informasi terdiri dari :

- **Hardware**, terdiri dari komputer,dan periferal (printer)
- **Software**, merupakan kumpulan dari perintah / fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu
- **Data**, merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.
- **Manusia**, yang terlibat dalam komponen manusia, seperti operator
- **Prosedur**, seperti dokumentasi prosedur / proses sistem, buku penuntun operasional (aplikasi) dan teknis

2.2. Siklus Hidup Pengembangan Sistem Informasi

Tahapan pengembangan sistem informasi, menurut Leman (1998) terdiri dari :

- **Survei Sistem**, bertujuan untuk mengetahui ruang lingkup pekerjaan
- **Analisis Sistem**, bertujuan untuk memahami sistem yang ada, mengidentifikasi masalah dan mencari solusinya
- **Desain Sistem**, bertujuan mendesain sistem yang baru yang dapat menyelesaikan masalah-masalah yang dihadapi.
- **Pembuatan Sistem**, membuat sistem baru (hardware dan / atau software)
- **Implementasi Sistem**, bertujuan untuk mengimplementasikan sistem yang baru
- **Pemeliharaan Sistem**, bertujuan agar sistem dapat berjalan secara optimal

Tahap-tahap pengembangan sistem informasi di atas dapat dijelaskan sebagai berikut :

❑ Survei Sistem

Survei sistem merupakan proses yang meliputi kegiatan identifikasi kebutuhan pengguna, definisi ruang lingkup pekerjaan dan penyusunan studi kelayakan. (Leman,1998:52)

Identifikasi kebutuhan pengguna (user requirements) dilaksanakan dengan mengunjungi divisi atau klien yang bersangkutan. Pendefinisian ruang lingkup mempunyai tujuan untuk mengetahui ruang lingkup dari aplikasi yang akan dikembangkan baik luas cakupan dari aplikasi maupun tahapan pekerjaannya.

Misalnya apakah konsultan akan mengembangkan dari awal sampai selesai atau hanya sampai tahap mendesain dan pemrograman dilakukan oleh klien. Penyususnan studi kelayakan adalah kegiatan berupa penyusunan proposal berdasarkan definisi ruang lingkup, yang pada dasarnya berisi gambaran umum dari aplikasi atau sistem yang akan dikembangkan.

- **Analisa Sistem**

Analisa sistem dapat diartikan sebagai suatu proses untuk memahami sistem yang ada dengan menganalisa jabatan dan uraian tugas (Busines Users / Pelaksana Bisnis), Proses Bisnis (Business Process), ketentuan atau aturan yang ada (Business Rules / Aturan Bisnis), Masalah Bisnis, Business Tools dan Rencana Bisnis (Leman, 1998:56)

Pelaksana Bisnis adalah personel yang terlibat di dalam sistem yang akan dikembangkan. Dengan menganalisa uraian tugas masing-masing jabatan atau pelaku tersebut dapat diketahui tugas dari masing-masing pelaku bisnis.

Proses Bisnis menggambarkan rangkaian tugas yang harus diselesaikan menurut aturan-aturan tertentu untuk mendapatkan suatu hasil.

Aturan Bisnis adalah aturan, batasan atau ketentuan yang menjaga integritas atau keabsahan data yang diberlakukan oleh perusahaan atau pemakai sistem agar sistem tersebut dapat berjalan seperti yang diharapkan

Masalah Bisnis adalah identifikasi masalah-masalah yang ada yang berkaitan dengan sistem yang dikembangkan untuk dicari solusinya.

Business Tools adalah peralatan yang digunakan oleh pelaku bisnis dalam menjalankan sistem.

Rencana Bisnis merupakan tindak lanjut dari keseluruhan analisa sistem, yang dalam jangka pendek berarti teratasinya permasalahan yang ada, dan pada jangka menengah serta panjang merupakan rencana pengembangan sistem dan perusahaan.

- **Desain Sistem**

Desain sistem adalah penuntun programmer dalam mengembangkan aplikasi; mencakup desain hardware, software, database, aplikasi dan gambaran / uraian tugas. (Leman,1998:71)

- **Pembuatan Sistem**

Pembuatan Sistem adalah tahap dalam pengembangan sistem informasi yang mencakup pembuatan database, program aplikasi dan buku petunjuk. (Leman,1998:105)

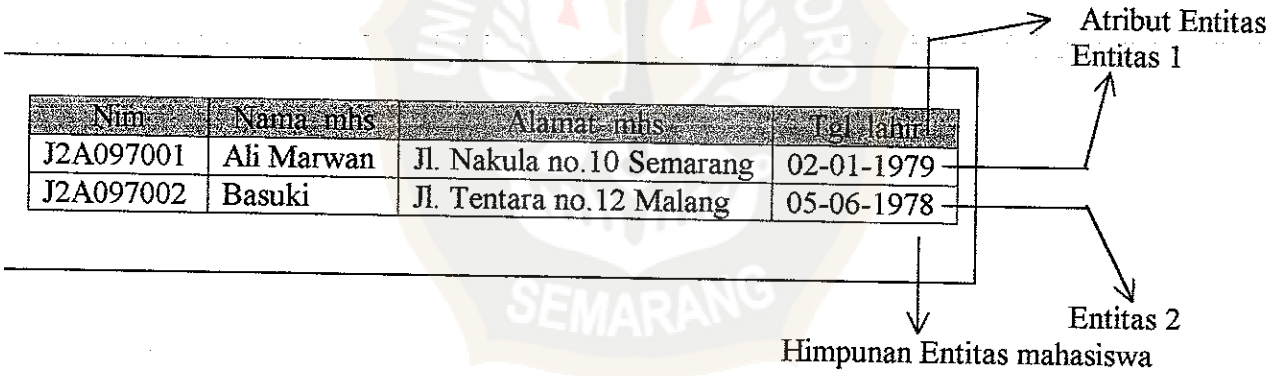
- **Implementasi Sistem**

Implementasi Sistem adalah tahapan yang meliputi proses persiapan sistem, konversi sistem, pelatihan, pengujian sistem dan pengoperasian sistem. (Leman,1998:110)

Persiapan sistem dilakukan sebelum konversi sistem dilakukan, berupa pengadaan dan atau instalasi perangkat keras, serta tersedianya sarana pendukung berupa tempat / ruangan yang memadai sehingga perangkat keras dan lunak dapat berfungsi sebagaimana mestinya. Konversi sistem merupakan proses untuk mengganti sistem lama dengan sistem baru. Adapun pelatihan diperlukan untuk menyesuaikan kemampuan sumber daya manusia yang ada dengan sistem baru yang dikembangkan.

- Candidate Key yang mungkin adalah :
  - (Nama_mhs) , jika dapat dijamin tidak ada nilai yang sama untuk atribut ini
  - (NIM)
- Atribut *NIM* merupakan *primary key* untuk Himpunan Entitas Mahasiswa, karena *NIM* merupakan Entitas yang paling unik dalam Himpunan Entitas tersebut. Atribut-atribut yang lain (*Nama_mhs, Alamat_mhs, Tgl_lahir*) merupakan atribut deskriptif.

Tabel 1. Himpunan Entitas Mahasiswa



Nim	Nama_mhs	Alamat_mhs	Tgl_lahir
J2A097001	Ali Marwan	Jl. Nakula no.10 Semarang	02-01-1979
J2A097002	Basuki	Jl. Tentara no.12 Malang	05-06-1978

2.3.2.3 Relasi (*Relationship*) dan Himpunan Relasi (*Relationship Sets*)

Ditulis oleh Korth (1986), Relasi adalah hubungan dari beberapa entitas. Himpunan relasi adalah kumpulan relasi dengan tipe yang sama. Secara matematis Himpunan Relasi adalah 2 (dua) atau lebih relasi dalam himpunan-himpunan entitas. Jika E_1, E_2,....E_n adalah himpunan entitas, maka himpunan relasi R adalah himpunan bagian dari $\{(e_1,e_2, ..., e_n)| e_1 \in E_1, e_2 \in E_2, ..., e_n \in E_n\}$ $n = 1,2,...,n$; dimana $(e_1,e_2, ..., e_n)$ adalah relasi.

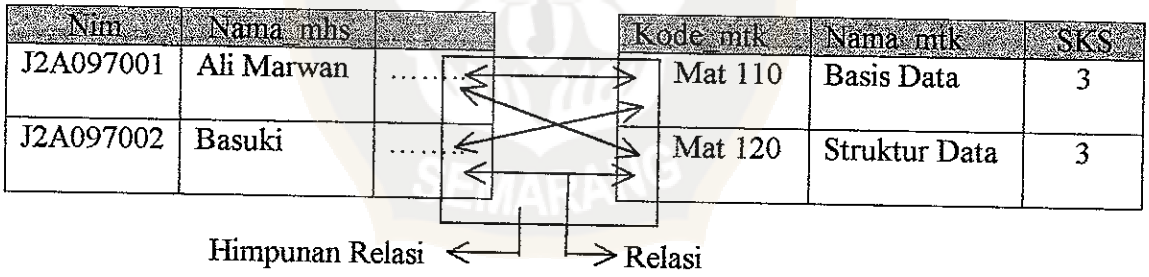
Contoh :

Diberikan himpunan entitas mahasiswa seperti pada Tabel 1 dan himpunan entitas mata kuliah sebagai berikut :

Tabel 2. Himpunan Entitas Mata Kuliah

Kode_mtk	Nama_mtk	SKS
Mat 110	Basis Data	3
Mat 120	Struktur Data	3

Relasi himpunan entitas mahasiswa dengan himpunan entitas mata kuliah digambarkan pada Gambar 2 :



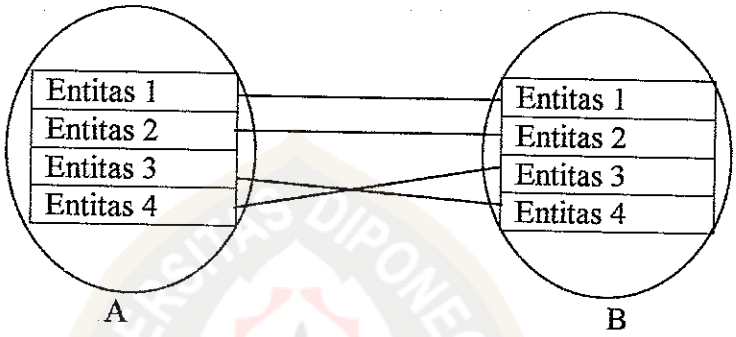
Gambar 2. Relasi dan Himpunan Relasi Antara Himpunan Entitas Mahasiswa dan Himpunan Entitas Mata Kuliah

2.3.2.4 Kardinalitas / Derajat Relasi

Kardinalitas / Derajat Relasi adalah jumlah entitas yang dapat dihubungkan dengan entitas lain menggunakan suatu relasi. Korth (1986).

Kardinalitas relasi merujuk pada hubungan maksimum yang terjadi antara 2 (dua) himpunan entitas. Kardinalitas ini dapat berupa satu ke satu (*one to one*), satu ke banyak (*one to many*), dan banyak ke banyak (*many to many*).

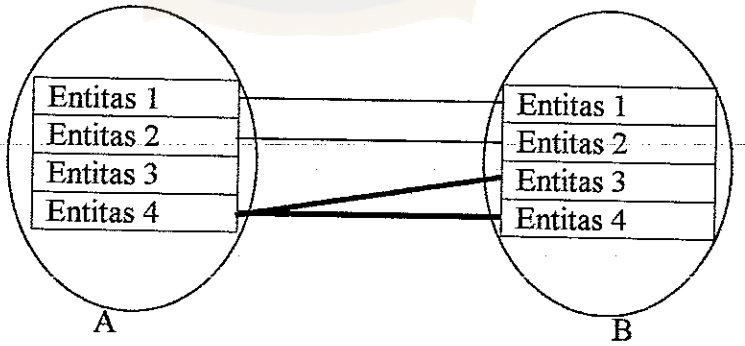
- **Satu ke Satu (One to One)**



Gambar 3. Kardinalitas Satu ke Satu

Setiap entitas pada himpunan entitas A berelasi dengan paling banyak satu entitas pada himpunan entitas B. Hal ini berlaku kebalikan.

- **Satu ke Banyak**

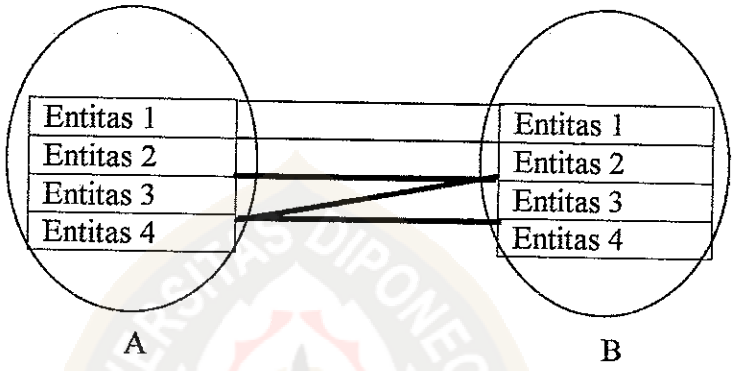


Gambar 4. Kardinalitas Satu ke Banyak

Setiap entitas pada himpunan entitas A dapat mempunyai relasi dengan lebih dari satu entitas pada himpunan entitas B. Hal ini tidak berlaku sebaliknya,

setiap entitas pada himpunan entitas B hanya diijinkan memiliki satu relasi dengan entitas pada himpunan entitas A, disebut juga dengan kardinalitas Banyak ke Satu (*Many to One*).

- **Banyak ke Banyak (*Many to Many*)**



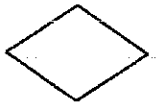
Gambar 5. Kardinalitas Banyak ke Banyak

Setiap entitas pada himpunan A dapat mempunyai relasi dengan lebih dari satu entitas pada himpunan entitas B, dan setiap entitas pada himpunan entitas B dapat memiliki relasi dengan lebih dari satu entitas pada himpunan entitas A.

Model *entity-relationship* dapat digambarkan dengan lebih sistematis menggunakan diagram *entity-relationship* (diagram E-R). Notasi-notasi atau simbol-simbol yang digunakan dalam diagram E-R adalah :


Persegi panjang, menyatakan himpunan entitas.

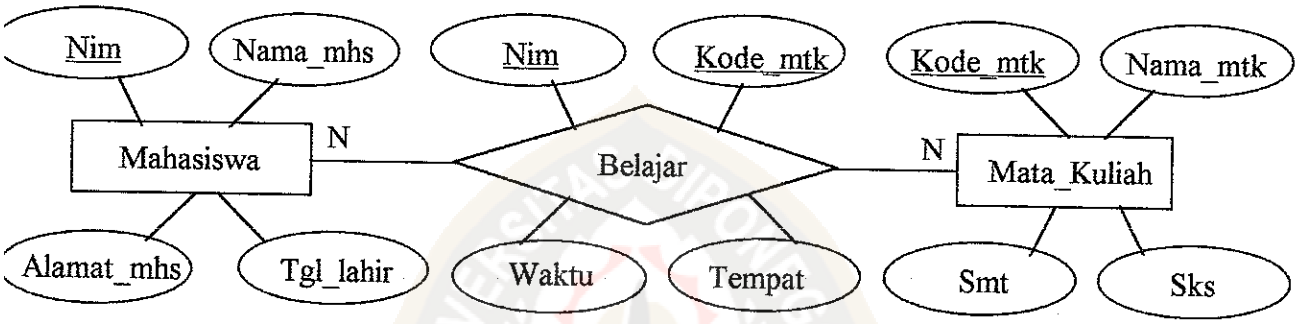

Elips / lingkaran, menyatakan atribut(atribut *key* ditunjukkan dengan garis bawah pada nama atribut).


Belah ketupat, menyatakan himpunan relasi.

Garis, sebagai garis penghubung antara himpunan entitas dengan atribut, dan himpunan relasi dengan himpunan entitas.

Kardinalitas relasi dapat ditunjukkan dengan banyaknya garis relasi, atau dengan pemakaian angka (1 dan 1 untuk relasi satu ke satu, 1 dan N untuk relasi satu ke Banyak, serta N dan N untuk relasi banyak ke banyak).

Contoh diagram E-R :



Gambar 6. Diagram E-R untuk Relasi Banyak ke Banyak

Gambar 6 dapat juga disajikan dengan menampilkan daftar atribut dalam bentuk kamus data, menjadi :

Kamus Data :

- Mahasiswa = {Nim, Nama_mhs, Alamat_mhs, Tgl_lahir}
- Mata_kuliah = {Kode_mtk, Nama_mtk, Sks, Smt}
- Belajar = {Nim, Kode_mtk, Waktu, Tempat}

Gambar 7. Diagram E-R dengan kamus Data

2.3.3 *Data Dictionary* (Kamus Data)

Data Dictionary (selanjutnya disebut DD) adalah perangkat pemodelan yang tidak berupa notasi grafis sebagaimana halnya DFD, tetapi berupa daftar yang berfungsi sama dengan kamus yang membantu dalam mencari arti kata baru. DD membantu pelaku sistem untuk mengerti aplikasi secara detil dan mereorganisasi semua elemen data yang digunakan dalam sistem secara presisi, sehingga pemakai dan penganalisa sistem punya dasar yang sama tentang masukan, keluaran, penyimpanan dan proses. (Pohan,1997:21)

Pohan (1997) menyatakan fungsi Data Dictionary (DD) sebagai berikut :

- Menjelaskan arti aliran data dan penyimpanan dalam DFD
- Mendeskripsikan komposisi paket data yang bergerak melalui aliran
- Mendeskripsikan komposisi penyimpanan data
- Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran
- Mendeskripsikan hubungan detil antar penyimpanan yang menjadi titik perhatian dalam entity-relationship diagram

2.3.4 Normalisasi

Normalisasi adalah proses konversi dokumen / laporan manual ke dalam struktur tabel dengan menghilangkan elemen yang sama dan data yang berulang-ulang.(Leman,1999:87)

Kristanto (1999) memberikan definisi normalisasi sebagai suatu proses pengelompokan data menjadi tabel-tabel yang menunjukkan entity dan relasinya.

Tujuan desain adalah mengkonstruksi relasi tanpa redudansi. Kondisi ini didefinisikan dalam terminologi relasi normal (normal relations). (Pohan, 1997:49)

Dalam pembahasan mengenai normalisasi, sebelumnya harus dikeenal terlebih dahulu pengertian dari *key* dan ketergantungan fungsional. Pengertian dari *key* telah dibahas pada poin 2.3.2.2.

2.3.4.1 Ketergantungan Fungsional

Fathansyah (1999) memberikan definisi Ketergantungan Fungsional (KF) sebagai berikut :

Diberikan 2 row r1 dan r2 dalam tabel T yang memiliki atribut A dan B dengan A → B , jika r1(A) = r2(A) maka r1(B) = r2(B).

Notasi A → B mempunyai arti A secara fungsional menentukan B, atau B secara fungsional tergantung pada A, jika hanya jika untuk setiap kumpulan baris data (row) yang ada dalam tabel T pasti ada dua row dalam tabel T dengan nilai A yang sama maka nilai untuk B pasti juga sama. (Fathansyah,1999:44)

Contoh Ketergantungan Fungsional:

Diberikan tabel nilai dengan atribut *nama_kul,nim,nama_mhs*, dan *indeks_nilai*.

Tabel 3. Tabel Nilai

	nama_kul	Nim	nama_mhs	indeks_nilai
Row 1	Struktur data	97001	Ali Akbar	A
Row 2	Struktur data	97004	Indah Susanti	B
Row 3	Basis data	97001	Ali Akbar	.....
Row 4	Basis data	97002	Ahmadi	.....
Row 5	Kalkulus	97002	Ahmadi	A

Dari data di atas, dengan asumsi bahwa data tersebut memadai dan dengan pertimbangan intuisi, maka KF yang dapat diajukan adalah :

- nim → nama_mhs

  yang berarti bahwa atribut nama_mhs hanya tergantung pada atribut nim. Hal ini dapat dilihat bahwa untuk setiap nilai nim yang sama maka pasti nilai nama_mhs-nya juga sama.

- nama_kul nim → indeks_nilai

  yang berarti bahwa atribut indeks_nilai tergantung pada atribut nama_kul dan nim secara bersama-sama. KF ini menunjukkan pengertian bahwa setiap indeks nilai diperuntukkan bagi mahasiswa tertentu untuk mata kuliah tertentu yang diambilnya.

2.3.4.2 Normalisasi dengan Ketergantungan Fungsional

Dalam perspektif normalisasi, sebuah basis data dapat dikatakan baik jika setiap tabel yang menjadi unsur pembentuk unsur pembentuk basis data tersebut juga telah berada dalam keadaan baik (efisien) atau normal. Adapun sebuah tabel dapat dikategorikan baik (efisien) atau normal jika telah memenuhi tiga kriteria berikut :

1. Jika ada dekomposisi (penguraian) tabel maka dekomposisinya dijamin aman (*Losless-Join Decomposition*)
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*)
3. Tidak melanggar Boyce-Code Normal Form (BCNF). Atau jika BCNF tidak terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap Ketiga (3rd *Normal Form* / 3NF)

(Fathansyah,1999:46)

Kriteria di atas adalah kriteria utama untuk mendapatkan predikat efisien / normal bagi sebuah tabel. Selain kriteria tersebut, terdapat kriteria-kriteria lain yang juga tercakup dalam kerangka normalisasi, tetapi bukan merupakan kriteria utama, yaitu :

- Bentuk Normal tahap Pertama (1st *Normal Form* / 1NF)
- Bentuk Normal tahap Kedua (2nd *Normal Form* / 2NF)
- Bentuk Normal tahap Keempat (4th *Normal Form* / 4NF)
- Bentuk Normal tahap Kelima (5th *Normal Form* / 5NF)

Dalam pembahasan tugas akhir ini, hanya akan menerapkan kriteria utama normalisasi.

- ***Losless-Join Decomposition***

  Dekomposisi adalah upaya penguraian tabel untuk memperoleh tabel yang baik. Akan tetapi, jika tidak hati-hati upaya ini justru dapat menghasilkan kesalahan. Dekomposisi yang benar terjadi jika tabel-tabel hasil dekomposisi digabungkan kembali dapat menghasilkan tabel awal sebelum didekomposisi. Dekomposisi yang benar inilah yang disebut dengan *Losless-Join Decomposition*. (Fathansyah,1999:49)

  Contoh Penerapan *Losless-Join Decomposition*:

  Diberikan sebuah tabel ABC yang didefinisikan dengan dua buah KF yaitu $A \rightarrow B$ dan $B \rightarrow C$. Kedua KF tersebut diperoleh dari pengamatan terhadap data yang kurang memadai atau karena asumsi yang kurang tepat. Pertama akan ditunjukkan bahwa apabila tidak hati-hati, dekomposisi akan menghasilkan kesalahan. Selanjutnya akan diberikan alternatif sehingga diperoleh *Losless-Join Dekomposition*.

Tabel 4. Tabel ABC

	A	B	C
Row 1	a1	100	c1
Row 2	a2	200	c2
Row 3	a3	300	c3
Row 4	a4	200	c4

Dengan isi seperti itu, pernyataan KF yang kedua B⟶C tidak sepenuhnya tepat karena pada row 2 dan row 4 untuk nilai atribut B yang sama diperoleh nilai atribut C yang berbeda. Tetapi di sini yang ingin ditekankan adalah adanya dua KF tersebut mendorong untuk mendekomposisi tabel ABC menjadi dua buah tabel yaitu Tabel AB dan Tabel BC sebagai berikut :

Tabel 5. Tabel AB

A	B
a1	100
a2	200
a3	300
a4	200

Tabel 6. Tabel BC

B	C
100	c1
200	c2
300	c3
200	c4

Apabila digabungkan kembali, maka hasilnya adalah :

Tabel 7. Tabel ABC Penggabungan Kembali

A	B	C
a1	100	c1
a2	200	c2
a2	200	c4
a3	300	c3
a4	200	c2
a4	200	c4

Hasil yang berbeda dengan Tabel 4 ini, disebut dengan *Lossy-Join Decomposition.*

Akan tetapi jika data pada row 4 di Tabel 4 diganti menjadi :

a4	200	c2

Sehingga Tabel 4 menjadi :

Tabel 8. Tabel ABC'

A	B	C
a1	100	c1
a2	200	c2
a3	300	c3
a4	200	c2

Maka kedua KF dapat dibenarkan dan hasil dekomposisi menjadi :

Tabel 9. Tabel AB'

A	B
a1	100
a2	200
a3	300
a4	200

Tabel 10. Tabel BC'

B	C
100	c1
200	c2
300	c3

Apabila digabungkan kembali maka akan diperoleh Tabel 8 yang sama. Penguraian semacam ini yang disebut dengan Dekomposisi Aman atau *Lossless-Join Decomposition*.

- **Dependency Preservation**

Fathansyah (1999) memberikan definisi dari Dependency Preservation sebagai kriteria yang harus bisa menjamin ketika terjadi perubahan data maka perubahan tersebut tidak menghasilkan inkonsistensi data yang mengakibatkan KF yang sudah benar menjadi tidak terpenuhi.

Contoh Dependency Preservation :

Jika diberikan tabel

Akademik(kode_kul,nama_kul,nim,nama_mhs,alamat,indeks_nilai)

dengan asumsi setiap mahasiswa tidak hanya mengambil satu mata kuliah dan satu mata kuliah tidak hanya diambil oleh satu mahasiswa, yang didekomposisi menjadi tabel-tabel :

Mahasiswa(nim,nama_mhs,alamat) dan

Nilai(kode_kul,nim,nama_mhs,indeks_nilai).

Misal tabel Nilai mempunyai dua buah KF yaitu :

kode_kul nim ⟶ indeks_nilai

nim ⟶ nama_mhs

Jika pada tabel Mahasiswa terdapat perubahan data pada atribut alamat, maka perubahan ini tidak perlu dijalarkan ke tabel Nilai karena di tabel Nilai atribut tersebut tidak dilibatkan. Akan tetapi jika perubahan itu terjadi pada atribut nama_mhs pada tabel Mahasiswa, maka perubahan ini harus dijalarkan pada tabel Nilai. Jika penjalaran perubahan ini hanya dilakukan pada satu baris data

(*row*) pada tabel Nilai dengan nilai nim yang sama dengan nilai nim di tabel Mahasiswa yang nama_mhs nya diubah tersebut, maka KF

nim → nama_mhs

di tabel Nilai tidak akan terpenuhi lagi. Hal ini karena perubahan ini akan mengakibatkan adanya 2 *row* di tabel Nilai yang nim-nya sama tetapi mempunyai nama_mhs yang berbeda. Sehingga untuk menghindari inkonsistensi data, penjalaran perubahan ini harus dilakukan pada seluruh *row* dengan nim yang sesuai. Permasalahannya, proses perubahan data satu per satu ini tidak efisien. Agar kriteria *Dependency Preservation* dapat terpenuhi, solusinya adalah dengan meniadakan atribut nama_mhs pada tabel Nilai.

- **Boyce Codd Normal Form**

Sebuah tabel dikatakan dalam *Boyce Codd Normal Form* (BCNF) jika untuk semua ketergantungan fungsional (KF) dengan notasi X → Y, maka X harus merupakan *superkey* pada tabel tersebut. Jika tidak demikian, maka tabel tersebut harus didekomposisi berdasarkan KF yang ada, sehingga X menjadi *superkey* dari tabel-tabel dekomposisi. (Fathansyah, 1999:52)

Contoh Penerapan BCNF :

Tabel Akademik dalam contoh *Dependency Preservation* tidak memenuhi BCNF. Sebagai bukti, jika diambil KF :

nim → nama_mhs, alamat

maka seharusnya nim merupakan *superkey* pada tabel Akademik. Pada kenyataanya nim pada tabel tersebut bersifat tidak unik. Sehingga untuk dapat

memenuhi BCNF, tabel Akademik tersebut harus didekomposisi sedemikian hingga syarat BCNF terpenuhi.

Dekomposisi yang aman dapat dilakukan dengan memilah berdasarkan KF. Pada tabel Akademik, KF yang ada :

nim ⟶ nama_mhs,alamat

kode_kul ⟶ nama_kul

nim kode_kul ⟶ indeks_nilai

Berbekal keempat KF tersebut, tabel Akademik dapat didekomposisi menjadi tabel Mahasiswa (nim,nama_mhs,alamat), Mata_Kuliah(kode_kul,nama_kul), dan Nilai(nim,kode_kul,indeks_nilai);yang masing-masing memenuhi kriteria BCNF

- **Bentuk Normal Tahap Ketiga (3rd *Normal Form*)**

Bentuk Normal Ketiga (3NF) merupakan kriteria alternatif, jika kriteria BCNF tidak dapat terpenuhi. Sebuah tabel dikatakan berada dalam Bentuk Normal tahap Ketiga (3NF), jika untuk setiap KF dengan notasi X ⟶ A, dengan A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah *superkey* pada tabel tersebut
- Atau A merupakan bagian dari key primer pada tabel tersebut

(Fathansyah,1999:57)

2.4 Microsoft Visual Basic 6.0 Enterprise Edition dan Microsoft Access 2000

Visual Basic 6.0 merupakan salah satu bahasa pemrograman berorientasi obyek. Bisual Basic memuat property, methods, dan event dari suatu obyek yang telah didefinisikan. Elemen penting dalam Visual Basic 6.0 adalah *Forms, Module*, dan *Class Module*.

- *Forms*

  Forms adalah suatu obyek yang berfungsi sebagai media komunikasi antara program dengan pemakai. Pada forms ini dapat diletakkan alat-alat kontrol seperti tombol (command button), tulisan (text box dan label), kotak gambar (picture box) dan lain-lain.

- *Module*

  Tempat penulisan program (modul-modul) yang berisi subprogram (sub dan function). Meskipun demikian kode-kode program untuk pengendali event (event handler) terletak pada obyek tersebut ditempatkan.

- *Class Module*

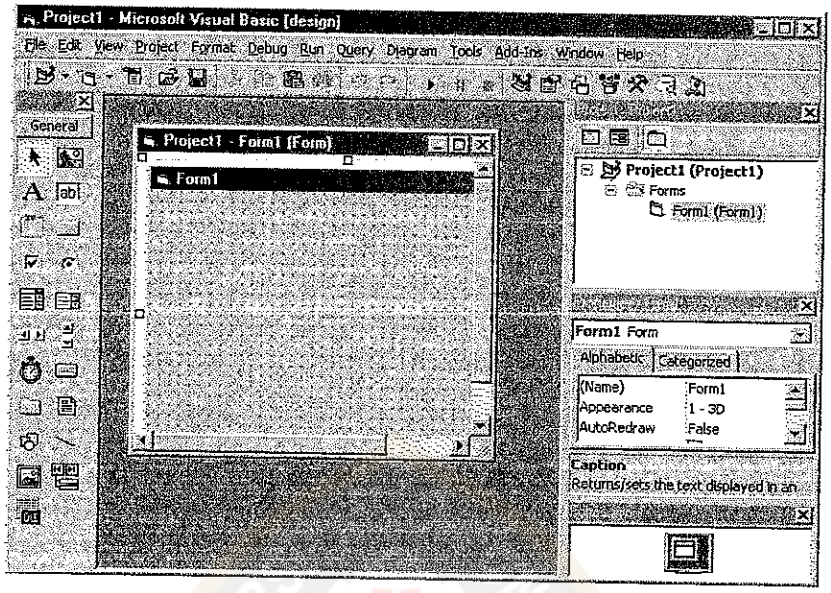
  Tempat pendefinisian obyek-obyek baru selain yang telah disediakan oleh Visual Basic 6.0

Kemampuan yang dimiliki oleh Visual Basic 6.0 seperti halnya bahasa berorientasi obyek yang lain, adalah melakukan *link* dan *Embedding* atau OLE dan kontrol ActiveX, sehingga suatu program dapat mengakses program lain meskipun tidak ada hubungan atau kaitannya dengan program yang sedang dijalankan. Selain itu juga dapat digunakan untuk pemrograman basis data.

Hardware requirements untuk menjalankan Visual Basic; dengan semua hardware berada dalam daftar *hardware* kompatibel Microsoft Windows 95 ke atas atau Windows NT 4.0 ke atas; adalah sebagai berikut :

Tabel 5. Hardware Requirements Visual Basic 6.0 Enterprise Edition

Computer / Processor	PC dengan processor 486 DX 66 MHz atau yang lebih tinggi. Pentium atau processor yang lebih tinggi direkomendasikan
Memory	16 MB RAM untuk Windows 95 dan berikutnya (direkomendasikan 32 MB); 24 MB untuk Windows NT 4.0 (direkomendasikan 32 MB)
Hard Disk	VB 6.0 : 76 MB typical; 94 MB Maximum IE : 43 MB typical; 59 MB maximum MSDN : 57 MB typical; 493 maximum Windows NT 4.0 Option Pack : 20 MB Windows 95 ke atas; 200 MB Windows NT 4.0
Drive	CDROM drive
Display	VGA atau resolusi yang lebih tinggi; SVGA direkomendasikan
Operating System	OS Microsoft Windows 95 ke atas, atau OS Microsoft Windows NT 4.0 dengan service pack 3 ke atas
Peripheral	Microsoft Internet Explorer 4.01 service pack 1 Mouse



Gambar 8. Lingkungan Microsoft Visual Basic 6.0

Adapun Microsoft Acces 2000 adalah Sistem Manajemen Basis Data yang merupakan grup dari Microsoft Office 2000 produk dari Microsoft, dengan database engine yang disebut Microsof Jet.