

BAB II

LANDASAN TEORI

2.1. Pengertian Sistem.

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedurnya mendefinisikan sistem sebagai berikut :

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.¹

Pendekatan sistem yang menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut :

Sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

2.1.1 Pengertian Informasi

Informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimannya yang menggambarkan suatu kejadian (*event*) yang nyata (*fact*).

2.1.2. Definisi Sistem Informasi

Sistem Informasi adalah komponen-komponen yang terkait satu sama lainnya yang bekerja sama untuk mengumpulkan, memproses, menyimpan, dan

¹ Jerry Fitzgerald, Ardra F Fitzgerald dan Warren D Stailling Jr, Fundamental of system analysis, Newyork, John wiley and sons (1981).

menyebarkan informasi guna mendukung pengambilan keputusan, koordinasi, pengendalian, analisis, dan memvisualisasikan di dalam organisasi².

2.1.3 Pemesanan Cek Laboratorium

pemesanan cek laboratorium ini berisi fungsi-fungsi dari bagian laboratorium rumah sakit yang nantinya akan disebutkan dalam penyusunan skripsi ini, yaitu :

1. Menerima pemesanan cek sampel.
2. Membuat bukti pembayaran.
3. Melakukan pemrosesan sampel.
4. Membuat laporan hasil pemeriksaan suatu sampel.
5. mengirimkan laporan hasil pemeriksaan sampel ke dokter yang bersangkutan.
6. Menyimpan data-data hasil pemeriksaan pasien.

2.1.3.1 Menerima Pemesanan Cek Sampel.

Ketika seorang dokter membutuhkan data lebih rinci mengenai penyakit seorang pasien yang membutuhkan pemeriksaan laboratorium, maka ia harus melakukan pemesanan periksa sampel ke bagian laboratorium rumah sakit. Pemesanan dilakukan dengan cara mengisi lembar formulir pemesanan yang telah disediakan. Formulir tersebut kemudian di berikan kepada pasien untuk diserahkan ke bagian registrasi untuk melakukan pembayaran atas pemeriksaan yang akan dilakukan pasien tersebut.

² H.M. Jogiyanto, *Analisis dan Design Sistem Informasi*, Pendekatan Terstruktur (Yogyakarta, Andi Offset, 1992) hal 1.

2.1.3.2. Membuat Bukti Pembayaran

Setelah pasien menyerahkan formulir pemesanan pemeriksaan laboratorium, maka petugas registrasi akan melihat pemeriksaan apa saja yang akan dilakukan oleh pasien tersebut, setelah petugas registrasi mengetahui jenis-jenis pemeriksaan yang akan dilakukan oleh pasien tersebut maka petugas registrasi akan mengetahui jumlah biaya yang harus dibayar oleh pasien tersebut, dan pasien tersebut harus melunasi pembayaran tersebut untuk dapat melakukan pemeriksaan laboratorium, pembayaran yang dilakukan oleh pasien dapat dilakukan dengan membayar secara tunai atau pasien tersebut dapat menggunakan Asuransi Kesehatan (ASKES) atau jaminan dari perusahaan atau kantor tempat pasien tersebut berkerja. Selanjutnya apabila pasien tersebut telah melunasi biaya registrasi, petugas registtrasi akan menyerahkan formulir pemesanan pemeriksaan laboratorium, kepada petugas laboratorium untuk diproses selanjutnya.

2.1.3.3. Melakukan Pemrosesan Sampel.

Setelah formulir tersebut diberikaan kepada bagian pengambilan sampel maka petugas pengambilan sampel akan mengambil sampel seperti misalnya darah, urin, tinja dan lain-lain dari pasien yang dibutuhkan guna mengetahui normal atau tidaknya pasien tersebut.

2.1.3.4. Membuat Laporan Hasil Pemeriksaan Suatu Sampel.

Hasil pemrosesan sampel akan diperiksa oleh kepala laboratorium sektor untuk pemberian persetujuan dan komentar.

2.1.3.5. Mengirimkan Laporan Hasil Pemeriksaan Sampel Ke Dokter Yang Bersangkutan.

hasil pemeriksaan laboratorium yang sudah dikomentari (diberi catatan) oleh kepala laboratorium diserahkan kepada dokter. Dokter kemudian mendiagnosis hasil tersebut.

2.1.3.6. Menyimpan Data-data Hasil Pemeriksaan Pasien.

Data-data pasien hasil pemeriksaan laboratorium pada akhirnya disimpan oleh dokter bersangkutan dan oleh pihak laboratorium itu sendiri

2.2. *Prototipe*

Prototipe memiliki arti sebagai berikut :

- a. Suatu benda atau sesuatu yang dibuat berdasarkan model aslinya.
- b. Sesuatu yang menunjukkan karakteristik pokok dari model selanjutnya.
- c. Contoh yang standar atau serupa.
- d. Bentuk yang sebenarnya pada awalnya dan biasanya merupakan bentuk fungsional dari suatu tipe yang baru atau rancangan dari sebuah konstruksi.

2.2.1. Pengertian *Prototipe*

Menurut Pressman pembuatan *prototipe* merupakan suatu proses yang memungkinkan pihak pengembang untuk membuat sebuah model dari piranti lunak yang akan dikembangkan.³ Model ini dapat berupa :

³ Roger. S. Pressman, Software Engineering : A Practitioner's Approach (Singapore : McGraw-Hill, 1992), hal 27.

- a. Suatu *prototipe* kertas yang menggambarkan hubungan manusia-mesin dalam sebuah bentuk yang memungkinkan *user* untuk mengerti bagaimana hubungan tersebut terjadi.
- b. Suatu *prototipe* pekerjaan yang mengimplementasikan berapa bagian dari kumpulan fungsi yang dibutuhkan untuk perangkat lunak yang diinginkan.
- c. Suatu program yang telah ada yang menampilkan sebagian atau keseluruhan dari fungsi yang diinginkan, tetapi memiliki karakteristik lain yang dikembangkan berdasarkan pengembangan yang baru.

2.2.2. Teknik Pembuatan *Prototipe*

Untuk mempercepat pembuatan *prototipe*, terdapat tiga metode dan perangkat bantu yaitu *4GT (Fourth Generation Techniques)*, *Reusable Software Component*, *Formal Specification and Prototyping Environment*.

a. *4GT (Fourth Generation Techniques)*⁴

Teknik pengembangan yang meliputi suatu array dari *query* basis data yang luas dan bahasa penghasil laporan, program, penghasil aplikasi serta bahasa non prosedural. Dengan *4GT* memungkinkan pembuatan piranti lunak dengan cepat dan cocok untuk pembuatan *prototipe*.

Teknik ini merupakan teknik pembuatan *prototipe* yang digunakan dalam membangun sistem ini, karena memiliki kelebihan sebagai berikut :

- Waktu yang dibutuhkan untuk pembuatan *prototipe* relatif cepat.

⁴ *Ibid*, hal 55.

- Lingkungan 4GT dapat menjangkau sebagian besar kategori aplikasi perangkat lunak.
- Cocok untuk pengembangan aplikasi yang tidak terlalu besar.

Selain kelebihan diatas teknik 4GT juga memiliki kekurangan antara lain : untuk pengembangan perangkat lunak yang besar membutuhkan analisis, desain dan pengujian yang sangat banyak.

b. *Reusable Software Component*⁵

Pendekatan ini lebih pada perakitan *prototipe* daripada membangun, dengan menggunakan kembali komponen piranti lunak yang telah ada. Komponen tersebut berupa struktur data atau komponen arsitektur piranti lunak (program dan modul). Dengan cara ini memungkinkan untuk menggunakan kembali komponen tanpa harus mengerti secara rinci proses yang terjadi, sehingga pengembangan bisa cepat.

c. *Formal Specification and Prototyping Environment*⁶

Dengan metode ini pihak pengembang mengembangkan lingkungan interaktif dengan menggunakan bahasa formal, yang :

- 1) Memungkinkan analis menciptakan spesifikasi berbasis bahasa secara interaktif dari sebuah sistem atau piranti lunak.
- 2) Meminta perangkat terotomatisasi yang menterjemahkan spesifikasi berbasis bahasa ke dalam kode yang dapat dieksekusi.
- 3) Memungkinkan untuk menggunakan kode *prototipe* yang dapat dieksekusi untuk memperjelas kebutuhan formal.

⁵ *Ibid*, hal 57.

⁶ *Ibid*, hal 39.

2.3. Alat bantu perancangan Aplikasi

Dalam merancang suatu aplikasi, seseorang akan membutuhkan peralatan (*tools*) yang membantu dalam perancangan suatu aplikasi. Peralatan yang dibutuhkan dalam perancangan akan sangat tergantung pada aplikasi apa yang ingin dihasilkan. Ada beberapa peralatan seperti Flowchart, *State Transition Diagram* (STD), *structured chart*, algoritma, dan peralatan perancangan yang lain.⁷

2.3.1. Flowchart (Bagan Alir)

Menurut Mulyadi bagan alir dokumen atau *flowchart* adalah simbol-simbol standar yang digunakan oleh bagian analisis sistem untuk membuat bagan alir dokumen yang menggambarkan sistem tertentu. Sebenarnya banyak cara untuk mengembangkan bagan alir dokumen suatu sistem, namun *flowchart* merupakan alat yang digunakan secara luas oleh para analis untuk menggambarkan aliran dokumen suatu sistem.⁸

Ada 5 Macam Bagan Alir Yaitu :

1. Bagan alir Program (Program Flowchart)
2. Bagan Alir Sistem (System Flowchart)
3. Bagan Alir Dokumen (Document Flowchart)
4. Bagan Alir Skematik (Schematic Flowchart)
5. Bagan Alir Proses (Process Flowchart)

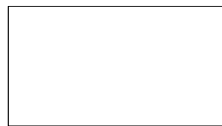
⁷ Budi Sutedjo Dharma Oetomo, Perencanaan dan Pembangunan Sistem Informasi, (Yogyakarta : Andi Offset, 2002), hal 131.

⁸ Mulyadi. (1993). Sistem Akuntansi. Edisi Ke-3. STIE YKPN, Yogyakarta, hal 60

2.3.1.1. *Flowchart Program*

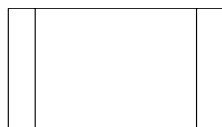
Flowchart program adalah Bagan alir yang menunjukan didalam Program secara logika. Bagan Alir Program digunakan untuk Alat bantu Komunikasi dan untuk Dokumentasi

2.3.1.2. *Simbol Flowchart*



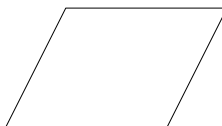
Processing :

Satu atau beberapa himpunan penugasan yang akan dilaksanakan secara berurutan.



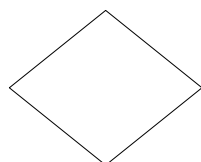
Predfined Process :

Digunakan untuk proses yang detilnya dijelaskan terpisah, misalnya dalam bentuk subroutine.



Input / Output :

Data yang akan dibaca dan dimasukkan kedalam memori komputer dari suatu alat input atau data dan harus melewati memori untuk dikeluarkan dari alat output



Descision (Keputusan) :

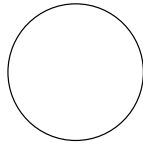
Tahap pengambilan keputusan, digunakan bila terdapat pilihan dalam pengambilan keputusan dan menunjukkan percabangan kepada jalur alternatif.



Terminator :

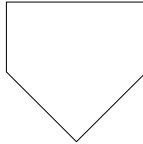
Menggambarkan simbol awal dan akhir dari suatu

aliran.



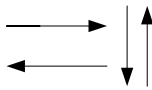
On-page Connector :

Menghubungkan aliran proses pada halaman yang sama.



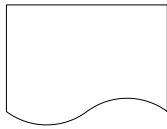
Off-page Connector :

Menghubungkan aliran proses pada halaman yang berbeda.



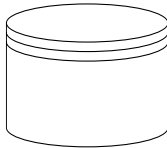
Flow line :

Menunjukkan bagan instruksi selanjutnya



Document :

Menggambarkan semua jenis dokumen atau laporan yang dibuat secara manual atau di cetak oleh komputer.



Hard disk :

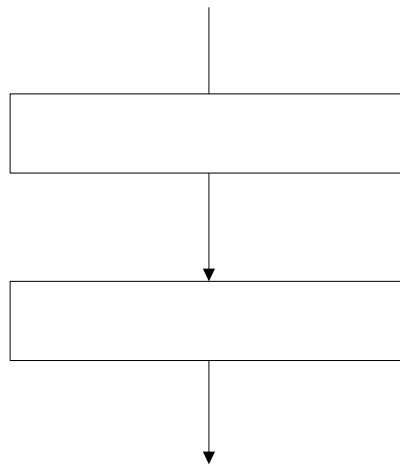
Input / output menggunakan hard disk.

2.3.1.3. Bentuk Struktur Logika Flowchart

Berikut ini adalah beberapa bentuk dasar struktur logika yang diwakili oleh bagan alir :

1) Struktur Urut Sederhana (*Simple Sequence*)

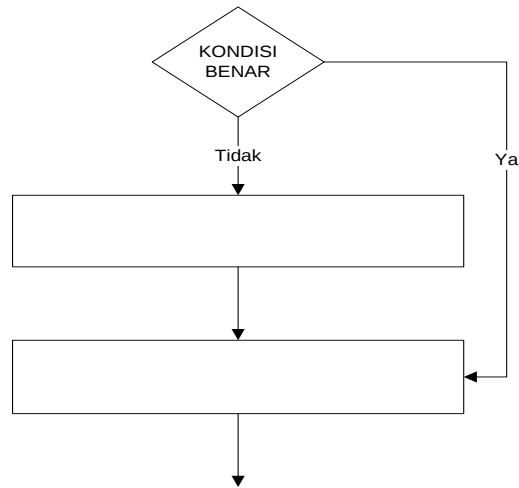
Gambar 2.2 menunjukkan struktur logika *simple sequence*, struktur ini hanya berisi langkah-langkah yang urut saja, satu diikuti yang lainnya.



Gambar 2.1 Struktur Logika Simple Sequence

2) Struktur Loncat (*Branch*)

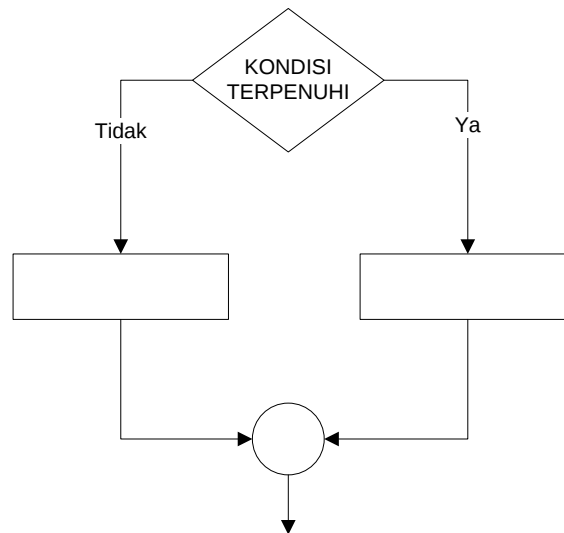
Gambar 2.3 menunjukkan struktur logika *branch*. Struktur ini berisi suatu loncatan ke proses tertentu oleh statemen *GOTO* atau statmen *IF*.



Gambar 2.2 Struktur Logika Branch

3) Selection

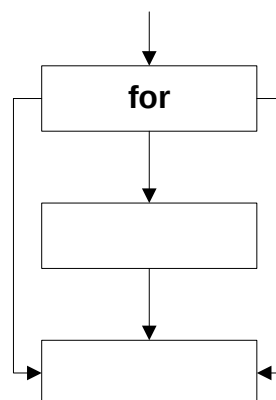
Gambar 2.4 menunjukkan struktur logika selection. Struktur ini merupakan penyelesaian kondisi yang menggunakan statmen *IF-THEN-ELSE*



Gambar 2.3 Struktur Logika Selection

4) Struktur Perulangan *For* (*For Loop Structure*)

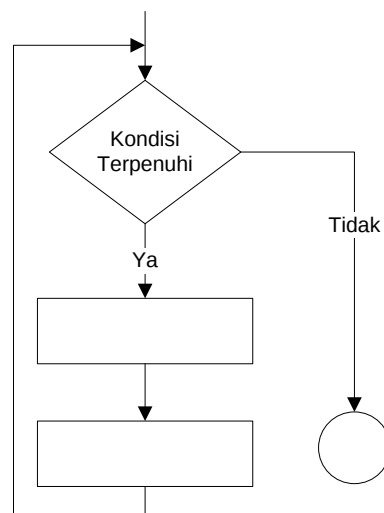
Gambar 2.5 menunjukan struktur logika *FOR Loop*. Struktur ini merupakan perulangan beberapa blok statemen yang dibentuk dengan statemen *FOR*.



Gambar 2.4 Struktur Logika For Loop

5) Struktur Perulangan *DO WHILE* (*Do While Loop*)

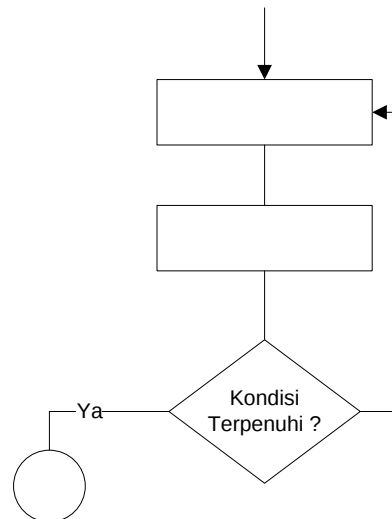
Gambar 2.6 menunjukkan struktur logika *DO WHILE Loop*. Struktur ini menunjukkan suatu blok statemen akan dikerjakan (*DO*) berulang-ulang selama (*WHILE*) kondisi yang diseleksi masih terpenuhi dan akan keluar dari lingkungan *loop* bila kondisi sudah tidak terpenuhi.



Gambar 2.5 Struktur Logika Do While Loop

6) Struktur Perulangan *DO UNTIL (Do Until Loop)*

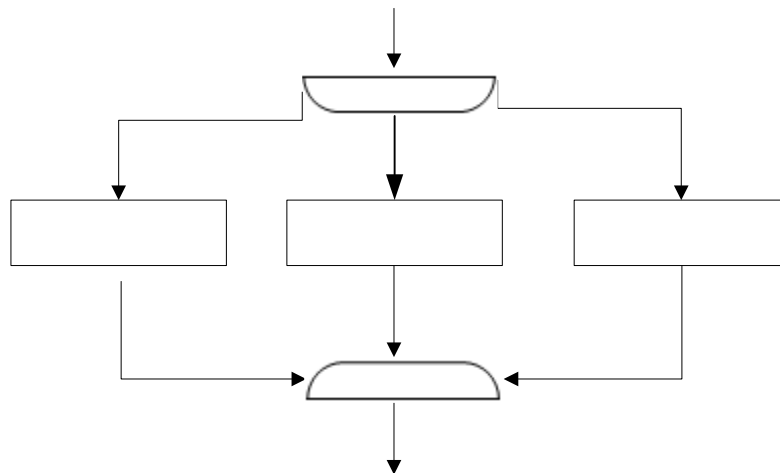
Gambar 2.7 menunjukkan struktur logika *DO UNTIL LOOP*. Struktur ini menunjukkan suatu blok statemen akan di kerjakan (*DO*) sampai (*UNTIL*) kondisi yang diseleksi tidek terpenuhi.



Gambar 2.6 Struktur Logika Do Until Loop

7) Struktur *Case*

Gambar 2.8 menunjukkan struktur logika *CASE*. Struktur ini akan memperoses sebuah blok statemen pada salah satu kondisi case yang terpenuhi dari sejumlah case yang ada.



Gambar 2.7 Struktur Logika Case

2.3.2. Diagram Alir Data (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang saling berkaitan⁹

Data Flow Diagram (DFD) adalah diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem secara terstruktur dan jelas tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut disimpan¹⁰.

Dengan hanya 4 simbol, DFD dapat mewakili sistem informasi fisik maupun logik. Keempat simbol yang digunakan dalam DFD mewakili data *flow*, data *store*, proses dan *sources/sinks* (eksternal entitas). DFD dapat digunakan

⁹ Raymond Mc Leod, Jr, *Sistem Informasi Management Versi Bahasa Indonesia Jilid 2 Edisi 7*, (Jakarta Prenhallindo, 2001) hal.590.

¹⁰ H. M. Jogiyanto, *Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur* (Yogyakarta, Andi Offset, 1992) hal.700-709.

untuk merepresentasikan suatu sistem yang otomatis maupun manual dengan melalui gambar yang berbentuk jaringan grafik.

Sebuah data flow diagram terdiri dari :

1. Eksternal Entitas

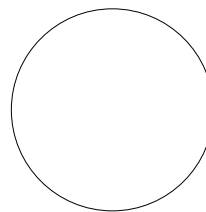
Entitas yang berada diluar sistem yang memberikan data kepada sistem atau yang menerima informasi dari sinks.



Gambar 2.8 Simbol Eksternal Entitas

2. Proses

Menggambarkan apa yang dilakukan oleh sistem. Berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan.



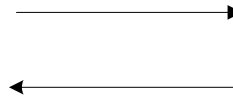
Gambar 2.9 Simbol Proses

Pedoman pemberian nama proses :

- a) Nama proses terdiri dari kata kerja dan kata benda yang mencerminkan fungsi proses tersebut.
- b) Tidak boleh ada beberapa proses yang memiliki nama yang sama.
- c) Proses harus diberi nomor.

3. Data Flow

Menggambarkan aliran data dari suatu entity ke entity lainnya, dan arah panah menggambarkan arah aliran data.

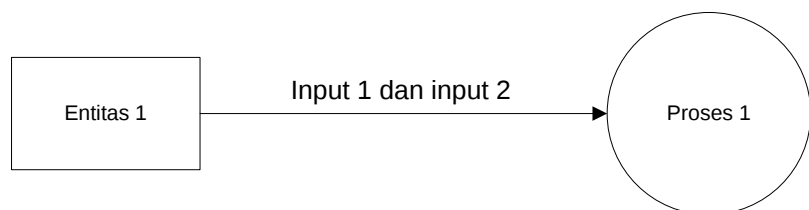


Gambar 2.10 Simbol Arah Data Flow

Macam-macam arus data :

a) Konsep paket dari data

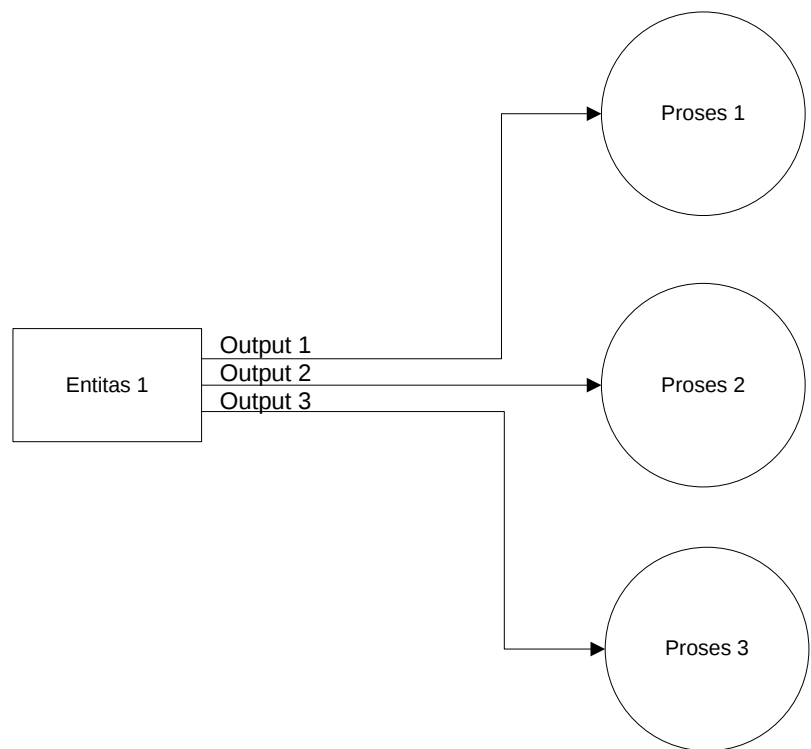
Bila ada data lebih dari satu harus disatukan menjadi arus data tunggal.



Gambar 2.11 Konsep Dari Paket Data

b) Konsep arus data menyebar

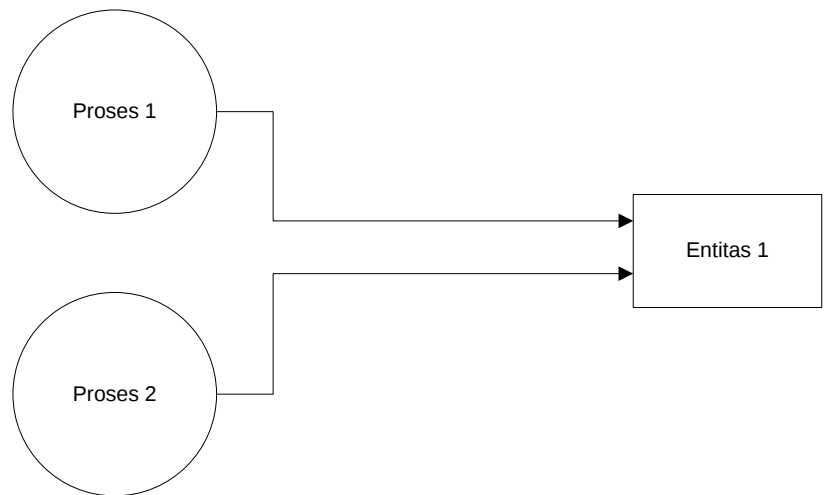
Arus data yang menyebar menunjukkan sejumlah tembusan dari arus data yang sama dari sumber yang sama ke tujuan berbeda.



Gambar 2.12 Konsep Arus Data Menyebar

c) Konsep arus data mengumpul.

Arus data yang mengumpul menunjukkan beberapa arus data yang berbeda dari sumber yang berbeda bergabung bersama-sama menuju ke tujuan yang sama.



Gambar 2.13 Konsep Arus Data Mengumpul

Pedoman pemberian nama aliran data :

- a) Nama aliran data yang terdiri dari beberapa kata dihubungkan dengan garis sambung.
- b) Tidak boleh ada aliran data yang namanya sama dan pemberian nama harus mencerminkan isinya.
- c) Sedapat mungkin nama aliran data ditulis lengkap.

4. Data store

Merupakan tempat simpanan data yang dapat berupa file, database di sistem komputer, arsip.



Gambar 2.14 Simbol Data Store

Pedoman pemberian nama data store :

- a) Nama harus mencerminkan isi dari data store tersebut
- b) Bila namanya lebih dari satu kata harus diberi tanda sambung.

Pedoman penggambaran data *store* :

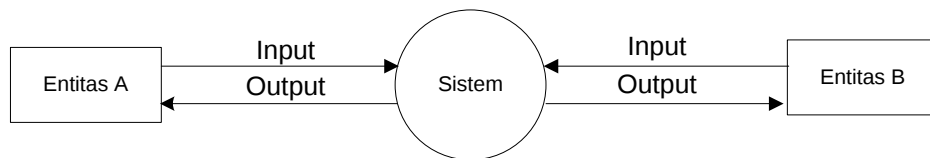
- a) Hanya proses saja yang berhubungan dengan simpanan data, karena yang menggunakan atau merubah data disimpanan data adalah suatu proses.
- b) Arus data yang menuju ke simpanan data dari proses sifatnya update terhadap data yang disimpan.
- c) Arus data yang berasal dari simpanan data ke suatu proses menunjukan bahwa proses tersebut menggunakan data disimpanan data.

Tingkatan diagram pada DFD :

- 1. *Context Diagram* (diagram hubungan, level 0)
- 2. Diagram Zero (diagram 0, level 1)
- 3. Diagram Rinci (level 2, level 3, dst)

Section 1.01 Context Diagram

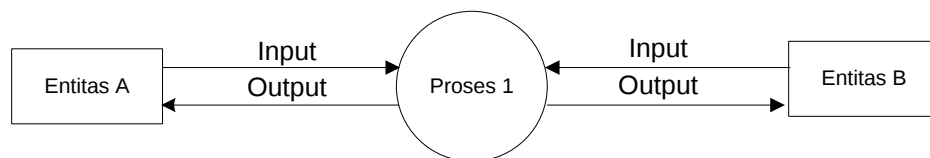
- a. Merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke atau output dari sistem.
- b. Memberikan gambaran tentang keseluruhan sistem.
- c. Hanya ada satu proses.
- d. Tidak boleh ada data store.



Gambar 2.15 Diagram Konteks

Section 1.02 *Diagram Zero*

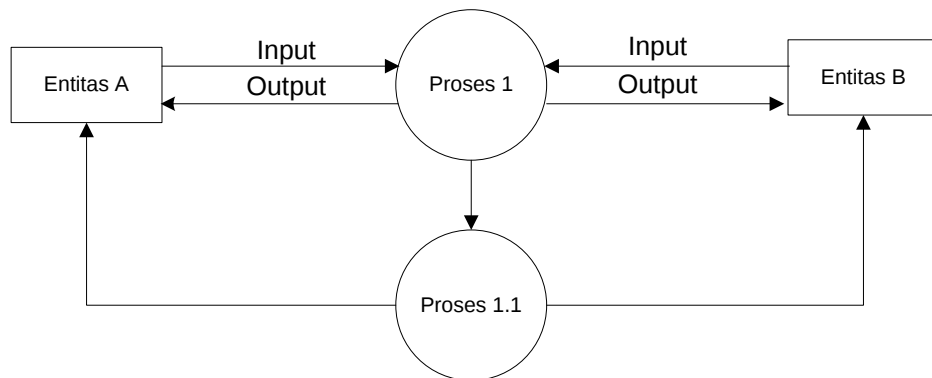
- a. Merupakan level di bawah context diagram yang menggambarkan proses secara rinci



Gambar 2.16 Diagram Zero (0)

Section 1.03 *Diagram rinci*

- a. Merupakan rincian dari diagram zero atau diagram level di atasnya.
- b. Didalam satu level setidaknya tidak terdapat lebih dari 7 buah proses dan maksimal 9, bila lebih maka harus dilakukan dekomposisi dimana lebih dari 9 dikumpulkan dulu baru kemudian dipecah pada level dibawahnya.

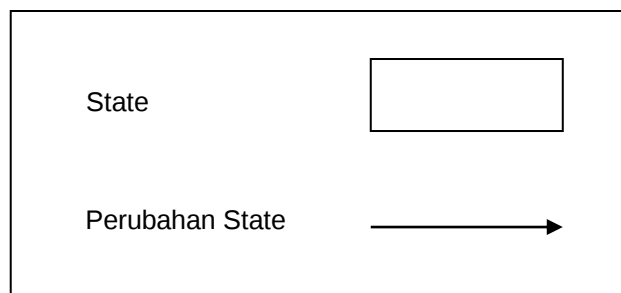


Gambar 2.17 Diagram Rinci

2.3.3. State Transition Diagram (STD)

State Transition Diagram (STD) merupakan suatu *modeling tool* yang menggambarkan sifat ketergantungan pada waktu dari suatu sistem.

Untuk menyusun suatu STD dibutuhkan beberapa notasi yang dapat menggambarannya. Sebuah kotak persegi panjang merepresentasikan sebuah *state*, sedangkan setiap transisi atau perubahan *state* dipresentasikan dengan sebuah garis berarah. Untuk lebih jelasnya perhatikan Gambar 2.3 berikut ini :



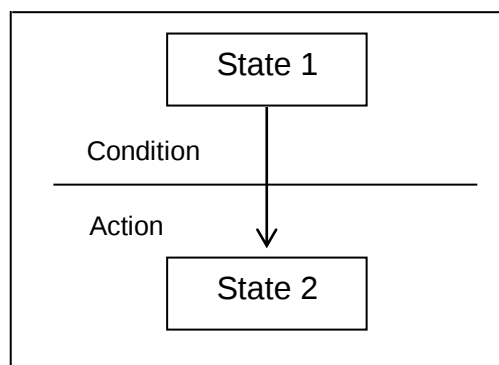
Gambar 2.18 Notasi STD

State adalah suatu bentuk yang mewakili keberadaan suatu proses atau komponen sistem yang ditentukan oleh suatu keadaan pada saat tertentu.

Sedangkan perubahan *state* adalah perubahan sistem dari satu *state* ke *state* lainnya.¹¹

Sebuah STD pun harus dilengkapi dengan *condition* dan *action*. *Condition* adalah suatu *event* pada *external environment* yang dapat dideteksi oleh sistem, dapat berupa sebuah sinyal, interupsi, atau data yang menyebabkan terjadinya perubahan dari suatu *state* lainnya. Sedangkan *action* adalah yang dilakukan sistem bila terjadi perubahan *state* atau merupakan reaksi terhadap *condition*.

Condition di tulis di dekat garis berarah yang menggambarkan arah perubahannya. Sedangkan *action* yang timbul di tulis dibawah *condition* tadi, dibatasi dengan sebuah garis. Untuk lebih jelasnya perhatikan gambar 2.7 berikut ini :



Gambar 2.19 Penggambaran STD

Setiap *state* selalu di pacu oleh sebuah *condition*. *State* yang tidak dipacu oleh *condition* apapun disebut sebagai *current state* dan *state* yang dipacu oleh sebuah *condition* namun tidak berubah menjadi *state* lainnya disebut sebagai *final*

¹¹ Hoffer Et Al (1996). hal 364

state. Dalam sebuah STD, *final state* tidak harus satu tapi dapat beberapa *state*, namun untuk *initial state* tidak boleh lebih dari satu.

Adapun pendekatan yang dapat diikuti untuk membuat suatu STD adalah sebagai berikut :

1. Identifikasikan setiap kemungkinan *state* dari sistem dan gambarkan masing *state* pada sebuah kotak. Lalu buatlah hubungan antara *state* tersebut.
2. Mulai dengan *state* pertama dan kemudian lanjutkan dengan *state - state* berikutnya sesuai dengan *flow* yang diinginkan.
3. Periksa apakah semua *state* dapat dicapai atau diakses.
4. Setiap *state* harus memiliki *successor* atau *output* kecuali jika merupakan *final state*.

Perubahan *state* harus dapat terjadi untuk segala macam kondisi, artinya sistem harus dapat pula mendeteksi bila terdapat *condition* maupun *action* yang tidak diharapkan.

2.3.4. Entity Relationship Diagram (ERD)

ERD adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan (dalam DFD). Dengan DFD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada tiga macam simbol yang digunakan yaitu¹² :

1. Entity

¹² Fatansyah, Sistem Basis Data

Entity adalah suatu obyek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang dibuat.



Gambar 2.20 Simbol Himpunan Entitas

2. Atribut

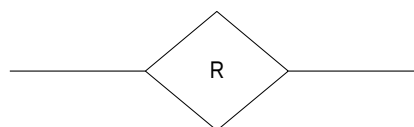
Entity mempunyai elemen yang disebut, dan berfungsi mendeskripsikan karakter entity.



Gambar 2.21 Simbol Atribut

3. Hubungan

Entity dapat berhubungan satu sama lain. Hubungan ini dinamakan *relationship*.

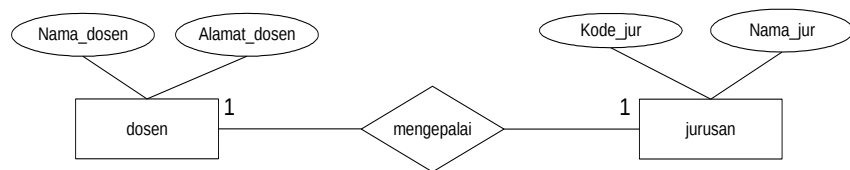


Gambar 2.22 Simbol Relasi Dan Link

Dalam ERD hubungan ini dapat terdiri dari sejumlah *Entity* yang disebut sebagai derajat hubungan. Tetapi pada umumnya hampir semua model hanya menggunakan hubungan dengan derajat dua (*binary-relationship*). Ada tiga jenis hubungan biner yaitu :

1. Satu ke satu

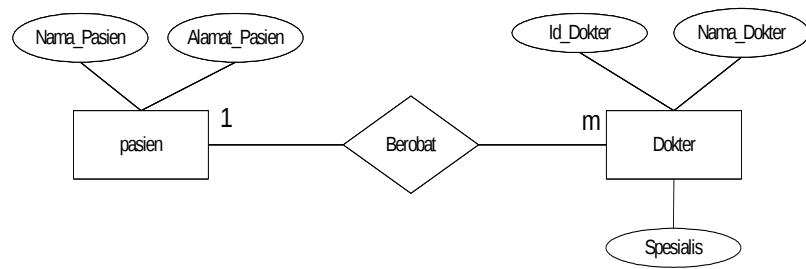
Sebagai contoh dalam suatu badan pendidikan bahwa setiap dosen mengepalai satu jurusan. Maka dalam permodelan dituliskan bentuk hubungan 1 ke 1.



Gambar 2.23 Relasi 1 Ke 1

2. Satu ke banyak dan banyak ke satu

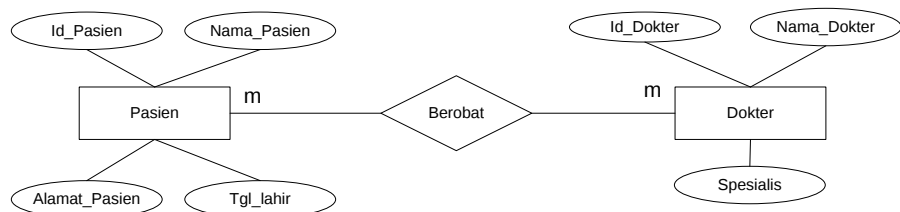
Sebagai contoh dalam suatu badan pendidikan selalu digunakan asumsi bahwa setiap dosen dapat mengajar lebih dari satu mata kuliah, sedang setiap mata kuliah diajar hanya oleh paling banyak satu orang dosen. Maka dalam permodelan dituliskan bentuk hubungan 1 ke m. Hubungan ini jika dibalik dari kanan ke kiri dibaca sebagai hubungan m ke 1.



Gambar 2.24 Relasi 1 Ke Banyak

3. Banyak ke banyak

Sebagai contoh pada relasi ini setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap mata kuliah dapat dipelajari oleh lebih dari satu orang mahasiswa. Maka dalam permodelan dituliskan bentuk m ke m.



Gambar 2.25 Relasi Banyak Ke Banyak

2.4. Basis Data

2.4.1. Pengertian Basis Data

Basis data terdiri atas dua kata, yaitu basis dan data, dapat didefinisikan dalam sejumlah sudut pandang seperti :¹³

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file / arsip / tabel yang saling berhubungan yang disimpan dalam media elektronik.

2.4.2. Sistem Pengelola Basis Data (*Database Management System/DBMS*)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus/spesifik. DBMS merupakan perangkat lunak yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga merupakan mekanisme pengamatan data secara bersama, pemaksaan keakuratan/konsistensi data dan sebagainya. Salah satu tujuan dari DBMS adalah untuk menyediakan fasilitas/antarmuka (*interface*) dalam melihat/menikmati data (yang lebih ramah *user oriented*) kepada pemakai/user.

¹³ Fathansyah, Ir, Basis Data (Bandung : CV. Informatika,1999), hal 2-3.

Perangkat lunak yang termasuk DBMS seperti dBaseIII+, dBaseIV, FoxBase, Rbase, MS-Acces, My SQL, dan sebagainya.

2.4.3. Bahasa Basis Data (*Database Language*)

Database Language merupakan bahasa khusus yang terdiri atas sejumlah perintah (*statement*) yang diformulasikan untuk mengatur cara berinteraksi/berkomunikasi antara pemakai dengan basis data dan dapat diberikan user dan dikenali/diproses oleh DBMS. Bahasa khusus tersebut ditetapkan oleh perusahaan pembuat DBMS. Salah satu contoh bahasa basis data adalah SQL. MySQL sendiri adalah sebuah program database server yang memerlukan sebuah bahasa permintaan dalam melayani permintaan user. SQL merupakan bahasa permintaan yang digunakan dalam MySQL.

SQL (Struktur Query Language) dibagi menjadi dua bentuk query, yaitu :

a. DDL (Data Definition Language)

Struktur atau skema basis data yang menggambarkan/mewakili desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut Data Definition Language (DDL). Dengan bahasa inilah kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dan sebagainya. Query yang termasuk dalam DDL adalah *CREATE,ALTER,DROP*.

b. DML (Data Manipulation Language)

DML merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa penyisipan/penambahan data baru ke suatu basis data,

penghapusan data, pengubahan data di suatu basis data. Query yang termasuk dalam DML adalah *SELECT, INSERT, DELETE, UPDATE*.

2.4.4. Istilah-istilah dalam Basis Data

Berikut adalah uraian tentang pengertian dari beberapa istilah yang digunakan dalam Basis Data, antara lain :

1. Entitas (*Entity*) merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain.
2. Atribut, merupakan karakteristik atau properti dari suatu entitas. Contoh : *entity* set Mahasiswa mempunyai atribut sebagai berikut : NIM (Nomor Induk Mahasiswa), Nama, Tanggal lahir, Alamat.
3. Record, adalah kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap. Suatu *record* mewakili satu data atau informasi tentang seseorang.
4. File, merupakan kumpulan *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda-beda data *valuenya*.

2.5. Dasar Pemrograman PHP dan MySQL

2.5.1. PHP

2.5.1.1. Definisi PHP

PHP adalah singkatan dari "*Hypertext Preprocessor*", yang merupakan sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat.

Halaman web biasanya disusun dari kode-kode html yang disimpan dalam sebuah file berekstensi *.html*. File html ini dikirimkan oleh server (atau file) ke browser, kemudian browser menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan. Lain halnya dengan program php, program ini harus diterjemahkan oleh web-server sehingga menghasilkan kode html yang dikirim ke browser agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode html sehingga dapat langsung ditampilkan bersama dengan kode-kode html tersebut. Program php dapat ditambahkan dengan mengapit program tersebut di antara tanda `<?php` dan `?>`. Tanda-tanda tersebut biasanya disebut tanda untuk *escaping* (kabur) dari kode html. PHP merupakan bahasa pemograman web yang bersifat *server-side HTML=embedded scripting*, di mana script-nya menyatu dengan HTML dan server. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di server tetapi disertakan HTML biasa

PHP dikenal sebagai bahasa scripting yang menyatu dengan tag HTML, dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis

seperti ASP (Active Server Pages) dan JSP (Java Server Pages). PHP pertama kali dibuat oleh Rasmus Lerdorf, seorang programmer C.

Semula php digunakannya untuk menghitung jumlah pengunjung di dalam webnya. Kemudian ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan script PERL yang dibuatnya untuk membuat halaman webnya menjadi dinamis. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah dapat mengakses database dan dapat terintegrasi dengan HTML. Pada tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang softwarenya. Versi terbaru, yaitu PHP 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya Zend Engine yang dibuat oleh Zend Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP scripting engine. Yang lainnya adalah build in HTTP session, tidak lagi menggunakan library tambahan seperti pada PHP. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan web server.

2.5.1.2. Kelebihan PHP

PHP memiliki beberapa kelebihan, diantaranya :

- a. PHP mudah dibuat dan kecepatan akses tinggi.
- b. PHP dapat berjalan dalam web server yang berbeda (apache, xitami, microsoft personal web server, dll) dan dalam sistem operasi yang berbeda pula (windows98, windowsXP, unix, dll).
- c. PHP diterbitkan secara gratis.
- d. PHP adalah termasuk bahasa yang *embedded* (dapat disisipkan) dalam *tag* HTML).
- e. PHP termasuk *server side programming*.

Sebelum memulai membuat program dalam PHP diperlukan perangkat sebagai berikut:

- a. *Text editor* (Notepad, PHPed, Ultra Edit, dll).
- b. *Browser* (IE 4.0 atau lebih, Netscape navigator, dll).
- c. *Web server* yang mendukung PHP.

2.5.1.3. Script PHP

Setiap program disebut dengan *script*. *Script* berupa *file* teks, yang dapat dibuat dengan menggunakan program *editor file* teks (notepad, PHPed, Ultra Edit, dll). *Script* PHP diawali dengan *tag* (<?) Kemudian diakhiri dengan *tag* (?>). Setiap baris perintah/statement harus diakhiri dengan menggunakan tanda titik koma (;). Umumnya setiap statement dituliskan dalam satu baris.

Script PHP merupakan *script* yang digunakan untuk menghasilkan halaman-halaman *web*. Cara penulisan dibedakan menjadi *embedded* dan *non*

embedded script. Nama *file script* PHP menggunakan ekstensi file .php sebagai standart.

2.5.2. MySQL

2.5.2.1. Pengertian MySQL

MySQL adalah sebuah *database server* buatan T.e.X Data Consult AB, Swedia. *Database server* ini sedikit eksentrik, karena bila digunakan di *software* non Microsoft, MySQL adalah gratis, sedangkan bila digunakan di Microsoft Windows sifatnya adalah *shareware* (walaupun tidak memiliki *expire date*).

MySQL di desain agar dapat berjalan sampai 100 akses secara simultan. MySQL didesain untuk berjalan cepat, hal ini yang membuat MySQL menghilangkan kemampuan seperti *transactional processing*. MySQL juga mampu berjalan pada *database* sampai 100 GB. MySQL dapat digunakan sebagai *database* sumber data untuk informasi yang akan ditampilkan pada suatu halaman *web*.

2.5.2.2. Tipe Data dalam MySQL

Perintah INT merupakan singkatan dari INTeger, yaitu jenis data bilangan bulat. VARCHAR merupakan singkatan dari VARiable CHARacter, yaitu jenis data teks/karakter. DATE sendiri merupakan jenis data tanggal (dan waktu). Sedangkan NULL, merupakan jenis data “tanpa nilai” . Secara umum jenis data pada MySQL dibagi menjadi 4 jenis, yaitu :

1. *Numeric Values*, yaitu angka atau bilangan.
2. *String/Character Values*, yaitu teks atau karakter.

3. *Date and Time Values*, yaitu tanggal dan waktu.

4. NULL

Numeric Values adalah angka atau bilangan seperti 10;123;100.50;-10-1.2E+17;2.7e-11; dan sebagainya. Jenis data bilangan dibagi lagi menjadi 2 (dua), yaitu Bilangan Bulat (*Integer*) dan Bilangan Pecahan (*Floating Point*). Bilangan bulat adalah bilangan tanpa tanda desimal, sedangkan bilangan pecahan adalah bilangan dengan tanda desimal. Kedua jenis bilangan ini dapat bernilai positif (+) dan juga negatif (-). Jika bilangan tersebut menggunakan tanda (+) atau (-), maka disebut SIGNED. Sebaliknya, bila tanpa tanda apapun disebut UNSIGNED. Karena tanda (+) dapat diabaikan penulisannya, maka pada bilangan yang bernilai positif disebut juga *Unsigned*.

Character Values, adalah semua karakter (atau teks) yang penulisannya selalu di apit oleh tanda kutip, baik kutip tunggal (') maupun kutip ganda ("). Hal ini tidak hanya berlaku pada huruf alfabet saja, tetapi angka yang ditulis dengan tanda kutip pun akan menjadi karakter atau string.

2.5.3. Integrasi dan Interaksi MySQL dan PHP

PHP memiliki banyak fungsi yang dapat digunakan sebagai *interface* pada MySQL sehingga data yang tersimpan dalam *database* MySQL dapat dilihat secara interaktif melalui media *internet*. Interaksi PHP dengan *database* MySQL yang terjadi ketika *klien* mengakses *web server* yang mendukung PHP dan *database* MySQL melalui *browser*, antara lain ketika *server* menerima dan membaca permintaan dari browser, *server* akan mengeksekusi perintah-perintah dalam kode PHP. Kemudian PHP akan melakukan *query* (permintaan) pada

database MySQL melalui Application programming Interface (API) dan mengkompilasi hasilnya. Selanjutnya web server akan mengirim halaman hasil ke browser klien.

PHP terdiri atas rangkaian fungsi-fungsi *interface* (antarmuka) pada MySQL. Kita dapat membuat aplikasi yang *powerful* dengan hanya menggunakan fungsi-fungsi sederhana.