BAB II
DASAR TEORI

Basis data adalah kumpuan data yang saling berelasi dan disimpan dengan tetap dalam sebuah komputer. Sedangkan sistem manajemen basis data (SMBD) adalah perangkat lunak yang digunakan oleh satu orang atau lebih untuk memanfaatkan dan atau memodifikasi data dalam basis data. Hal tersebut dikemukakan oleh Kroenke [1988].

Berdasarkan definisi diatas, dapat diketahui bahwa satu sistem manajemen basis data (SMBD) berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Set program pengelola berfungsi untuk menambah, menghapus, dan memperbaiki/memperbaharui data. Pemanipulasian data ini dapat dilakukan pada/dari :

- Tabel lokal, tempat data disimpan dan diambil pada komputer yang sama. Ini dapat digunakan oleh satu atau banyak pemakai pada satu komputer saja.
- Tabel yang terletak pada komputer lain, tabel ini dapat digunakan oleh banyak pemakai.

Basis data adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu basis data menunjukkan satu kumpulan data yang dipakai dalam satu lingkup instansi atau institusi.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.1 Kegunaan Sistem Basis Data

Menurut Korth [1986] penyusunan basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu:

2.1.1 Redudansi dan inkonsistensi data

Jika file-file dalam program aplikasi diciptakan dengan konsep yang berbeda-beda, maka ada beberapa bagian data yang mengalami penggandaan. Penggandaan atau kerangkapan data dapat menyebabkan timbulnya masalah dalam pengaksesan data, dan menyebabkan data tidak konsisten.

2.1.2 Kesulitan pengaksesan data

Pada sistem basis data terdapat data-data fleksibel yang dimaksudkan/ditujukan untuk memberikan kemudahan dalam menampilkan kembali data-data yang dipilih dan diperlukan dalam basis data dan merepresentasikan data-data tersebut dalam format yang berbeda.

2.1.3 Banyak pemakai (*multiple user*)

Sistem basis data yang dibuat dengan tujuan untuk dimanfatkan oleh banyak pemakai. Baik untuk memperbaharui data ataupun sekedar memanfaatkan informasi yang terdapat didalamnya.

2.1.4 Masalah keamanan (*security*)

Keamanan data diperlukan untuk melindungi data terhadap akses yang tidak berhak dari pihak-pihak yang tidak berwenang yang bermaksud merugikan atau bahkan merusak data.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.1.5 Masalah kesatuan (*integrasi*)

Sistem basis data berisi file-file yang saling berkaitan. Pemrograman basis data memungkinkan hubungan antar file maupun antar data dapat berlangsung dengan baik melalui field kunci yang mengaitkan file-file tersebut.

2.1.6 Masalah kebebasan data (*data independence*)

Jika data dikelola dengan baik sebagai sumber daya yang independen, maka pemisahan file-file data dari program-program aplikasi merupakan hal yang sangat penting.

Wilson price [1988] menulis sistem manajemen basis data yang terbentuk mempunyai kemampuan untuk :

a. Menetapkan dan menambah file data baru dalam sistem basis data
b. Menambah data dalam file data
c. Memudahkan pengaksesan data secara langsung dengan banyak cara
d. Menampilkan laporan berdasar basis data
e. Menciptakan cara untuk memperbaharui data
f. Memelihara kesatuan dan kekonsistenan data

Fatansyah [1999] mengungkapkan perancangan basis data dapat dilakukan dengan menerapkan normalisasi terhadap struktur tabel yang diketahui atau dengan langsung membuat model *Entity-Relationship*. Masing-masing cara akan dibahas sebagai berikut :

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.2 Menerapkan Normalisasi Terhadap Struktur Tabel yang Telah Diketahui

Normalisasi merupakan cara pendekatan dalam membangun desain lojik basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel normal.

Kriteria tabel normal adalah :

1. Jika ada dekomposisi (penguraian tabel), maka dekomposisinya harus dijamin aman (*Losless-Join Decomposition*).
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*)
3. Tidak melanggar *Boyce-Code Normal Form* (BCNF)

Jika kriteria ketiga tidak dapat terpenuhi, maka paling tidak, tabel tersebut tidak melanggar Bentuk Normal tahap Ketiga (*3rd Normal Form* / 3NF)

Dalam pembahasan tentang normalisasi basis data digunakan istilah-istilah seperti atribut, *key*, domain dan ketergantungan fungsional, dalam hal ini akan dijelaskan terlebih dahulu pengertian dari istilah-istilah tersebut.

2.2.1 Atribut Tabel

Atribut sebenarnya identik dengan istilah kolom data. Atribut berfungsi sebagai pembentuk karakteristik (sifat-sifat) yang melekat pada sebuah tabel. Penerapan aturan-aturan normalisasi terhadap atribut-atribut pada sebuah tabel bisa berdampak pada penghilangan kolom tertentu, penambahan kolom baru, atau

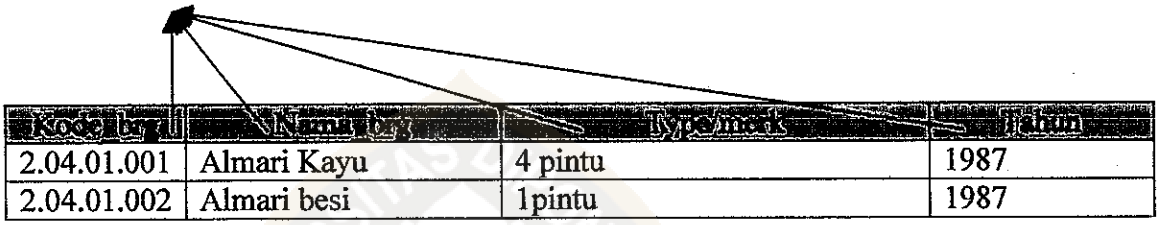
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

bahkan penambahan tabel baru. Penjelasan tentang hal ini akan dijelaskan pada subbab selanjutnya.

Contoh atribut dari Tabel Barang adalah *Kode_brg, Nama_brg, type/merk, tahun*

Tabel 1. Tabel Barang

Atribut Tabel



Kode_brg	Nama_brg	Type/merk	Tahun
2.04.01.001	Almari Kayu	4 pintu	1987
2.04.01.002	Almari besi	1pintu	1987

2.2.1.1 *Key*

Pada dasarnya, *key* adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam tabel secara unik. Artinya, jika suatu atribut dijadikan *key*, maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut.

Ada 3 (tiga) macam *key* yang dapat diterapkan pada sebuah tabel yaitu *superkey, candidate_key dan key primer.*

a. *Superkey*

Merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Bisa terjadi, ada lebih dari 1 kumpulan atribut yang mempunyai sifat ini pada sebuah tabel. Misal pada Tabel Barang yang dapat menjadi *superkey* adalah :

- *(Kode_brg, Nama_brg, type/merk, tahun)*
- *(Kode_brg, Nama_brg, type/merk)*
- *(Kode_brg, Nama_brg)*
- *(Kode_brg)*

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

b. *Candidate-key*

Merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Sebuah *candidate_key* suatu tabel tidak boleh berisi atribut atau kumpulan atribut yang telah manjadi *superkey* tabel yang lain. Jadi, sebuah *candidate_key* pasti merupakan *superkey* pada tabel tersebut, tapi belum tentu berlaku sebaliknya.

Fatansyah[1999] menulis definisi formal tentang *candidate_key* adalah :

Diberikan sebuah Tabel T, dengan atribut (T) =A1, A2, …, An. *Canditate-key* dari Tabel T adalah himpunan atribut K=Ai1,…,Aik dengan i = 1,2,...,n ; k≤n dan mempunyai 2 sifat :

1) Jika u,v mewakili 2 baris data yang berlainan dalam Tabel T, maka u[K]≠v[K]; artinya, akan ada paling sedikit satu atribut (Aim) yang merupakan anggota K dimana u[Aim] ≠v[Aim]
2) Tidak ada himpunan atribut yang lebih kecil yang menjadi bagian dari K yang memiliki sifat ke-a diatas.

c. *Key* Primer (*Primary-Key*)

Jika dalam satu tabel terdapat lebih dari satu *candidate_key*, maka salah satu dari *candidate_key* tersebut dapat dijadikan sebagai *key* primer. Pemilihan *key* primer dari *candidate_key* didasari oleh :

- *Key* tersebut lebih natural (lebih sering) untuk dijadikan sebagai acuan.
- *Key* tersebut lebih ringkas.
- Jaminan keunikan *key* tersebut lebih baik.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.2.1.2 Atribut Deskriptif

Atribut deskriptif adalah atribut-atribut yang tidak menjadi anggota dari *key* primer . Dalam Tabel Barang *Kode_brg* telah menjadi *key* primer, maka *Nama_brg, Type/merk, Tahun* merupakan atribut deskriptif.

2.2.1.3 Atribut Harus bernilai (*Mandatory Attribute*) dan Nilai Null

Dalam sebuah tabel, ada atribut yang harus berisi data (*mandatory attribute*) dan atribut yang nilainya boleh dikosongkan (*non mandatory attribute*). Atribut yang harus berisi data misalnya atribut *Kode_brg* dan *Nama_brg* pada Tabel Barang. Nilai (konstanta) null digunakan pada *non mandatory attribute* yang nilainya belum siap atau tidak ada.

Nilai null berbeda dengan spasi, spasi memberikan makna spasi sedangkan null berarti atribut tersebut belum/tidak memiliki nilai. Dari segi representasi fisik, spasi ekuivalen dengan karakter ke-32 dalam Tabel ASCII, sedangkan nilai Null ekivalen dengan karakter ke-0.

2.2.2 Domain dan tipe Data

Tipe data merujuk pada kemampuan penyimpanan data **yang mungkin** bagi suatu atribut secara fisik, tanpa melihat layak/tidaknya data tersebut bila dilihat dari kenyataan pemakainya. Sementara domain nilai lebih ditekankan pada batas-batas nilai **yang diperbolehkan** bagi suatu atribut.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Pada perancangan basis data yang dipertimbangkan adalah domain nilai dari setiap atribut, tipe data dari suatu atribut diperhitungkan pada saat implementasi basis data.

2.2.3 Ketergantungan Fungsional (*functional dependency*)

Diberikan sebuah Tabel T berisi paling sedikit 2 buah atribut A dan B, maka dapat dinyatakan notasi :

$$A \rightarrow B$$

Berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A, jika dan hanya jika untuk setiap kumpulan baris data (*row*) yang ada di Tabel T, pasti ada 2 baris data (*row*) di Tabel T dengan nilai untuk A yang sama, maka nilai untuk B pasti juga sama.

Secara lebih formal definisi ketergantungan fungsional menurut Korth[1986] adalah :

Diberikan 2 *row* t1 dan t2 dalam Tabel T dimana K *superkey* dan $K \rightarrow R$, jika $t1(K) = t2(K)$, maka $t1(R) = t2(R)$, berarti $t1 = t2$.

Misal pada Tabel Barang, terdapat *Kode_brg* $\rightarrow$ *Nama_brg* yang berarti bahwa atribut *Nama_brg* tergantung pada atribut *Kode_brg*, hal ini dibuktikan dengan fakta: untuk setiap *Kode_brg* yang sama, maka pasti nilai *Nama_brg* – nya juga sama.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.2.3.1 Determinan, Determinan Transitif dan Identitas

Misalkan suatu tabel mempunyai atribut A,B,C. A disebut determinan dari B jika A menjadi penentu B, sedangkan B diijinkan bernilai kosong atau mempunyai duplikasi.

Sedangkan apabila terjadi kondisi A merupakan determinan B, dan B determinan C, maka dikatakan A determinan transitif dari C.

Dalam penyusunan tabel basis data tidak diijinkan adanya dua baris yang mempunyai nilai atribut sama. Dalam hal ini dipilih satu atribut yang dapat digunakan sebagai pembeda atau identitas dari tabel. Nilai atribut yang menjadi identitas ini tidak pernah sama ataupun kosong.

2.2.4 Normalisasi dengan Ketergantungan Fungsional

Dalam perspektif normalisasi, sebuah basis data dapat dikatakan baik, jika setiap tabel yang menjadi unsur pembentuk basis data tersebut juga telah berada dalam keadaan baik atau normal. Cara mendapatkan tabel yang normal adalah dengan menerapkan langkah-langkah normalisasi pada tabel.

2.2.4.1 Bentuk Normal Pertama (1^{st} Normal Form)

Bentuk normal pertama dicapai bila tiap nilai atribut adalah tunggal, yaitu dengan melakukan penghilangan data ganda(*repeating groups*).

Contoh bentuk tidak normal yang dibawa ke bentuk normal pertama:

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Nama_ruang	Kode_brg
A201	2.01.04.001
	2.01.04.002
A301	2.01.04.001

Dibawa ke →

Nama_ruang	Kode_brg
A201	2.01.04.001
A201	2.01.04.002
A301	2.01.04.001

2.2.4.2 Bentuk Normal Kedua (2nd Normal Form)

Bentuk normal kedua dicapai bila atribut yang dijadikan identitas benar-benar menjadi determinan dari semua atribut. Bentuk normal kedua diperoleh dengan penguraian/*dekomposisi* tabel atau manipulasi data tabel pada kondisi bentuk normal pertama.

Misal dimiliki relasi :

(Nama_ruang, Fungsi, Kode_brg, Nama_brg)

Relasi ini berada dalam bentuk normal pertama karena terdapat dua buah atribut yang dapat dijadikan kunci relasi. Relasi ini akan dibawa kebentuk normal kedua dengan penguraian tabel. Penguraian ini membentuk 2 relasi :

Nama_ruang → Fungsi

Kode_brg → Nama_brg

2.2.4.3 Bentuk Normal Ketiga (3rd Normal Form)

Bentuk normal kedua dapat dibawa ke bentuk normal ketiga dengan menghilangkan ketergantungan fungsional antara atribut bukan utama.

Misal dimiliki relasi :

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

(Nama_ruang, Ukuran, Kapasitas)

relasi ini akan diuraikan menjadi :

Nama_ruang ⟶ Ukuran

Ukuran ⟶ Kapasitas

2.2.4.4 Bentuk Normal Boyce Codd (Boyce Codd Normal Form)

Tabel yang berada dalam bentuk normal ketiga juga berada dalam bentuk normal Boyce Codd apabila setiap atribut memiliki ketergantungan fungsional dengan atribut yang menjadi identitas.

Misal dalam relasi terdapat dua buah atribut yang dapat dijadikan identitas. Dalam hal ini diambil contoh atribut Kode_brg dan Nama_brg dalam Tabel Barang. Kedua atribut tersebut sama-sama unik, akan tetapi harus dipilih satu atribut untuk dijadikan identitas. Dalam hal ini dipilih Kode_brg sebagai kunci relasi, karena Kode_brg akan selalu berbeda untuk barang yang berbeda, namun Nama_barang masih mempunyai kemiripan nama.

2.2.4.5 Bentuk Normal Keempat (4^{th} Normal Form)

Bentuk normal keempat adalah tabel bentuk normal ketiga dengan nilai atribut tidak tergantung pada banyak nilai (dependensi nilai ganda/*multi valued dependencies).*

Dependensi nilai ganda terjadi apabila suatu atribut mempunyai kemungkinan untuk diisi lebih dari satu nilai. Misal :

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

(Nama_ruang, Ukuran, Kode_brg)

Atribut Kode_brg dapat berisi lebih dari satu, karena satu ruang bisa berisi lebih dari satu barang. Relasi ini diubah menjadi :

Nama_ruang ⟶ Ukuran

Nama_ruang ⟶ Kode_brg

2.2.4.6 Bentuk Normal Kelima (5th Normal Form)

Bentuk normal kelima berhubungan dengan ketergantungan pada gabungan beberapa atribut(*join dependency*). Tabel yang dalam proses normalisasinya mengalami pernguraian, apabila dikembalikan tidak akan mengubah nilai tabel awal.

2.3 Membuat Model Ketergantungan-Entitas (*Entity-Relationship Model*)

Pada model *Entity-Relationship(E-R)* , data dari dunia nyata ditransformasikan menjadi sebuah diagram *Entity-Relationship*(diagram E-R). 2 (dua) komponen utama pembentuk model E-R adalah Entitas (*Entity*) dan Relasi (*Relation*).

2.3.1 Entitas (*Entity*) dan Himpunan Entitas (*Entitas Sets*)

Korth [1986] menulis bahwa Entitas merupakan sebuah obyek yang nyata (*exists*) dan dapat dibedakan dari obyek yang lain.

Sekelompok Entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah Himpunan Entitas (*Entity Set*), Entitas menunjukkan individu

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

dan Himpunan Entitas menunjukkan pada rumpun (*family*). Tapi dalam kenyataan sehari-hari penyebutan Himpunan Entitas seringkali digantikan dengan Entitas saja.

Contoh Himpunan Entitas :

- Semua Kursi, atau Kursi saja,

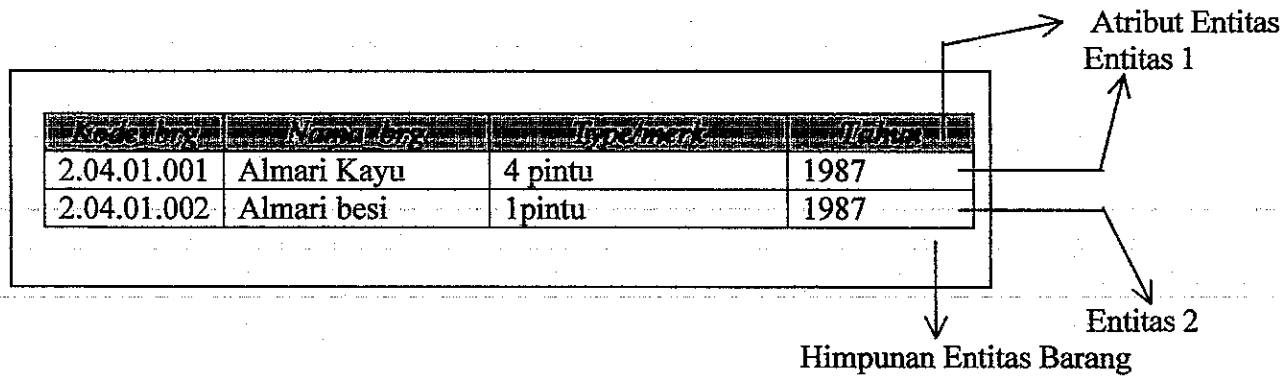
  dengan entitas kursi lipat,kursi kayu dengan meja, kursi kayu tanpa meja dan lain-lain.

2.3.2 Atribut (*Attribute/Properties*)

Entitas memiliki Atribut yang mendeskripsikan karakteristik (*properti*) dari entitas tersebut. Dalam model E-R, kedudukan atribut dalan entitas harus dapat membedakan atribut yang berfungsi sebagai *key* primer dan yang berfungsi sebagai atribut deskriptif. Contoh :

- Atribut *Kode_brg* merupakan *key* untuk Himpunan Entitas Barang, karena *Kode_brg* merupakan Entitas yang paling unik dalam Himpunan Entitas tersebut. Atribut-atribut yang lain (*Nama_brg, Type/merk, Tahun*) merupakan atribut deskriptif.

Tabel 2. Himpunan Entitas Barang



Kode_brg	Nama_brg	Type/merk	Tahun
2.04.01.001	Almari Kayu	4 pintu	1987
2.04.01.002	Almari besi	1pintu	1987

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

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

Ditulis oleh Korth [1986], Relasi adalah hubungan dari beberapa entitas. Dan 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, dan himpunan relasi R adalah himpunan bagian dari

$$\{(e_1,e_2, \ldots.., e_n) | e_1 \in E_1, e_2 \in E_2, \ldots., e_n \in E_n)$$

dimana $(e_1,e_2, \ldots.., e_n)$ adalah relasi.

Contoh :

Kode_brg	Nama_brg	.....	Nama_ruang	Fungsi	.....
2.04.01.001	Almari Kayu	……..	A101	R.Kuliah	……..
2.04.01.002	Almari Besi	……..	A201	R.Sidang	……..

Himpunan Relasi Relasi

Gambar 1. Relasi dan Himpunan Relasi Antara Himpunan Entitas Barang dan Himpunan Entitas Ruang

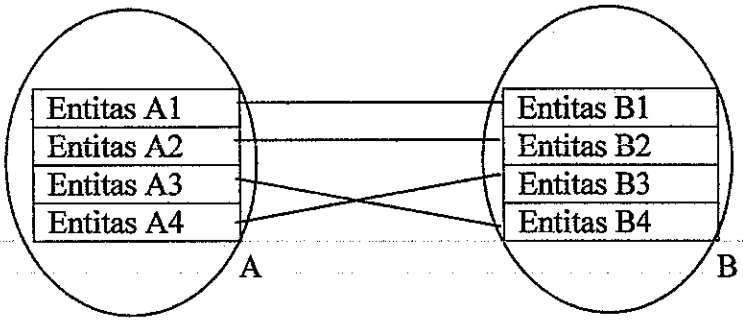
2.3.4 Kardinalitas / Derajat Relasi

Kardinalitas/Derajat Relasi menunjukkan jumlah entitas yang dapat dihubungkan dengan entitas lain menggunakan suatu relasi.

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*).

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

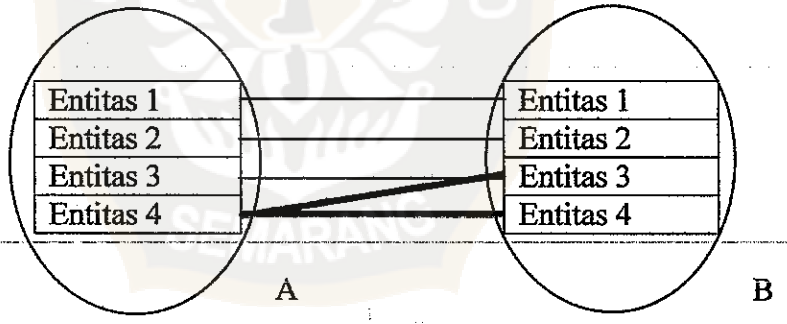
2.3.4.1 Satu ke Satu (*One to One*)



Gambar 2. Kardinalitas Satu ke Satu

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

2.3.4.2 Satu ke Banyak (*One to Many*)

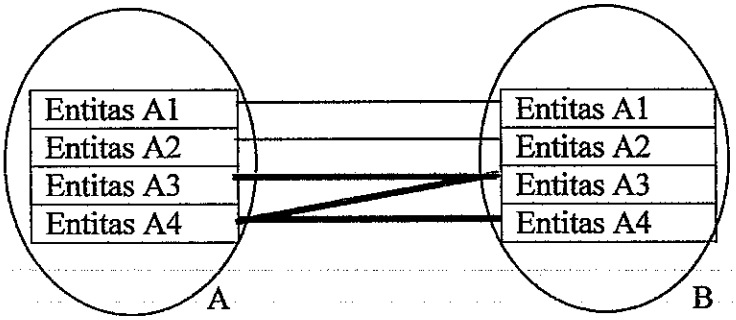


Gambar 3. Kardinalitas Satu ke Banyak

Setiap entitas pada himpunan entitas A dapat mempunyai relasi dengan banyak 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*).

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.3.4.3 Banyak ke Banyak (*Many to Many*)



Gambar 4. 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.

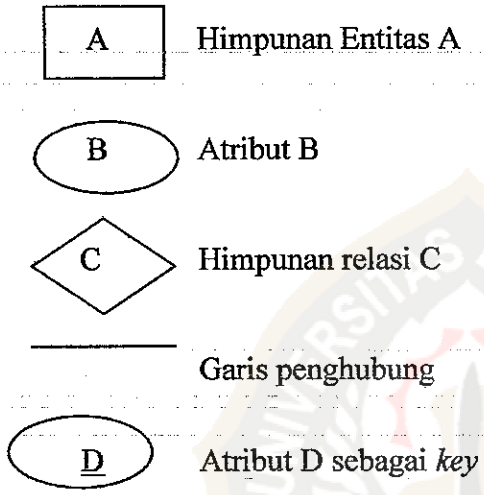
2.3.5 Diagram *Entity-Relationship* (Diagram E-R)

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 :

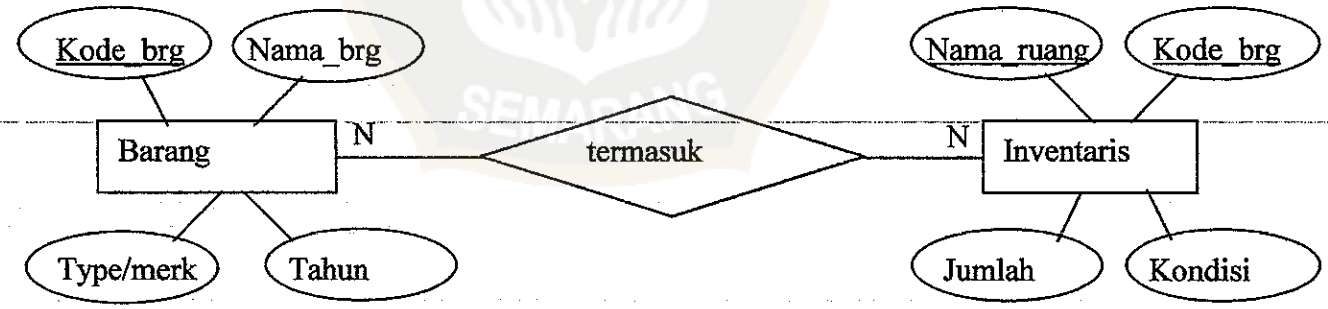
- Persegi panjang, menyatakan himpunan entitas.
- Elips / lingkaran, menyatakan atribut(atribut *key* ditunjukkan dengan garis bawah pada nama atribut).
- Belah ketupat, menyatakan himpunan relasi.
- Garis, sebagai garis penghubung antara himpunan entitas dengan atribut, dan himpunan relasi dengan himpunan entitas.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

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).

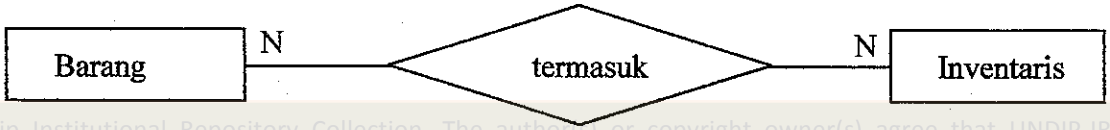


Contoh diagram E-R :



Gambar 5. Diagram E-R untuk Relasi Banyak ke Banyak

Jika penggambaran diatas dirasakan terlalu rumit, maka daftar atribut dapat disajikan dalam bentuk diagram eksternal. Diagram eksternal dari relai diatas adalah :



This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

dengan :

Barang = {Kode_brg, Nama_brg, Type/Merk, Tahun}

Inventaris = {Kode_brg, Nama_ruang, Jumlah, Kondisi}

2.3.6 Kamus Data (*Data Dictionary*)

Ditulis oleh Pohan[1997], Kamus data mempunyai fungsi membantu pelaku sistem untuk mengerti aplikasi secara detil, dan mereorganisasi semua elemen data yang digunakan dalam sistem sehingga pemakai dan penganalisa sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, pemyimpanan dan proses.

Notasi yang umum digunakan dalam menganalisa sebuah sistem dengan menggunakan sejumlah simbol yaitu :

No	Simbol	Uraian
1	=	Terdiri dari, mendefinisikan, diuraikan menjadi, artinya
2	+	Dan
3	()	Opsional (boleh ada atau boleh tidak)
4	{}	Pengulangan
5	[ ]	Memilih salah satu dari sejumlah alternatif, seleksi
6	**	Komentar
7	@	Identifikasi atribut kunci
8	|	Pemisah sejumlah alternatif pilihan antara simbol [ ]

2.4 *Structured Query Language* (SQL)

Structured Query Language (SQL) atau sering disebut sebagai *sequel* merupakan bahasa strandar untuk pengolahan database, dan berupa program manajemen basis data yang bekerja berdasarkan data yang disimpan secara

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

sistematis dan teratur didalam tabel: SQL mulai dikembangkan pada akhir 1970-an di laboratorium IBM, San Jose, California. Kurniawan [2001].

Sintaks penggunaan SQL yang paling umum digunakan diantaranya :

2.4.1 *Select*

Perintah Select digunakan untuk mengambil data dari suatu tabel. Sintaksnya adalah sebagai berikut :

SELECT {*|namafield} FROM namatabel

[INTO tabeltujuan][WHERE kondisi]

2.4.2 *Insert*

Perintah Insert digunakan untuk menyisipkan data kedalam tabel. Sintaksnya sebagai berikut :

INSERT INTO namatabel [(field1[,field2,….])}

VALUE (ekspresi1[,ekspresi2,…])

atau : INSERT INTO namatabel FROM ARRAY namaarray

2.4.3 *Delete*

Perintah delete digunakan untuk menghapus sebuah record dari tabel. Sintaksnya sebagai berikut :

DELETE FROM namatabel WHERE kondisi

2.4.4 *Update*

Perintah update digunakan untuk memperbaharui nilai suatu data. Sintaksnya sebagai berikut :

UPDATE namatabel

SET kriteria WHERE kondisi

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.4.5 Operator dan Fungsi SQL

SQL mendukung penggunaan operator-operator dan fungsi-fungsi, diantaranya adalah :

2.4.5.1 Operator Aritmetika

Operator	Keterangan
+	Tambah
-	Kurang
*	Kali
/	Bagi
%	Modulus

2.4.5.2 Operator Perbandingan

Operator	Keterangan
=	Sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan
<> atau !=	Tidak sama dengan

2.4.5.3 Operator Logika

Operator	Keterangan
AND	Dan
OR	Atau
NOT	tidak

2.4.5.4 Operator Karakter

Operator karakter yang didukung SQL adalah kata kunci LIKE yang diikuti oleh operator :

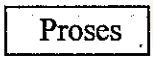
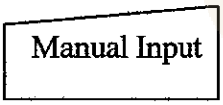
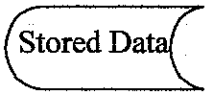
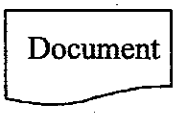
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Operator	Keterangan
%	Sembarang karakter, berapapun jumlahnya
-	Sembarang satu karakter
{}	Sembarang karakter yang terletak di dalam kurung siku

2.5 Diagram Blok (*Block Chart*)

Diagram blok berfungsi memodelkan masukan, keluaran, referensi, master, proses ataupun transaksi dalam simbol-simbol tertentu. Pada dasarnya tidak berorientasi pada fungsi, waktu, ataupun aliran data, tetapi lebih ke arah proses. Pohan [1997].

Simbol-simbol yang digunakan dalam diagram blok terdiri dari :

Simbol	Uraian
	Proses digambarkan dengan persegi panjang. Umumnya mendefinisikan mekanisme perekaman dan pelaporan.
	Perangkat masukan: digambarkan dengan kombinasi segitiga. Umumnya mendefinisikan fungsi pemasukan data atau *key in*. dapat berarti masukan untuk disimpan ataupun tidak
	Data tersimpan : digambarkan dengan kombinasi garis lengkung dan lurus, umumnya mendefinisikan file referensi, file master maupun file temporer yang digunakan dalam proses.
	Monitor : digambarkan dengan kombinasi garis lengkung, umumnya mendefinsikan keluaran dalam bentuk layar (*screen*).
	Dokumen : digambarkan dengan kombinasi persegi panjang dan garis lengkung. Umumnya mendefinisikan dokumen masukan(formulir) dan dokumen keluaran (laporan).

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

2.6 Tahapan Perancangan Basis Data

Seperti dikemukakan oleh Waljiyanto [2000] tahapan perancangan basis data meliputi :

a. Koleksi dan analisis persyaratan

   Proses pengumpulan dan analisis tujuan dan harapan pemakai basis data. Untuk itu pada tahap ini dilakukan penentuan dan penetapan persyaratan data dan persyaratan proses.

b. Perancangan konseptual basis data

   Pada tahap ini dihasilkan skema konseptual dari basis data yang bebas SMBD tertentu. Digunakan pemodelan bahasa tingkat tinggi seperti model *entity relationship(E-R)* atau model *enhanced entity relationship(EER)*. Pada tahap ini juga ditentukan transaksi data yang dapat dilakukan terhadap sistem basis data.

c. Pemilihan sistem manajemen basis data (SMBD)

   Pemilihan SMBD ditentukan oleh faktor teknik, ekonomi, dan politik dalam organisasi.

d. Perancangan logical basis data

   Tujuan dari tahap ini adalah menyusun rancangan konseptual dan skema eksternal yang sesuai dengan SMBD yang dipilih. Dalam hal ini dilakukan transformasi model konseptual dan skema eksternal yang dihasilkan pada tahap sebelumnya kedalam model data yang sesuai dengan SMBD yang digunakan.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

e. Perancangan fisikal basis data (pemetaan model data)

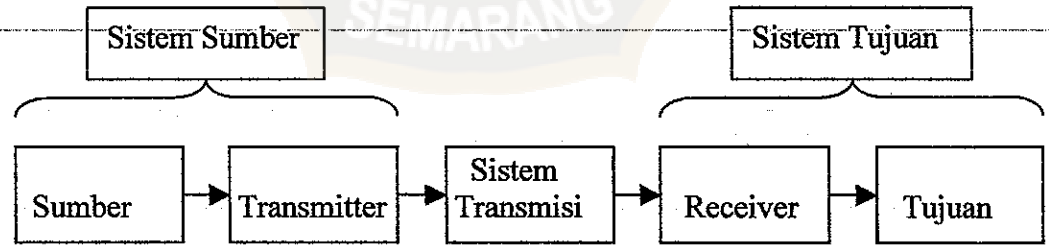
Tahap ini bertujuan untuk membuat spesifikasi struktur penyimpanan dan jalur akses data sehingga diperoleh kemampuan sistem yang baik untuk berbagai aplikasi, dengan mempertimbangkan waktu tanggap *(response time),* penggunaan memori komputer, dan transaksi data.

f. Implementasi sistem basis data

Tahap ini merupakan implementasi dari hasil pemodelan logical dan fisikal dalam operasional sistem basis data.

2.7 Jaringan, *World Wide Web (WWW)* dan *Web Database*

Menurut Stalligs[2001], pada jaringan komputer, baik pada jaringan Intranet maupun jaringan Internet, komunikasi data yang terjadi secara umum dapat digambarkan sebagai berikut :



Gambar 6. Model Komunikasi Data Secara Umum

Sistem sumber akan membangkitkan data yang dapat ditransmisikan ke sistem tujuan melalui sistem transmisi. Perbedaan antara jaringan Intranet dan jaringan Internet terletak pada sistem transmisinya. Jaringan Intranet memanfaatkan HUB sebagai sistem transmisi, sedangkan jaringan Internet memanfaatkan jaringan telepon sebagai sistem transmisi. Dari perbedaan ini,

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

maka pada dasarnya program yang dapat dijalankan pada jaringan Intranet dapat dijalankan pada jaringan Internet.

World Wide Web (WWW) bukanlah internet, demikian pula sebaliknya. Namun demikian, WWW dan internet sangat berkaitan. Internet adalah suatu jaringan komputer global, sedangkan WWW bukan hanya terdiri atas jaringan tetapi didalamnya terdapat suatu set aplikasi komunikasi dan sistem perangkat lunak yang memiliki karakteristik sebagai berikut :

- Umumnya terdapat pada Internet host dan client
- Umumnya mengunakan protokol TCP/IP
- Mengerti HTML
- Mengikuti model *client/server* untuk komunikasi data dua arah
- Memungkinkan client untuk mengakses *server* dengan berbagai protokol seperti HTTP, FTP, Telnet, dan Gopher
- Memungkinkan client untuk mengakses informasi dalam berbagai media, seperti teks, audio dan video.
- Menggunakan model alamat Uniform Resource Locators (URL).

Peraturan-peraturan yang terdapat pada WWW sekarang ini berkembang dari ide dan konsep yang ditelurkan oleh Tim Berners-Lee, seorang peneliti pada CERN Particle Lab di Jenewa, Swiss. Pada tahun 1989, Berners-Lee merumuskan suatu proposal tentang sebuah sistem *hypertext* yang memiliki tiga komponen sebagai berikut :

- Antarmuka yang konsisten untuk semua platform. Antarmuka ini harus menyediakan akses yang dapat digunakan oleh berbagai jenis komputer.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

- Akses informasi yang universal. Setiap pengguna harus dapat mengakses setiap informasi yang tersedia.
- Antarmuka yang menyediakan akses terhadap berbagai jenis dokumen dan protokol.

Seperti sistem basis data yang lain, Web database juga merupakan sistem penyimpanan data yang dapat diakses oleh bahasa pemrograman tertentu. Namun tidak seperti sistem database konvensional yang hanya ditujukan untuk platform tertentu, Web database dapat diakses oleh aplikasi Web yang lebih bersifat umum. Web database dapat diakses oleh aplikasi-apikasi Web yang dikembangkan dengan HTML tag, kontrol ActiveX, dan pemrograman yang bersifat *server-side* melalui CGI, Microsoft IIS *(Internet Information Server)*, atau skrip yang bersifat *server-side* seperti *Microsoft Active Server Pages* (ASP).

Kemampuan untuk mengintegrasikan database ke dalam aplikasi yang dapat diakses pengguna menggunakan *Web browser* inilah yang menjadikan suatu database biasa disebut Web database.

2.8 Microsoft Active Server Pages (ASP) dan Microsoft Acces 2000

2.8.1 Microsoft Active Server Pages (ASP)

Microsoft Active Server Pages (ASP) merupakan suatu skrip yang bersifat server-side yang ditambahkan pada HTML untuk membuat sebuah Web menjadi lebih menarik, dinamis, dan interaktif. Dengan ASP anda dapat mengolah data yang diambil dengan sebuah form, membuat aplikasi-aplikasi tertentu dalam sebuah Web, ataupun membuat database dalam sebuah Web.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

ASP bersifat *server-side*, ini berarti proses pengerjaan skrip berlangsung di server, bukan di browser / client. Dengan kata lain jika anda menggunakan sebuah browser untuk memanggil sebuah file ASP, maka browser tersebut mengirimkan permintaan ke Web server, kemudian server tersebut mengeksekusi setiap skrip yang ada dan hasilnya dikirimkan kembali ke browser. Karena bersifat server-side, maka untuk dapat dijalankan pada sebuah PC biasa yang berbasis windows, PC tersebut perlu disimulasikan menjadi sebuah *Web server* dengan menginstal Microsoft Personal Web Server (PWS) atau Microsoft Internet Information Services (IIS).

Bahasa skrip standar yang digunakan oleh ASP adalah Microsoft VBScript dan Microsoft JScript.

ASP dapat dijalankan dengan baik minimal pada windows 95 OSR 2, selain itu dibutuhkan sebuah browser dan sebuah teks editor atau HTML editor. Untuk membuat sebuah aplikasi database, diperlukan juga perangkat lunak pengolah database seperti Microsoft Acces, Microsoft Visual FoxPro, dBase, dan lain-lain.

2.8.2 Microsoft Acces 2000

Microsoft Access adalah sistem manajemen basis data canggih yang dapat digunakan secara efisien untuk menampilkan dan mengelola data . Microsoft Access 2000 dapat dioperasikan untuk mengelola data pada komputer tunggal maupun pada jaringan.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)