BAB II
DASAR TEORI

Pada bab ini dijelaskan kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai pembuatan Web Interaktif pada Lembaga Bimbingan Belajar IECC Nganjuk.

2.1 Sistem Informasi

Sesungguhnya yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer-Based Information Systems* atau CBIS). Dalam prakteknya, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting.

2.1.1 Pengertian Sistem

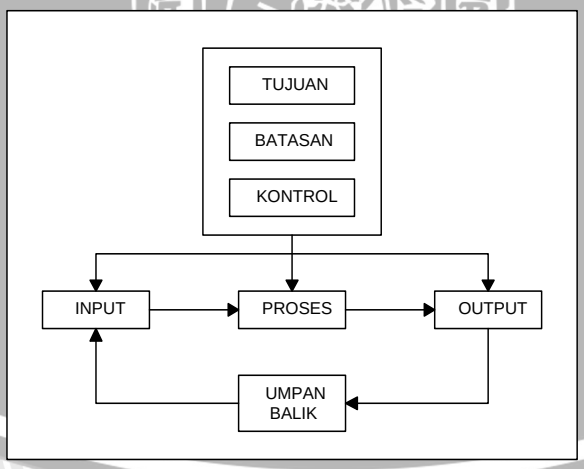
Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sistem didefinisikan himpunan komponen atau variabel yang terorganisasi, saling berinteraksi, saling bergantung satu sama lain dan terpadu [MCL-95:9]. Menurut Murdick definisi sistem lebih ditekankan pada prosedur, sistem adalah seperangkat elemen yang membentuk suatu prosedur yang mencari suatu tujuan bersama dengan mengoperasikan data pada waktu tertentu untuk menghasilkan informasi [MUR-97:16]. Sedangkan menurut Jogiyanto sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu [JOG-90:1].

Dari berbagai definisi di atas, secara umum sistem dikelompokkan menjadi beberapa pengertian dasar, yaitu : terdiri dari elemen-elemen yang terintegrasi, membentuk suatu prosedur untuk mencapai suatu tujuan dan menghasilkan suatu keluaran (*output*).

Seperti yang telah disebutkan di atas, sistem terdiri dari elemen-elemen yang saling berkaitan. Elemen-elemen yang menyusun sebuah sistem tersebut terdiri dari:

1. *Tujuan*, merupakan tujuan dari sistem tersebut.
2. *Batasan*, merupakan batasan-batasan yang ada dalam mencapai tujuan dari sistem.
3. *Kontrol*, merupakan pengawas dari pelaksanaan pencapaian tujuan sistem.
4. *Input*, merupakan bagian dari sistem yang bertugas untuk menerima data masukan.
5. *Proses*, merupakan bagian yang memproses masukan data menjadi informasi sesuai dengan keinginan penerima.
6. *Output*, merupakan keluaran atau tujuan akhir dari sistem.
7. *Umpan balik*, merupakan informasi dari sistem dan menyediakan informasi bagi kontrol dimana umpan balik ini dapat berupa perbaikan, pemeliharaan dan lain-lain.

Keterkaitan elemen-elemen sistem dapat dilihat dalam Gambar 2.1.



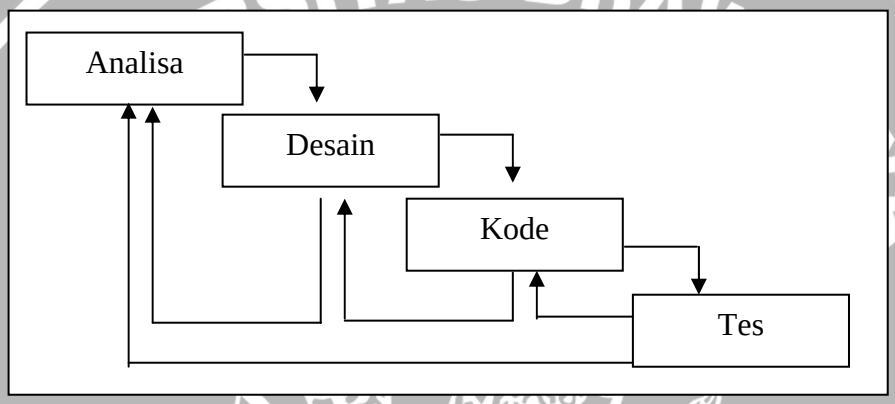
Gambar 2.1 Elemen-elemen sistem.

Sumber : Mahyuzir, T.D, 1997 : 11

2.1.2 Tahapan Perancangan dan Pengembangan Sistem Perangkat Lunak

Dalam tahap perancangan dan pengembangan sistem, Pressman mengemukakan bahwa terdapat suatu model yang sering juga disebut dengan model Sekuensial Linier atau disebut juga dengan *Waterfall* model [ROG-02:36].

Model ini mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode dan pengujian. Model tersebut dapat dilihat dalam Gambar 2.2.



Gambar 1.5 Model Sekuensial Linier.
Sumber : Pressman, R.S 2006 : 37

1. Analisis kebutuhan perangkat lunak, merupakan proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, analisis harus memahami domain informasi, tingkah laku, unjuk kerja dan *interface* yang diperlukan.
2. Desain, merupakan proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, antara lain struktur data, arsitektur perangkat lunak, representasi *interface* dan detail prosedural.
3. Generasi Kode, merupakan proses penterjemahan desain ke dalam bentuk mesin yang dapat dibaca.
4. Tes atau Pengujian, merupakan sebuah proses yang memfokuskan pada logika internal perangkat lunak, dengan memastikan bahwa semua pernyataan sudah

diuji dan untuk menemukan kesalahan sehingga dapat memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

2.1.3 Pengertian Informasi

Informasi adalah sesuatu yang nyata atau setengah nyata yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan atau kejadian. Informasi adalah data yang telah diproses [MCL-95:12]. Menurut Jogiyanto informasi adalah data yang telah diolah ke dalam suatu bentuk yang berguna bagi penerimanya dan nyata atau berupa nilai yang dapat dipahami dalam keputusan sekarang maupun masa depan [JOG-90:8]. Sedangkan menurut Murdick informasi adalah data yang telah diambil kembali, diolah, atau sebaliknya digunakan untuk tujuan informatif atau kesimpulan, argumentasi, atau sebagai dasar untuk peramalan atau pengambilan keputusan [MUR-97:5].

Dari segi kualitas, informasi harus dapat memenuhi syarat-syarat sebagai berikut:

1. Lengkap

   Informasi harus dapat menyajikan gambaran lengkap dari suatu permasalahan atau suatu penyelesaian.

2. Akurat

   Informasi idealnya harus memiliki keakuratan data yang berkaitan langsung dengan permasalahan yang ada.

3. Relevan, informasi memiliki relevansi jika berkaitan langsung dengan permasalahan yang ada.

4. Tepat waktu

   Informasi harus tersedia untuk pemecahan masalah sebelum situasi krisis menjadi tidak terkendali.

2.1.4 Pengertian Sistem Informasi

Sistem informasi didefinisikan sebagai sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil

keputusan atau untuk mengendalikan organisasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [JOG-90:11].

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok yaitu:

1. Blok Masukan (*Input Block)*, yaitu data-data yang masuk ke dalam sistem informasi. Masukan *(Input)* di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.
2. Blok Model, terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data masukan dan data yang akan tersimpan di basis data.
3. Blok Keluaran *(Output Block)*, adalah produk dari sistem informasi yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok Teknologi *(Technology Block)*, merupakan kotak alat *(toolbox)* dalam sistem informasi, yang digunakan untuk menerima masukan, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian, yaitu teknisi *(humanware atau brainware)*, perangkat lunak *(software)* dan perangkat keras *(hardware).*
5. Blok Basis Data, merupakan kumpulan dari data yang saling berhubungan satu sama lain, tersimpan di perangkat keras komputer dan digunakan oleh perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut.
6. Blok Kendali *(Control Block)*, beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah, seperti sabotase, kegagalan sistem dan sebagainya.

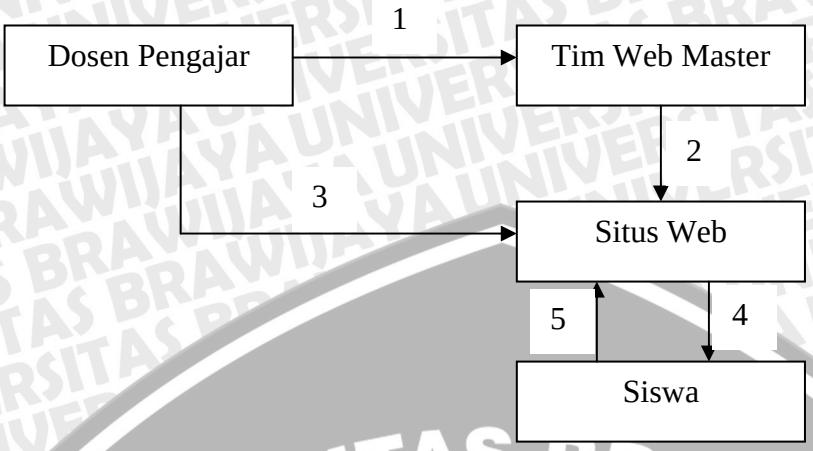
2.2. Sistem Web Interaktif

Istilah web interaktif dapat didefinisikan sebagai sebuah bentuk teknologi informasi yang diterapkan di bidang pendidikan dalam bentuk sekolah maya. web interaktif lebih ditujukan sebagai usaha untuk membuat sebuah transformasi proses belajar mengajar yang ada di sekolah ke dalam bentuk digital yang dijembatani oleh teknologi internet. Fokus web interaktif lebih pada efisiensi proses belajar-mengajar, cara pengajaran maupun materi ajar masih dapat mengacu pada kurikulum nasional [PUR-03:1].

Dalam teknologi web interaktif, semua proses belajar mengajar yang biasa didapatkan di dalam sebuah kelas dilakukan secara live namun virtual, artinya pada saat yang sama seorang guru mengajar di depan sebuah komputer yang ada di suatu tempat, sedangkan para siswa mengikuti pelajaran tersebut dari komputer lain di tempat yang berbeda. Dalam hal ini, secara langsung guru dan siswa tidak saling berkomunikasi namun secara tidak langsung mereka saling berinteraksi pada waktu yang sama. Materi pelajaran pun dapat diperoleh secara gratis dalam bentuk file-file yang dapat di download. Sedangkan interaksi antara guru dan siswa dalam bentuk forum diskusi dan e-mail.

Perkembangan teknologi informasi yang sangat pesat dewasa ini, khususnya perkembangan teknologi internet turut mendorong berkembangnya konsep pendidikan jarak jauh ini. Ciri teknologi internet yang selalu dapat diakses kapan saja, dimana saja, multi user serta menawarkan segala kemudahan telah menjadikan internet suatu media yang sangat tepat bagi perkembangan pendidikan jarak jauh selanjutnya. Bila dibatasi pada pendidikan jarak jauh berbasis internet, maka pengguna atau dalam hal ini pengajar dan siswa memerlukan fasilitas internet untuk tetap menjaga konektivitas dengan pendidikan jarak jauh tersebut. Kemampuan peserta untuk menjaga konektivitas menentukan bagi kesinambungan suatu sistem pendidikan jarak jauh.

Suatu pendidikan jarak jauh dapat diformulasikan seperti gambar berikut :



Gambar 2.1 Aliran informasi situs web interaktif

Pada gambar 2.1 diatas dapat dilihat aliran dari informasi situs web interaktif, dimana terdapat empat komponen yang terlibat yaitu : dosen pengajar, tim web master, situs web, dan siswa.

Dimana proses yang terjadi antara lain yang pertama adalah dosen pengajar memberikan materi kuliah dan informasi umum tentang kuliah kepada tim web master . Kedua, pengerjaan web oleh tim web master. Selanjutnya yang ketiga, dosen pengajar secara langsung mengubah isi situs web untuk memberikan informasi dan jawaban atas pertanyaan (diskusi). Yang keempat, siswa mendapatkan informasi kuliah. Dan yang terakhir, siswa melakukan tanya jawab (diskusi).

Melalui pemanfaatan e-learning, akan dapat diperoleh beberapa keuntungan yang cukup besar dibandingkan dengan usaha pembangunan sekolah konvensional. Adapun keuntungannya yaitu, keuntungan secara finansial dimana keuntungan ini diperoleh dari berkurangnya biaya yang diperlukan untuk mengimplementasikan sistem secara keseluruhan jika dibandingkan dengan biaya untuk mendirikan bangunan sekolah beserta seluruh perangkatnya termasuk pengajar. Dan juga keuntungan dari segi efisiensi waktu dengan tidak diperlukannya berada di dalam kelas, namun dari segala tempat yang dapat mengkases internet saja, waktu perjalanan dapat ditekan seminimal mungkin.

2.3. Basis Data

2.3.1 Definisi

Sistem basis data adalah sebuah sistem penyimpan record terkomputerisasi, yaitu sistem terkomputerisasi yang bertujuan untuk menghimpun informasi dimana informasi tersebut tersedia bila diperlukan. Sistem ini meliputi data (tersimpan dalam database), hardware, software dan yang paling utama adalah user [DAT-03:1].

2.3.2 Database Management System

Database Management system (DBMS) terdiri dari sekumpulan data yang saling berelasi dan serangkaian program program untuk mengakses data tersebut. Tujuan utama DBMS adalah menyediakan sebuah lingkungan yang sesuai dan efisien untuk mengambil dan menyimpan informasi database [SIL-04:1].

Pengolahan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat yang disebut dengan DBMS. DBMS akan memastikan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secada bersama, pemaksaan keakuratan/konsistensi data dan sebaginya.

Berdasarkan orientasi pemakaiannya, kita dapat mengelompokkan DBMS dalam 2 kategori yaitu :

1. DBMS yang berorientasi untuk satu atau sedikit pemakai.

   DBMS yang termasuk dalam kelompok ini antara lain : MS-Access, dBase/Clipper, FoxBase, Borland-Paradox.

2. DBMS yang berorientasi untuk banyak pemakai.

   DBMS yang termasuk dalam kelompok ini antara lain : Oracle, Borland-Interbase, MS-SQL Server, CA-OpenIngres, Sybase, Informix, IBM DB2.

   DBMS ini lebih ditujukan untuk pemakaian pada sistem jaringan komputer.

2.3.3 Model Basis Data Relasional

Model Basis Data Relasional sering pula disebut sebagai model relasional atau basis data relasional. Pada model relasional, basis data akan disebar atau dipilah-pilah ke dalam tabel 2 dimensi. Setiap tabel selalu terdiri atasa lajur mendatar yang disebut dengan baris data (Row/Record) dan lajur vertikal yang biasa disebut dengan kolom (Column/Field).

Di setiap pertemuan baris data dan kolom itulah item-item data (satuan data terkecil) ditempatkan. Dalam kehidupan sehari-hari tabel merupakan bentuk natural dalam menyatakan fakta (data yang sering kita gunakan). Itulah sebabnya model ini lebih mudah kita terpakan daripada model basis data yang lain.

Ada 4 macam *key* yang dapat diterapkan pada suatu tabel dalam model basis data relasional, antara lain :

- *Superkey*

  Merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.
- *Candidate-Key*

  Merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.
- *Primary-Key*

  Merupakan *key* yang diambil dari beberapa *candidate-key*, dengan didasari pada :
  - Key tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan.
  - Key tersebut lebih ringkas.
  - Jaminan keunikan key tersebut lebih baik.
- *Foreign Key*

  Merupakan kunci tamu atau kunci asing dari suatu tabel dimana kunci ini juga terdapat pada tabel lain sebagai *primary key*.

[DAT-03:45] menyebutkan beberapa kelebihan sistem database dibandingkan dengan sistem tradisional dengan metode penyimpan record berbasiskan kertas, yaitu :

1. *Compactness* : tidak memerlukan file kertas yang tebal.
2. *Speed* : mesin bias mengambil dan mengubah data lebih cepat daripada manusia.
3. *Less drudgery* : tugas mekanik selalu lebih baik bila dikerjakan oleh mesin.
4. *Currency* : informasi yang akurat.

2.3.4. Normalisasi

Normalisasi adalah suatu teknik yang terstruktur dan dalam cara-cara tertentu yang membantu mengurangi atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data dalam basis data. Normalisasi juga dapat diartikan sebagai proses mengubah relasi menjadi bentuk normal atau proses untuk mengidentifikasi dan mengeliminasi anomali [KRO-02:1]. Proses normalisasi menghasilkan struktur *record* yang konsisten secara logik, yang mudah untuk dimengerti, dan sederhana dalam pemeliharaannya.

Terdapat 3 bentuk normal yang diidentifikasikan dalam struktur file pada relasi. Masing-masing bentuk normal mempunyai batasan tertentu yang dapat dipergunakan untuk menentukan nama bentuk normalnya. Adapun bentuk normal tersebut adalah :

1. Bentuk Normal Pertama (1NF), suatu relasi berada dalam bentuk normal pertama jika semua atribut mempunyai nilai yang bersifat atomik, tetapi masih terdapat beberapa atribut yang muncul secara berulang *(redundancy)*.
2. Bentuk Normal Kedua (2NF), jika suatu relasi berada pada kondisi normal pertama dan mengalami kondisi sebagai berikut :
   a. Jika *primary key* hanya terdiri dari satu atribut.
   b. Atribut lain selain *primary* key tidak dapat berperan seperti *primary key*, melainkan hanya merupakan bagian atau komponen dari *primary key*.
   c. Setiap *record* dalam relasi tersebut, fungsinya tergantung pada fungsi *primary key*.
3. Bentuk Normal Ketiga (3NF), relasi dikatakan dalam bentuk normal ketiga jika relasi berada pada kondisi normal kedua dan setiap atribut bukan kunci

tidak memiliki *dependency transitif* terhadap kunci primer. Dengan kata lain setiap atribut bukan kunci haruslah bergantung pada *primary key* dan pada *primary key* secara menyeluruh.

Berikut akan diberikan contoh untuk tabel mahasiswa yang belum normal yang dapat dilihat pada tabel 2.1 dan tabel mahasiswa yang sudah dalam keadaan normal setelah dilakukan proses normalisasi yang dapat dilihat pada tabel 2.2.

Tabel 2.1 Contoh tabel belum normal

Nim	Nama	Alamat	Tgl_Lahir	Hobi
001	Ali	Bandung	03/03/1981	Berenang
002	Budi	Yogya	12/05/1979	Membaca
003	Nur	Jakarta	10/12/1980	Sepakbola
004	Sari	Surabaya	10/09/1980	Bernyanyi

Tabel 2.2 Contoh tabel yang sudah normal

Tabel Siswa

Nim	Nama	Alamat	Tgl_Lahir
001	Ali	Bandung	03/03/1981
002	Budi	Yogya	12/05/1979
003	Nur	Jakarta	10/12/1980
004	Sari	Surabaya	10/09/1980

Tabel Hobi

Nim	Hobi
001	Berenang
002	Membaca
003	Sepakbola
004	Bernyanyi

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.4. Konsep dan Prinsip Desain

Desain adalah langkah pertama dalam fase pengembangan bagi setiap produk atau sistem yang direkayasa. Desain dapat didefinisikan sebagai "proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya". [ROG-02:400]

Desain digambarkan sebagai proses multi-langkah di mana representasi struktur data, struktur program, karakteristik *interface* dan detail prosedur, disintesis dari persyaratan informasi.

2.4.1. Desain Arsitektur

Sasaran utama desain arsitektur adalah untuk mengembangkan struktur program modular dan merepresentasikan hubungan kontrol antar modul. Desain arsitektur juga membentuk struktur program dan struktur data dengan menentukan *interface* yang memungkinkan data mengalir melalui program.

Desain yang berorientasi pada aliran data merupakan suatu metode desain arsitektur yang mengijinkan transisi yang baik dari model analisis ke deskripsi desain dari struktur program. Transisi dari aliran informasi (yang ditunjukkan sebagai diagram aliran data) ke struktur dilakukan sebagai bagian dari proses lima langkah:

1. Tipe aliran informasi dibangun
2. Batas aliran informasi ditentukan
3. *Data Flow Diagram* dipetakan ke dalam struktur program
4. Hirarki kontrol ditentukan dengan percabangan aliran informasi
5. Struktur resultan disaring atau diperhalus dengan menggunakan teknik heuristik. Teknik heuristik adalah teknik yang digunakan untuk mengeliminasi beberapa kemungkinan solusi tanpa harus mengeksplorasinya secara penuh. Selain itu, teknik heuristik juga membantu memutuskan kemungkinan solusi mana yang pertama kali perlu dievaluasi. Tipe aliran informasi adalah kendali untuk pendekatan pemetaan yang diperlukan pada langkah 3.

Dua tipe aliran yang ada pada proses desain: [ROG-02:440]

1. Aliran Transfomasi

   Informasi memasuki sistem bersama dengan jalur yang mentransformasikan data eksternal ke dalam bentuk internal dan akan diidentifikasi sebagai aliran masuk. Pada inti perangkat lunak terjadi transisi. Data yang masuk dilewatkan melalui pusat *transformasi* dan mulai bergerak sepanjang jalur yang sekarang mengarah keluar dari perangkat lunak. Data yang mengalir di sepanjang jalur-jalur itu disebut aliran keluar. Keseluruhan aliran data terjadi dalam cara yang berurutan dan mengikuti satu atau hanya beberapa jalur "garis lurus". Bila segmen dari diagram aliran data menunjukkan karaktersitik tersebut, maka di situ ada aliran *transformasi*.

2. Aliran Transaksi

   Model sistem fundamental mengimplikasikan aliran *transformasi*, karena itu untuk mengkarakterisasi semua aliran data dalam kategori ini. Tetapi aliran informasi sering ditandai oleh sebuah *item* data tunggal, yang disebut transaksi, yang memicu aliran data lain sepanjang salah satu dari beberapa jalur. Aliran transaksi ditandai dengan pergerakan data sepanjang jalur masuk yang mengkonversikan informasi dunia eksternal ke dalam suatu transaksi. Transaksi tersebut dievaluasi, dan berdasarkan nilainya, aliran sepanjang satu dari beberapa jalur aksi diinisiasi. Pusat aliran informasi dari mana banyak jalur aksi berasal disebut pusat transaksi.

Langkah–langkah Desain Pemetaan Transformasi: [ROG-02:441]

1. Mengkaji model sistem fundamental.
2. Mengkaji diagram aliran data untuk perangkat lunak tersebut dan disaring atau diperhalus.
3. Menentukan apakah DFD memiliki karakteristik aliran transformasi atau transaksi.
4. Isolasi pusat transformasi dengan mengkhususkan batas aliran masuk dan keluar.
5. DFD dipetakan ke suatu struktur khusus yang memberikan kontrol bagi pemrosesan informasi masuk, transformasi dan informasi keluar.
6. Memetakan transformasi individual dari suatu DFD ke dalam modul–modul yang sesuai dalam struktur program.

7. Struktur program iterasi pertama disaring dengan menggunakan heuristik desain bagi kualitas perangkat lunak yang telah ditingkatkan. Penyaringan ditentukan saat pengontrol untuk aliran data masuk sama sekali tidak diperlukan.

Langkah – langkah Desain Pemetaan Transaksi: [ROG-02:441]

1. Mengkaji model sistem fundamental.
2. Mengkaji diagram aliran data untuk perangkat lunak tersebut dan disaring atau diperhalus.
3. Menentukan apakah DFD memiliki karakterstik aliran transformasi atau transaksi.
4. Identifikasi pusat transaksi dan karakteristik aliran sepanjang masing-masing jalur aksi
5. Aliran transaksi dipetakan ke dalam suatu struktur program yang sesuai dengan pemrosesan transaksi (cabang masuk dan cabang pengiriman).
6. Memfaktorkan struktur transaksi dan struktur masing-masing jalur aksi dari diagram aliran data berdasarkan karakteristik aliran informasi.
7. Pemetaan transaksi struktur program iterasi pertama dengan menggunakan heuristik desain untuk kualitas perangkat lunak yang dikembangkan.

2.5. Diagram Arus Data

Data Flow Diagram (DFD) adalah sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output* [ROG-02:364]. DFD sering digunakan untuk menggambarkan sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. DFD merupakan dokumentasi dari sistem yang baik. Komponen-komponen dari diagram arus data:

1. Kesatuan luar (*external entity*) atau Batas Sistem (*boundary)*

   Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Dalam hal ini, sejumlah *input* dapat

menjadi hanya satu *output* ataupun sebaliknya. Kesatuan luar merupakan kesatuan di luar lingkungan sistem yang dapat berupa orang, organisasi atau sistem yang lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. Arus Data (*data flow*)

   Komponen ini direpresentasikan dengan menggunakan tanda panah yang menuju ke atau dari proses. Alir data ini menunjukkan arus data yang dapat berupa masukan untuk sistem atau hasil dari suatu proses sistem. Beberapa kesalahan yang harus dihindari dalam membuat DFD (*Data Flow Diagram*) antara lain adalah tidak diperbolehkan:

   1. Menghubungkan satu tempat penyimpanan dengan tempat penyimpanan yang lain.
   2. Membuat hubungan langsung tempat penyimpanan dengan entitas luar.
   3. Membuat hubungan langsung entitas luar dengan entitas luar yang lain.

3. Proses (*process*)

   Proses menunjukkan transformasi dari masukan sampai keluaran. Suatu proses terjadi karena adanya arus data yang masuk dan hasil dari proses adalah juga merupakan arus data yang mengalir. Suatu proses harus menerima arus data dan menghasilkan arus data. Beberapa hal yang harus dihindari dalam menggambarkan suatu proses adalah:

   1. Proses yang mempunyai masukan tetapi tidak memiliki keluaran, kesalahan ini disebut *Black Hole* (lubang hitam).
   2. Proses yang menghasilkan keluaran tetapi tidak mendapatkan masukan atau proses yang menghasilkan keluaran lengkap dengan data terbatas, kesalahan ini disebut dengan *miracle* (ajaib).

4. Penyimpanan Data (*data store*)

   Penyimpanan data digunakan untuk menyimpan data hasil proses maupun menyediakan data untuk diproses pada waktu yang lain. Beberapa ketentuan dalam menggambarkan penyimpanan data:

   1. Hanya proses saja yang dapat berhubungan dengan penyimpanan data.

2. Arus data yang menuju ke simpanan data dari suatu proses menunjukkan proses *update* terhadap data yang tersimpan di simpanan data.
3. Arus data yang berasal dari simpanan data ke suatu proses menunjukkan bahwa proses tersebut menggunakan data yang ada di simpanan data.

Tabel 2.1 Simbol Data Flow Diagram.

Sumber: Jogiyanto 1990 : 701

SIMBOL	NAMA	FUNGSI
	Entitas Luar	Merupakan kesatuan di luar sistem yang memberikan *input* dan *output* kepada sistem.
	Alir Data	Menunjukkan arus data yang dapat berupa masukan atau hasil dari proses sistem.
	Proses	Menunjukkan transformasi dari masukan sampai keluaran.
	Penyimpanan Data	Untuk menyimpan data hasil proses maupun menyediakan data untuk diproses di waktu lain.

Terdapat 2 bentuk DFD yaitu diagram arus data fisik (*physical data flow diagram*) dan diagram arus data logika (*logical data flow diagram*). Diagram arus data fisik lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama) dan lebih ditekankan pada bagaimana proses dari sistem diterapkan (dengan cara bagaimana, oleh siapa dan dimana), termasuk proses-proses manual sedangkan diagram arus data logika lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem baru) dan lebih ditekankan logika dari kebutuhan sistem, yaitu proses-proses apa secara logika dibutuhkan oleh sistem. Simbol-simbol dalam DFD tersebut dapat dilihat dalam Tabel 2.1.

Diagram konteks merupakan aliran yang menggambarkan hubungan antara sistem dengan kesatuan luar di luar sistem, yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Aliran dalam diagram konteks memodelkan masukan ke sistem dan keluaran dari sistem. Aliran data hanya digambarkan dan hanya diperlukan untuk mendeteksi kejadian dalam

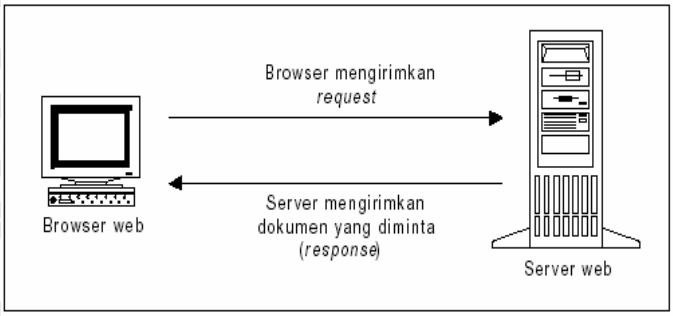
lingkungan sistem dan harus memberikan respon atau membutuhkan untuk menggambarkan transportasi antara sistem dan eksternal entitas.

2.6 Internet

Internet berasal dari kata *interconnection networking* yang mempunyai arti hubungan berbagai komputer dan berbagai tipe komputer yang membentuk sistem jaringan yang mencakup seluruh dunia (jaringan global) dengan melalui jalur telekomunikasi seperti telepon, wireless dan lainnya. Dapat disimpulkan Internet merupakan hubungan antar berbagai jenis komputer dan jaringan di dunia yang berbeda sistem operasi maupun aplikasinya dimana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protokol standar dalam berkomunikasi yaitu protokol TCP/IP. Perlu dibedakan antara sebuah internet dan Internet, karena Internet adalah definisi resmi dari internet secara global [ANR-07].

2.6.1 *World Wide Web*

World Wide Web (WWW) adalah jaringan beribu - ribu komputer yang dikategorikan menjadi dua yaitu *client dan server*. *World Wide Web* atau lebih sering dikenal sebagai *Web* adalah layanan Internet yang paling banyak memiliki tampilan grafis dan kemampuan link yang sangat bagus. Keistimewaan inilah yang telah menjadikan *Web* sebagai *service* yang paling cepat pertumbuhannya. *Web* mengijinkan pemberian *highlight* (penyorotan/penggaris bawahan) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, *frase*, *movie clip* atau file suara. *Web* dapat menghubungkan dari sebuah dokumen atau gambar ke dokumen lain. Dengan sebuah *browser* yang memiliki *Graphical User Interface* (GUI), *link-link* dapat dihubungkan ke tujuannya dengan menunjuk *link* tersebut dengan mouse dan menekannya. Dalam cara kerja dari *World Wide Web* ada dua hal yang terpenting yaitu *software web server* dan *software web browser* [WEB-07:57]. Cara kerja dari *World Wide Web* dapat dilihat dalam Gambar 2.6.



Gambar 2.6 Konsep dasar pada *browser* dan *web serve.*

Sumber : Sutarman, 2003 : 5

Server menyimpan atau menyediakan informasi dan memproses permintaan dari *client*, apabila ada *client* yang meminta informasi maka *server* akan mengirimkannya. Informasi yang diakses dapat berupa teks, gambar maupun suara. Server juga mengirim perintah-perintah ke *client* tentang bagaimana cara menampilkan semua informasi tersebut.

2.6.2 *HyperText Transfer Protocol*

HyperText Transfer Protocol (HTTP) adalah suatu protokol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan perlu diikuti juga oleh *web server* dalam menyediakan dokumen yang diminta *web browser*. Protokol ini merupakan protokol standar yang digunakan untuk mengakses dokumen HTML (*Hypertext Markup Language*). HTML merupakan program penulisan informasi sebuah *homepage*. Data yg di-*passing* dari *browser* ke *Web server* disebut sebagai *HyperText Transfer Protocol Request* yg meminta *web page* dan kemudian *web server* akan mencari data HTML yg ada dan dikemas dalam TCP *protocol* (*Transmission Control Protocol*). TCP merupakan satu set protokol standar yang digunakan untuk menghubungkan jaringan komputer dan mengalamati lalu lintas dalam jaringan dan dikirim kembali ke *browser*. Data yg dikirim dari *server* ke *browser* disebut sebagai *HyperText Transfer Protocol response* [ANS-07].

2.6.3 Uniform Resource Locator

Uniform Resource Locator (URL) adalah suatu sarana yang digunakan untuk menentukan lokasi informasi pada suatu *web server* [ANT-07]. *Uniform Resource Locator* dapat diibaratkan suatu alamat dimana alamat tersebut terdiri atas:

1. Protokol yang digunakan oleh suatu *browser* untuk mengambil informasi
2. Nama komputer (*server*) dimana informasi tersebut berada
3. Jalur atau path serta nama file dari suatu informasi

Format umum dari URL adalah sebagai berikut: *Protokol_transfer : / / nama_host / path / nama_fil*. Contoh: http://www.amazon.com / buku / index.html

Website (situs *web*) adalah alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasar pada topik tertentu. *Web Page* (halaman *web*) merupakan halaman khusus dari situs *web* tertentu yang tersimpan dalam bentuk file. *Home page* merupakan halaman pertama atau sampul dari suatu *website* yang biasanya berisi tentang apa dan siapa dari perusahaan atau instansi ataupun organisasi pemilik *website* tersebut.

2.6.4 Pemrograman Web

Situs atau *web* dapat dikategorikan menjadi dua yaitu *web statis* dan *web dinamis* atau *interaktif*. *Web statis* adalah *web* yang berisi atau menampilkan informasi-informasi yang sifatnya statis atau tetap, sedangkan *web dinamis* adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan *user* yang sifatnya dinamis. Sehingga untuk membuat *web* dinamis dibutuhkan kemampuan pemrograman *web*. Dalam pemrograman *web* ada 2 kategori yaitu *Server-side Programming* dan *Client-side Programming* [ANU-07].

Pada *Server-side Programming*, perintah-perintah program (*script*) dijalankan di *web server*, kemudian hasilnya dikirimkan ke *browser* dalam bentuk HTML biasa, contohnya CGI/Perl, ASP (*Active Server Pages*), JSP (*Java Server Pages*), PHP, CFM (*ColdFussion*) dan lain-lain.

Contoh *script* PHP:

```
<HTML>
<HEAD>
<TITLE>Test</TITLE>
</HEAD>
<BODY>

Test 1.<BR>
<?php
    printf ("Tgl. Sekarang : %s ", Date ("d F Y"));
?>
</BODY>
</HTML>
```

Dari *script* PHP di atas dapat dilihat hasilnya dalam Gambar 2.7 sebagai berikut:

Gambar 2.7 Hasil *script* PHP

Sedangkan *Client-side Programming* perintah program dijalankan di *web browser*, sehingga ketika *client* meminta dokumen yang mengandung *script*, maka *script* tersebut akan di-*download* dari *server*-nya yang kemudian dijalankan di *browser* yang bersangkutan, contohnya JavaScript, VbScript, HTML dan lain-lain.

2.7 Structured Query Language

Structured Query Language (SQL) adalah bahasa *non prosedural* yang maksudnya menjelaskan bagaimana data ditampilkan, dihapus atau dimasukkan dan bukan menjalankan prosedur pemrograman untuk menampilkan data, [LOR-91].

SQL merupakan bahasa standar yang meliputi perintah - perintah untuk menyimpan, menerima, memelihara dan mengatur akses - akses ke basis data serta digunakan untuk memanipulasi dan menampilkan data.

SQL membuat *programmer* atau *database administrator* dapat melakukan hal-hal berikut:

1. Memodifikasi struktur *database*.
2. Menambahkan wewenang *user* pada *database* atau tabel.
3. Menampilkan informasi dari *database*.
4. Mengubah isi dari *database*.
5. Membuat keamanan data.

Penggunaan SQL sebagai *database* dapat dikoneksikan dengan PHP dengan beberapa parameter berikut [ANV-07]:

1. *Server name*, merupakan nama *server* atau no. IP(*Internet Protocol*) *server* dimana MySQL tersebut di-*install*.
2. *Username*, merupakan nama *user* yang diberikan wewenang untuk mengakses *database* dalam MySQL.
3. *Password*, merupakan *password* yang dimiliki *username* dalam rangka pengelompokan *user* dan keamanan.
4. *Database name*, merupakan nama *database* dalam MySQL yang diakses.

Perintah yang paling sering dipakai dalam SQL adalah perintah SELECT yang berfungsi untuk menampilkan data dari *database*. Perintah SQL lain adalah DELETE, CREATE dan UPDATE yang digunakan untuk menghapus, membuat dan memodifikasi *database*.

2.8 *HyperText Markup Language*

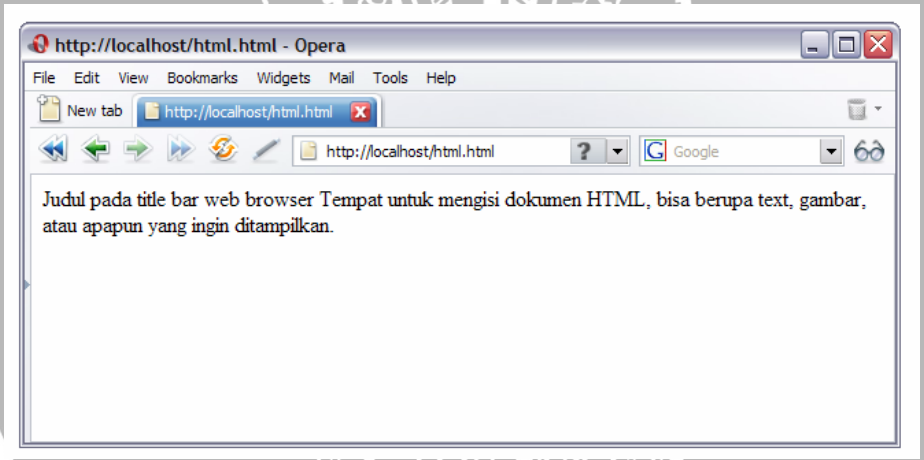
HyperText Markup Language (HTML) digunakan untuk membangun suatu halaman *web*. HTML sebenarnya bukan merupakan bahasa pemrograman, karena seperti namanya HTML merupakan suatu bahasa Markup (penandaan).

Tag HTML biasanya berupa *tag-tag* yang berpasangan dan ditandai dengan simbol '<' dan'>'. Pasangan atau penutup perintah dari sebuah *tag* ditandai dengan tanda '/'. Misalnya pasangan dari *tag* <contoh> adalah </contoh>.

Setiap dokumen atau halaman HTML memiliki struktur atau susunan *file* sebagai berikut :

```
<HTML>
<HEAD><TITLE>Judul pada title bar web browser<TITLE>
</HEAD>
<BODY>
Tempat untuk mengisi dokumen HTML, bisa berupa text, gambar, atau apapun yang ingin ditampilkan.
</BODY>
</HTML>
```

Jika kita simpan dalam format HTML, hasil pada *browser* akan terlihat seperti gambar berikut :



Gambar 2.8 Hasil Script HTML

Seperti terlihat, struktur *file* HTML diawali dengan sebuah *tag* <HTML> dan ditutup dengan *tag* </HTML>. Di dalam *tag* ini terdapat dua buah bagian besar, yaitu yang diapit oleh *tag HEAD* merupakan *header* dari halaman HTML dan tidak di tampilkan pada *browser*. Bagian ini berisi *tag-tag header* seperti <TITLE> ... </TITLE> yang berfungsi untuk mengeluarkan judul pada *title bar window web browser*.

Bagian kedua, yang diapit oleh *tag BODY* merupakan bagian yang akan ditampilkan pada halaman *web browser* nantinya. Pada bagian ini akan dituliskan

semua jenis informasi berupa teks dengan bermacam format maupun gambar yang ingin disampaikan[ANZ-07].

2.9 Personal Home Page

Personal Home Page (PHP) merupakan bahasa berbentuk *script* yang ditempatkan dalam *server* dan diproses di *server*, yang secara khusus dirancang untuk membentuk *web* dinamis. Artinya dapat membentuk suatu tampilan berdasarkan permintaan terkini. Pada prinsipnya, PHP mempunyai fungsi yang sama dengan skrip–skrip yang lain seperti ASP (*Active Server Page*), *Cold Function*, Perl dan lainnya [KAD-02:3].

PHP ini mempunyai kelebihan diantaranya adalah:

1. *Cross Platf*orm, PHP dapat dipakai dihampir semua *web server* seperti PWS (*Personal Web Server*), Apache, AOL *Server*, IIS, Xitami dan sebagainya. Dan dapat dijalankan di berbagai sistem operasi seperti Linux, FreeBSD, Solaris, Windows NT, Windows 98 dan Macintosh.
2. PHP mendukung dan mampu berkomunikasi dengan berbagai jenis basis data meskipun dengan kelengkapan yang berbeda seperti MySQL, mSQL, Oracle, Ms Access, Sybase, Solid, Generic ODBC dan Postgres SQL.
3. PHP bersifat *open source* atau gratis dan bebas pakai, selalu *up to date* sesuai dengan perkembangan.
4. PHP mudah dibuat dan kecepatan akses tinggi.
5. PHP termasuk bahasa yang *embedded* (bisa ditempel atau diletakkan dalam tag HTML).
6. PHP termasuk *server-side programming.*

Kode program PHP ditulis menyatu dengan tag - tag HTML dalam satu file. Kode PHP diawali dengan tag <? atau <?PHP dan ditutup dengan tag ?>. Contoh penulisan skrip PHP sebagai berikut (ada 3 cara) :

```
1. <?
   Script PHP
   ?>

2. <?php
   Script PHP
   ?>

3. <SCRIPT LANGUAGE="php">
   Script PHP
   </SCRIPT>
```

File yang berisi tag HTML dan kode PHP ini berisi ekstensi .php atau .phtml atau ekstensi lainnya yang dapat diterapkan pada *webserver*. Berdasar pada ekstensi ini pada saat file diakses, *web server* akan mengetahui bahwa file ini mengandung kode-kode PHP. *Web server* akan menerjemahkan kode ini dan menghasilkan keluaran dalam bentuk tag HTML yang akan dikirim ke *browser* klien yang mengakses file tersebut.

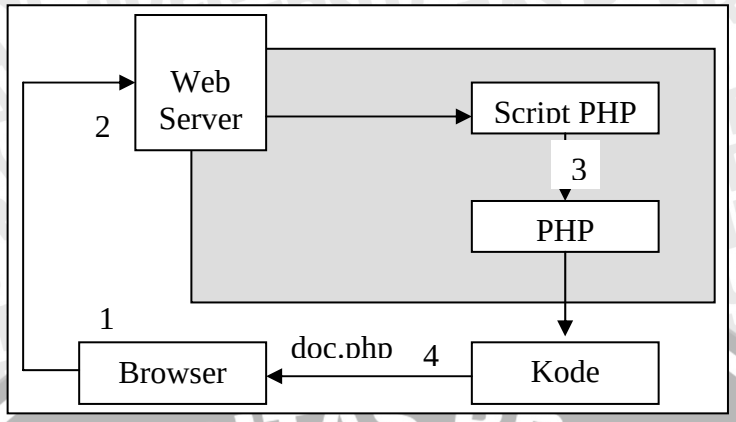
Sedangkan aturan penulisan pada PHP adalah sebagai berikut:

1. Harus dimulai dengan huruf atau *under_score* (_)
2. Tidak boleh menggunakan tanda baca
3. Variabel diawali dengan tanda dolar ($)
4. Nama fungsi dan variabel tidak boleh sama dengan nama yang telah tersedia dalam PHP.

2.9.1 Konsep Kerja PHP

PHP sebagai bahasa pemrograman yang dirancang khusus untuk pengembangan *web* memiliki kaidah - kaidah pemrograman sebagaimana ditemui bahasa pemrograman yang lain.

Pada prinsipnya kode PHP serupa dengan kode HTML, *browser* meminta berkas pada *web server*, ketika berkas PHP yang diminta didapatkan oleh *web server*, isinya segera dikirimkan ke mesin PHP dan diproses kemudian hasilnya diberikan ke *web server* untuk selanjutnya disampaikan kepada *browser* yang dapat dilihat dalam Gambar 2.9. [KAD-02:10].



Gambar 2.9 Skema PHP.

Sumber : Kadir A, 2002 :10

2.9.2 Tipe Data pada PHP

PHP mengenal tiga macam representasi data, yaitu *integer*, *floating point* dan *string*. *Integer* menyatakan tipe data bilangan bulat, *floating point* lebih dikenal dengan nama *double* yang selalu dalam bentuk desimal. *String* diawali dengan tanda petik ganda (") atau petik tunggal (').

Sedangkan variabel dalam PHP tidak harus dideklarasikan sebelum variabel tersebut digunakan. Tidak seperti dalam bahasa-bahasa tertentu yang mengharuskan untuk mendeklarasikan variabel terlebih dahulu. Konstanta adalah variabel yang nilainya tetap. Pada PHP telah didefinisikan beberapa konstanta, antara lain `PHP_VERSION`, Konstanta `TRUE` dan `FALSE`, `E_ERROR`, `E_WARNING`, `E_NOTICE`.

2.10 Apache Web Server

Kepopuleran Apache sebagai server Web atau HTTP dimulai sejak April 1996. Apache dinobatkan sebagai server Web terpopuler oleh Netcraft (www.netcraft.com) dan digunakan oleh banyak *Internet server* di seluruh dunia.

Apache adalah server Web yang dikembangkan oleh kelompok tenaga sukarela Unix dan disebarkan secara gratis berikut kode sumbernya. Kelebihan dari Apache ialah gratis, selain itu unjuk kerja dan fleksibilitasnya yang baik.

Apache dapat berjalan hampir disemua sistem operasi utama seperti Unix, Windows NT, Linux, Windows 95, dan lain-lain.

Pada pembuatan *Web server* ini, menggunakan Apache Abriasoft yang menggunakan sistem operasi pada Windows XP. Adapun Apache Abriasoft dapat di-*download software*-nya pada alamat http://www.apache.org.

2.11 Entity-Relationship Model (ER Model)

Pada *Model Entity-Relationship*, semesta data yang ada di 'Dunia Nyata' diterjemahkan/ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data, yang disebut sebagai *Diagram Entity-Relationship* (Diagram E-R). [FAT-99:46]

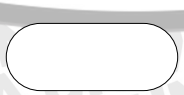
Komponen-komponen dalam *Model Entity-Relationship* adalah sebagai berikut:

1. Entitas (*Entity*) dan Himpunan Entitas (*Entity Sets*)

   Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Sedangkan sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah Himpunan Entitas (*Entity Sets*). Dalam diagram *Entity Relationship*, himpunan entitas digambarkan dengan persegi panjang.

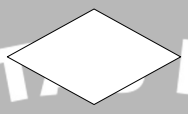
2. Atribut (*Attributes/Properties*)

   Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*properti*) dari entitas tersebut. Penetapan atribut bagi sebuah entitas umumnya memang didasarkan pada fakta yang ada. Dalam diagram *Entity Relationship*, atribut digambarkan dengan elips.



3. Relasi (*Relationship*) dan Himpunan Relasi (*Relationship Sets*)

   Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Sedangkan kumpulan semua relasi di antara entitas - entitas yang terdapat pada himpunan entitas - himpunan entitas tersebut membentuk Himpunan Relasi (*Relationship Sets*). Dalam diagram *Entity Relationship*, himpunan relasi digambarkan dengan belah ketupat.



4. Kardinalitas/Derajat Relasi

   Kardinalitas Relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

   Kardinalitas Relasi yang terjadi di antara dua himpunan entitas (misalnya A dan B) dapat berupa :

   1. Satu ke Satu (*One to One)*

      Berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B dan begitu sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

   2. Satu ke Banyak (*One to Many*)

      Berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi sebaliknya, di mana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

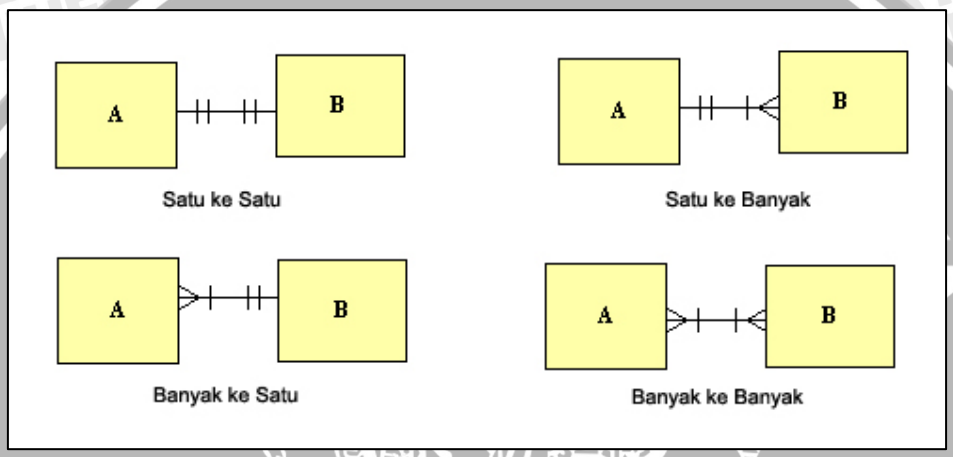
   3. Banyak ke Satu (*Many to One*)

      Berarti banyak entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap satu entitas pada himpunan entitas B berhubungan dengan banyak entitas pada himpunan entitas A.

4. Banyak ke Banyak (*Many to Many*)

   Berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan B dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

Kardinalitas Relasi yang terjadi di antara himpunan A dan B dapat dilihat dalam Gambar 2.10.



Gambar 2.10 Kardinalitas Relasi Himpunan A dan Himpunan B.
Sumber : Roger S.Pressman, 2002 : 360

Model Entity-Relationship yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari 'Dunia Nyata' yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity-Relationship*.

2.12 MySQL

MySQL merupakan *Relational Database Management System* (DBMS) yang didistribusikan secara gratis, dimana setiap orang bebas untuk menggunakan. MySQL sebenarnya merupakan turunan salah satu konsep utama *database* sejak lama yaitu SQL. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data [ANY-07].

Sebagai *database server* yang memiliki konsep *database* modern MySQL memiliki banyak sekali keistimewaan antara lain:

1. *Portability*, dimana dapat berjalan stabil pada berbagai sistem operasi.
2. *Open Source*. dimana pengembangan bersifat terbuka dan semua orang bebas berpartisipasi.
3. *Performance running*, yaitu memiliki kecepatan yang handal.
4. *Security*, yaitu memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
5. *Scalability* yaitu mampu menangani *database* dalam skala besar.

Perintah - perintah SQL dikelompokkan menjadi 5 macam yaitu:

1. *Data Definition Language* (DDL) adalah perintah SQL yang digunakan untuk menjelaskan objek dari *database*. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka *database*. Perintahnya adalah:
   - `create` : untuk membuat/menciptakan objek *database*.
   - `alter` : untuk memodifikasi/mengubah objek *database*.
   - `drop` : untuk menghapus objek *database*.
2. *Data Manipulation Language* (DML) adalah perintah yang digunakan untuk mengoperasikan atau memanipulasi isi *database*. SQL menyediakan 4 perintah DML yaitu:
   - `select` : digunakan untuk mengambil data dari *database*.
   - `delete` : digunakan untuk menghapus data pada *database*.
   - `insert` : untuk menambahkan data pada *database*.
   - `update` : untuk memodifikasi data pada *database*.
3. *Security* adalah perintah-perintah yang digunakan untuk menjamin keamanan data. Antara lain terdiri atas:
   - `grant` : memberi akses kepada *user* tertentu untuk akses ke *database*.
   - `revoke` : mencabut hak akses dari *user*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4. *Integrity* adalah perintah-perintah yang digunakan untuk menjaga kesatuan data.

   Contoh: `recover table`, untuk memperbaiki tabel pada *database*.

5. *Auxilary* adalah perintah-perintah pelengkap atau tambahan seperti: `unload` dan `rename`.

Berikut ini akan dijelaskan dalam Tabel 2.2 beberapa fungsi dari perintah-perintah yang ada dalam *database* MySQL yaitu:

Tabel 2.2 Fungsi-fungsi Perintah MySQL.
Sumber : Sutarman 2003 : 337

Perintah	Fungsi
`Mysql_connect`	Digunakan untuk membuat hubungan ke *database* MySQL yang terdapat pada suatu *host*
`Mysql_close`	Digunakan untuk menutup hubungan ke *database* MySQL
`Mysql_select_db`	Digunakan untuk memilih *database*
`Mysql_query`	Digunakan untuk mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel
`Mysql_db_query`	Digunakan untuk menjalankan suatu permintaan terhadap suatu *database*
`Mysql_num_rows`	Digunakan untuk memperoleh jumlah baris dari suatu permintaan *(query)* yang menggunakan SELECT
`Mysql_num_fields`	Digunakan untuk memperoleh jumlah kolom pada suatu hasil permintaan
`Mysql_fetch_row`	Digunakan untuk menghasilkan suatu *array* yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil
`Mysql_fetch_field`	Digunakan untuk memperoleh informasi suatu kolom

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

`Mysql_data_seek`	Digunakan untuk memindah *pointer* pada suatu himpunan hasil supaya menunjuk ke baris tertentu
`Mysql_field_seek`	Digunakan untuk memindah *pointer* pada suatu himpunan hasil supaya menunjuk ke kolom tertentu
`Mysql_create_db`	Digunakan untuk menciptakan *database* MySQL
`Mysql_drop_db`	Digunakan untuk menghapus *database* MySQL
`Mysql_list_dbs`	Digunakan untuk menghasilkan daftar *database* MySQL
`Mysql_list_tables`	Digunakan untuk menghasilkan daftar nama tabel dalam suatu *database*
`Mysql_list_fileds`	Digunakan untuk menghasilkan daftar nama kolom dalam suatu *database*
`Mysql_createdb`	membuat *database* baru di server *database* MySQL
`Mysql_fieldlen`	menghasilkan lebar *field* ke-i dari hasil
`Mysql_fieldname`	menghasilkan nama kolom *field* ke-i dari hasil
`Mysql_dbname`	menghasilkan nama dari *database* yang disimpan di *field* ke-i
`Mysql_dropdb`	menghapus *database* dan semua tabel di dalamnya

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA