

## **BAB II**

### **LANDASAN TEORI**

Pada landasan teori ini diuraikan sejumlah teori untuk membantu dan memecahkan permasalahan yang ada. Beberapa landasan teori tersebut meliputi konsep dasar dan definisi-definisi yang berkaitan dengan perangkat lunak yang digunakan serta faktor-faktor pendukung dalam pelaksanaan perancangannya. Teori-teori yang bersangkutan dengan sistem yang akan dibangun akan dijelaskan dibawah ini.

#### **1.1 Pengertian Sistem Informasi**

Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang (Davis, 1995) [1].

##### **1.1.1 Definisi**

Sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Dibawah ini merupakan beberapa pandangan mengenai sistem, yaitu :

Pengertian sistem menurut Jogiyanto (2005 : 1) Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu[2].

Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya :

1. Sistem alamiah (*Natural System*)

Sistem alamiah yaitu sistem yang terjadi melalui proses alam, tidak dibuat manusia. Seperti sistem perputaran bumi, sistem matahari, sistem luar angkasa, sistem reproduksi dan lain sebagainya.

2. Sistem buatan manusia (*Human Made System*)

Sistem buatan manusia yaitu sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*.

Informasi mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut (Abdul Kadir, 2002 : 31). Sedangkan pengertian informasi menurut Jogiyanto (2005 : 8), Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya[2].

Sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi (Budi Sutedjo, 2006 : 3)[2]. Dengan kata lain sistem informasi merupakan kesatuan elemen-elemen yang saling berinteraksi secara

sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya perusahaan.

### **1.1.2 Pemakai Sistem Informasi**

Sebagian besar sistem informasi berlandaskan komputer terdapat di dalam suatu organisasi dalam berbagai jenis. Anggota organisasi adalah pemakai informasi yang dihasilkan sistem tersebut termasuk manager yang bertanggung jawab atas pengalokasian sumber daya untuk pengembangan dan pengoperasian perusahaan.

### **1.1.3 Komponen Sistem Informasi**

Suatu system informasi terdiri dari beberapa komponen-komponen yang sering disebut blok bangunan atau building blok, yang terdiri dari beberapa komponen antara lain komponen input, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan juga komponen kontrol. Semua komponen itu saling berinteraksi dan berkaitan erat satu sama lain untuk membentuk suatu kesatuan untuk mencapai sebuah sasaran. Untuk lebih jelasnya, saya akan menjelaskan satu per satu macam-macam komponen-komponen itu.

#### **1. Komponen *Input***

Komponen ini mewakili data yang masuk kedalam suatu sistem informasi. komponen *input* disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

#### **2. Komponen *Model***

Komponen ini tersusun atas kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan sebuah keluaran yang diinginkan.

### 3. Blok *Output*

Komponen ini merupakan hasil dari sistem informasi yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai atau pengguna suatu sistem

### 4. Komponen Teknologi

Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

### 5. Komponen Basis Data

Komponen basis data atau sering disebut (*database*) ialah kumpulan atau gabungan data yang saling berkaitan dan berhubungan satu dengan yang lain dan tersimpan di perangkat keras komputer dan bisa juga menggunakan perangkat lunak untuk memanipulasinya. Sebuah data perlu disimpan dalam basis data untuk keperluan penyimpanan informasi selanjutnya. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan bisa berkualitas dan bisa membawa kebanggaan bagi si pembuatnya.

### 6. Komponen Kontrol

Dalam komponen ini banyak hal yang dapat merusak suatu system informasi ,misalnya bencana alam, *temperature*, api, air, debu, kecurangan-kecurangan, atau bahkan kegagalan-kegagalan system itu sendiri,bisa juga ketidak efisienan,sabotase dan lain-lain.beberapa pengendalian perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak suatu sistem tersebut dapat dicegah atau bahkan bila terlanjur terjadi kesalahan –kesalahan dapat langsung segera diselesaikan ataupun ditangani[2].

## **1.2 Pengembangan Perangkat Lunak**

Pengembangan perangkat lunak (*Software development*) merupakan salah satu dari tahap rancangan sistem rinci/detail dari Siklus Hidup Pengembangan Sistem (*Software Development Life Cycle* atau SDLC).

Tim proyek sistem mungkin mulai mencari paket perangkat lunak komersial yang sesuai atau mendukung spesifikasi rancangan system dan berjalan pada rancangan arsitektur komputernya. Paket perangkat lunak komersial secara luas tersedia untuk aplikasi fungsi spesifik dan aplikasi bisnis yang telah ditetapkan secara baku.

Tetapi untuk rancangan sistem yang terkait dengan kebutuhan khusus atau unik (memenuhi keperluan pemakai dan spesifikasi rancangan sistem) maka paket perangkat lunak komersial mungkin tidak sesuai atau mendukung kebutuhan pemakai secara langsung. Perangkat lunak yang diharapkan untuk mendukung rancangan sistem tersebut harus dibuat sendiri dari awal (*scratch*)[2].

### **1.2.1 Model Proses Perangkat Lunak**

Proses Pengembangan Perangkat Lunak (*Software Development Process*) adalah suatu penerapan struktur pada pengembangan suatu Perangkat Lunak (*Software*), yang bertujuan untuk

mengembangkan sistem dan memberikan panduan untuk menyelesaikan proyek pengembangan sistem melalui tahapan-tahapan tertentu. Dalam prosesnya, terdapat beberapa paradigma model pengembangan sistem perangkat lunak yang terdiri dari: model *waterfall* (model air terjun atau model sekuensial *linier*), model *prototype* (*prototyping paradigma*), model RAD (*Rapid Application Development*), model *evolusioner*, model formal, dan model teknik generasi ke empat. *Paradigm* atau model yang digunakan penulis dalam membangun perangkat lunak adalah model *waterfall* [3].

### **1.2.1.1 Model *Prototyping***

Menurut Raymond McLeod, *prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara system berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping*.

*Prototyping* adalah proses pembuatan model sederhana *software* yang mengizinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan.

Tahapan-tahapan dalam *Prototyping* adalah sebagai berikut:

#### 1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

## 2. Membangun *prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).

## 3. Evaluasi *prototyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulang langkah 1, 2, dan 3.

## 4. Mengkodekan sistem

Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

## 5. Menguji sistem

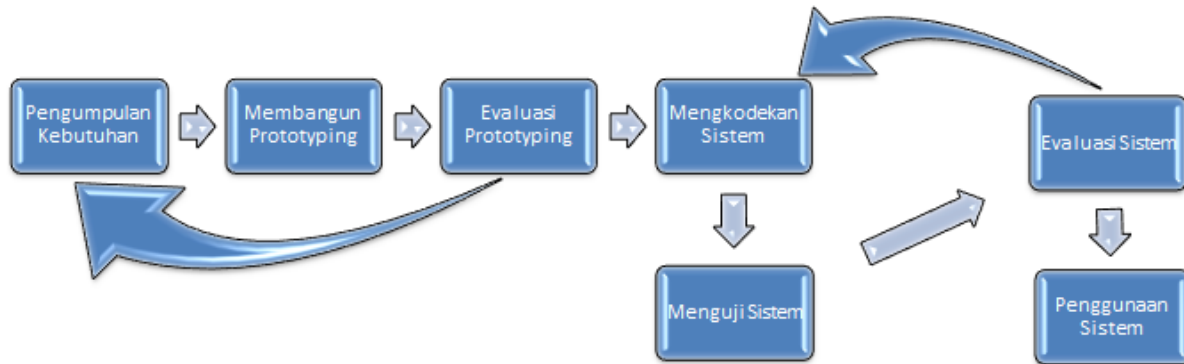
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.

## 6. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

## 7. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan[3].



**Gambar 2.1** Model Prototyping. “Sumber : Raymond McLeod (1960)”

### 1.3 Konsep Basis Data

Hampir disemua aspek pemanfaatan perangkat komputer dalam sebuah organisasi atau perusahaan senantiasa berhubungan dengan basis data. Perangkat komputer dalam suatu organisasi atau perusahaan biasanya digunakan untuk menjalankan fungsi pengolahan sistem informasi, yang dewasa ini sudah menjadi suatu keharusan demi untuk meningkatkan efisiensi, daya saing, dan kecepatan operasional perusahaan[4].

#### 1.3.1 Pengertian Basis Data

Menurut Ir. Harianto Kristanto (2004:3) Database adalah kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu.

Menurut Abdul Kadir (2006:465) database merupakan bentuk pengorganisasian data pada media *eksternal* (*disk*) dengan tujuan mempermudah pengaksesan (penyimpanan atau pengambilan) data[4].



### 1.3.2 Database Management System (DBMS)

Menurut James A. Hall, DBMS adalah sebuah sistem perangkat lunak khusus yang diprogram untuk mengetahui elemen data mana yang bisa diakses (didapatkan otorisasinya) oleh pemakai.

Menurut Connolly, *DBMS* atau *Database Management System* merupakan sebuah perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, mengambil data, dan mengontrol akses kepada *database* (Connolly, p16). *DBMS* merupakan sebuah perangkat lunak yang mengintegrasikan *database* dengan aplikasi program pada pengguna. Biasanya, DBMS menyediakan fasilitas sebagai berikut :

- a. *Data Definition Language* (DDL) memperbolehkan pengguna untuk mendeskripsikan database, misalnya merinci tipe dan batasan data yang akan disimpan dalam *database*.
- b. *Data Manipulation Language* (DML) memperbolehkan pengguna untuk memanipulasi data, misalnya memasukkan data, menghapus data, dan mendapatkan data dari *database*.
- c. Menyediakan akses terkontrol ke *database*, misalnya *security system*, *integrity system*, *concurrency control system*, *recovery control system*, *user-accessible catalog*[4].

### 1.3.3 Keunggulan DBMS

Dengan menggunakan sistem manajemen basis data penyimpanan data menjadi lebih baik selain itu juga memiliki banyak kelebihan diantaranya.

### 1. *Performance*

Dari segi performance dapat diketahui bahwa dengan menggunakan sistem basis data dapat menyimpan *file* berukuran besar, sekaligus juga membuat lebih efisien dan praktis.

### 2. *Integritas*

Dengan penggunaan DBMS integritas data menjadi lebih terjamin. Masalah redundansi sering terjadi dalam DBMS. Redundansi adalah kejadian berulangnya data atau kumpulan data yang sama dalam sebuah database yang mengakibatkan pemborosan media penyimpanan.

### 3. Independensi

Perubahan struktur database dimungkinkan terjadi tanpa harus mengubah aplikasi yang mengaksesnya sehingga pembuatan antarmuka ke dalam data akan lebih mudah dengan penggunaan DBMS.

### 4. Sentralisasi / Pusat data

Data yang terpusat akan mempermudah pengelolaan *database*. kemudahan di dalam melakukan bagi pakai dengan DBMS dan juga kekonsistenan data yang diakses secara bersama-sama akan lebih terjamin dari pada data disimpan dalam bentuk *file* atau *worksheet* yang tersebar.

### 5. Sekuritas / Keamanan Data

DBMS memiliki sistem keamanan yang lebih fleksibel daripada pengamanan pada file sistem operasi. Keamanan dalam DBMS akan memberikan keluwesan dalam pemberian hak akses kepada pengguna[4].

## **1.3.4 Komponen DBMS**

DBMS memiliki macam komponen yang digunakan untuk mengelola dan mengorganisasikan data, yaitu:

1. Perangkat keras

DBMS dan program aplikasi memerlukan perangkat keras untuk menjalankannya. Perangkat keras terdiri dari komputer pribadi, sampai ke *mainframe*, atau suatu jaringan komputer.

2. Perangkat lunak

Komponen Perangkat lunak terdiri dari perangkat lunak DBMS dan program aplikasi, bersama-sama dengan sistem operasi, mencakup perangkat lunak jaringan jika DBMS digunakan pada suatu jaringan.

3. Data

Bagi user komponen paling utama DBMS adalah data. Data bertindak sebagai suatu jembatan antara komponen mesin dan komponen manusia. *Database* berisi kedua-duanya : data yang operasional dan meta-data.

4. Prosedur

Prosedur memuat aturan-aturan untuk mendisain dan penggunaan *database*. Para pemakai sistem *database* memerlukan dokumentasi prosedur yang berisi cara menggunakan atau menjalankan sistem itu.

5. Personil

Komponen terakhir adalah personil yang terlibat didalam system.[4].

## 1.4 Pemodelan Sistem

Model analisis, yang sebenarnya merupakan serangkaian model, merupakan representasi teknis yang pertama dari sistem. Tetapi saat ini ada dua yang mendominasi landasan pemodelan analisis. Yang pertama, analisis terstruktur yaitu metode pengenalan klasik, dan analisis berorientasi objek.

Analisis terstruktur adalah aktivitas pembangunan model, dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional dapat menciptakan model yang menggambarkan muatan dan aliran informasi (data kontrol), membagi sistem secara fungsional dan secara *behavioral*, dan menggambarkan esensi dari apa yang harus dibangun.

*Entity Relationship Diagram* (ERD) adalah notasi yang digunakan untuk melakukan aktivitas pemodelan data. Atribut dari masing-masing objek data yang ditulis pada ERD dapat digambarkan dengan menggunakan deskripsi objek data.

Sedangkan *data flow diagram* (DFD) memberikan informasi tambahan yang digunakan selama analisis domain informasi dan berfungsi sebagai dasar bagi pemodelan fungsi .

### 1.4.1 *Entity Relationship Diagram* (ERD)

*Entity-Relationship* adalah salah satu metode pemodelan basisdata yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem. *Diagram Entity-Relationship* melengkapi penggambaran grafik dari struktur logika . Dengan kata lain Diagram E-R menggambarkan arti dari aspek data seperti bagaimana *entity-entity*, atribut-atribut dan *relationship-relationship* disajikan. Sebelum membuat Diagram E-R , tentunya kita harus memahami betul data yang diperlukan dan ruang lingkupnya. Di dalam pembuatan diagram E-R

perlu diperhatikan penentuan sesuatu konsep apakah merupakan suatu *entity*, atribut atau *relationship*.

Dalam rekayasa perangkat lunak, sebuah *Entity-Relationship Model* (ERM) merupakan abstrak dan konseptual representasi data. *Entity-Relationship* adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem. Dimana sistem seringkali memiliki basis data relasional, dan ketentuannya bersifat top-down. Diagram untuk menggambarkan model *Entity-Relationship* ini disebut *Entity-Relationship* diagram, ER diagram, atau ERD.

Ada pun Elemen-Element Diagram Hubungan *Entity* :

Entitas Adalah segala sesuatu yang dapat digambarkan oleh data. Entitas juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999). Ada dua macam entitas yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang tidak memiliki ketergantungan dengan entitas lainnya. Contohnya entitas anggota. Sedangkan entitas lemah merupakan entitas yang kemunculannya tergantung pada keberadaan entitas lain dalam suatu relasi.

Atribut, Atribut merupakan pendeskripsian karakteristik dari entitas. Atribut digambarkan dalam bentuk lingkaran atau *elips*. Atribut yang menjadi kunci entitas atau *key* diberi garis bawah.

Relasi atau Hubungan, Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

- Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut dinyatakan dalam bentuk garis[5].

### 1.4.2 *Data Flow Diagram (DFD)*

*Data Flow Diagram* (Diagram Arus Data) atau DFD adalah suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk- bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang saling berkaitan (McLeod Jr., p316). Walaupun nama diagram ini menekankan pada data, situasinya justru sebaliknya : penekanannya ada pada proses. DFD mungkin cara paling alamiah untuk mendokumentasikan proses DFD terdiri dari empat simbol. Simbol-simbol itu digunakan untuk (McLeod Jr., p316) :

#### 1. Elemen-elemen lingkungan yang berhubungan dengan sistem

Elemen-elemen lingkungan berada di luar batas sistem. Elemen- elemen ini menyediakan bagi sistem input data dan menerima output data dari sistem. Pada DFD, tidak dibuat perbedaan antara data dan informasi, semua arus dipandang sebagai data.

Nama terminator digunakan untuk menggambarkan elemen- elemen lingkungan, yang menandai titik-titik berakhirnya sistem. Suatu terminator dapat berupa orang, organisasi, maupun sistem lain. Terminator digambarkan sebagai suatu kotak atau segi empat dan tiap simbol terminator diberi label nama elemen lingkungan.

#### 2. Proses

Proses adalah sesuatu yang mengubah input menjadi output. Tiap simbol proses diidentifikasi oleh label. Proses dapat digambarkan dengan lingkaran, segi empat horisontal, atau segi empat tegak dengan sudut-sudut yang membulat.

#### 3. Arus Data

Arus data terdiri dari sekelompok elemen data yang berhubungan secara logis yang bergerak dari satu titik atau proses ke titik atau proses yang lain. Arus data terdiri

dari satu atau beberapa struktur data. Struktur data adalah sekelompok elemen data yang menggambarkan suatu hal atau transaksi tertentu. Arus data dapat bercabang (*diverge*) ketika data yang sama bergerak ke beberapa lokasi dalam sistem, dapat pula memusat (*converge*) untuk menggambarkan beberapa arus data yang sama bergerak ke satu lokasi. Arus data dapat digambarkan dengan tanda panah, panah tersebut dapat digambar sebagai garis lurus atau lengkung.

#### 4. Penyimpanan data

Jika data perlu dipertahankan karena suatu sebab, maka digunakan penyimpanan data, dalam istilah DFD, penyimpanan data (*data storage*) adalah suatu tempat penampungan data. Penyimpanan data dapat digambarkan sebagai satu set garis paralel, segi empat terbuka, atau bentuk lonjong [6].

### 1.4.3 Diagram Konteks

Menurut [Jog'05] diagram konteks adalah model atau gambar yang menggambarkan hubungan sistem dengan lingkungan sistem. Untuk menggambarkan diagram konteks kita deskripsikan data apa saja yang dibutuhkan oleh sistem dan dari mana sumbernya, serta informasi apa saja yang akan dihasilkan oleh sistem tersebut dan kemana informasi tersebut akan diberikan. Diagram konteks adalah kasus dari DFD atau bagian dari DFD yang berfungsi memetakan modul lingkungan yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan system [6].

### 1.4.4 Kamus Data

Kamus Menurut [Ken'03] Suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari, kamus data merupakan hasil referensi data mengenai data (maksudnya metadata) suatu data yang tersusun oleh penganalisis sistem untuk membimbing mereka selama melakukan analisis dan desain. sebagai suatu dokumen, kamus data mengumpulkan dan mengkoordinasikan istilah-istilah data tertentu, dan menjelaskan apa arti setiap istilah yang ada.

Sebagai tambahan untuk dokumentasi serta mengurangi redundansi. Kamus data bisa digunakan untuk:

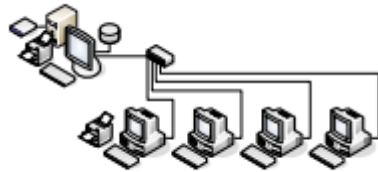
- a) Menvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan
- b) Menyediakan suatu titik awal untuk mengembangkan layar dan laporan laporan
- c) Menentukan muatan data disimpan dalam file-file
- d) Mengembangkan logika untuk proses-proses diagram aliran data [6]

#### ***1.4.5 Client Server***

Dalam Arsitektur jaringan *client server* merupakan pengembangan dari arsitektur file server. Arsitektur ini adalah model konektivitas pada jaringan yang mengenal adanya *server* dan *client*, dimana masing-masing memiliki fungsi yang berbeda satu sama lain. *Server* dapat berbagi pakai data, aplikasi dan peripheral seperti *harddisk*, *printer*, modem dan lain lain. Oleh karena itu, tidak jarang juga tercipta sebutan *print server*, *communication server* dan lain sebagainya. Prinsip kerjanya sangat sederhana, dimana server akan menunggu permintaan dari *client*, memproses dan



memberikan hasilnya kepada *client*. Sedangkan *client* akan mengirimkan permintaan ke *server*, menunggu proses dan melihat visualisasi hasil prosesnya[7].



**Gambar 2.2** Model Konektivitas Client Server.

“Sumber : google Image”

#### 1.4.6 PHP (*Personal Home Page*)

PHP adalah salah satu bahasa *server-side*. PHP dahulunya merupakan proyek pribadi dari Rasmus Lerdorf (dengan dikeluarkannya php versi 1) yang digunakan untuk membuat home page pribadinya. Versi pertama ini berupa kumpulan *script* PERL. Untuk versi keduanya, Rasmus menulis ulang *script-script* PERL tersebut menggunakan bahasa C, kemudian menambahkan fasilitas untuk form html dan koneksi MYSQL. Adapun PHP didapat dari singkatan *Personal Home Pages*.

PHP merupakan bahasa *script* yang digunakan untuk membuat aplikasi yang dinamis. Dinamis berarti halaman yang ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan. Oleh karena itu, spesifikasi *server* lebih berpengaruh pada eksekusi dari *script* PHP daripada spesifikasi *client*. Namun tetap diperhatikan

bahwa halaman aplikasi yang dihasilkan tentunya harus dapat dibuka oleh browser pada *client*. PHP masuk kedalam kategori *server-side scripting* dimana *browser* pada *client* tidak lagi bertanggung jawab dalam menjalankan kode-kode PHP[6] .

#### 1.4.7 MySQL

Dalam buku “Manajemen Database dengan My SQL, Arbie (2004)”, MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management sistem*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti *Apache*, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan

pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. *Open Source*

MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.

## 6. *Command dan functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

## 7. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

## 8. *Scalability dan limits*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

## 9. *Connectivity*

MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).

## 10. *Localisation*

MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

## 11. *Interface*

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

## 12. *Clients dan tools*

MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

### 13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan database lainnya semacam *PostgreSQL* ataupun *Oracle* [6].