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.

2.1 Pengertian Sistem Informasi

Sebelum membahas tentang konsep dasar sistem informasi terlebih dahulu kita harus menjelaskan sekilas tentang pengertian sistem dan informasi.

2.1.1 Definisi

Banyak definisi yang benar tentang sistem dan informasi mulai dari yang paling abstrak sampai ke yang paling kongkrit, tetapi definisi peraktisnya yang berguna menurut “Gordon B. Davis” mengenai sistem yaitu: Suatu kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu [3].

Sedangkan Jogiyanto H.M dalam bukunya Pengenalan Komputer mengatakan bahwa:

Sistem adalah kumpulan dari sub-sub sistem atau komponen-komponen yang saling berinteraksi, saling bekerjasama dalam satu kesatuan kerja untuk mencapai satu tujuan tertentu [6].

Menurut “Gordon B. Davis” informasi dapat di definisikan sebagai berikut: Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatan [3].

Sedangkan Jogiyanto H.M dalam bukunya Pengenalan Komputer berpendapat bahwa :

Informasi yaitu hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (events) yang nyata (fact) yang digunakan untuk pengambilan keputusan [6].

Informasi dapat diperoleh dari sistem informasi (*information system*), sistem informasi dapat di definisikan sebagai berikut: Sebuah sistem informasi adalah suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media. Prosedur-prosedur dan pengendalian yang ditunjukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan external yang penting dan menyediakan suatu dasar informasi untuk mengambil keputusan yang cerdik [3].

2.1.2 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 manajer yang bertanggung

jawab atas pengalokasian sumber daya untuk pengembangan dan pengoperasian perusahaan.

2.1.3 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang di sebutkan dengan istilah blok bangun (*building block*), komponen-komponen blok bangun yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data(*database block*), dan blok kendali (*controls block*) [6].

1. **Blok Masukan**

   Input mewakili data yang masuk kedalam sistem informasi. *Input* disini termasik metode-metode dan media untuk menangkap data yang akan dimasukan, yang berupa dokumen-dokumen dasar.

2. **Blok Model**

   Blok ini merupakan kombinasi prosedur, logika dan model matematika yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

3. **Blok Keluaran**

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumen tasi yang berguna untuk semua tingkatan menajemen serta semua pemakai sistem.

4. Blok 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. Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data-data yang berhubungan antara satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management System*).

6. Blok kendali

Beberapa pengendalian perlu dirancang dan ditetapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung diatasi.

2.2 Pengembangan Perangkat Lunak

Untuk menyelesaikan masalah aktual didalam sebuah seting industri rekayasa perangkat lunak atau tim perekayasa harus menggabungkan strategi pengembangan yang melingkupi lapisan peroses, metode, dan alat-alat bantu serta fase-fase generik. Strategi ini sering diacukan sebagai model proses atau paradigma rekayasa perangkat lunak. Evolusi dari perangkat lunak terdiri dari:

1. Pembangunan Perangkat Lunak

   Yang termasuk dalam pembangunan perangkat lunak itu sendiri meliputi dari daur hidup perangkat lunak, yaitu: permintaan (*requirement*), spesipikasi, perancangan (*design*). Testing dan sebagainya.

2. Perawatan Perangkat Lunak

   Merupakan proses-proses untuk memperpanjang waktu pengguna sistem perangkat lunak yang ada sehingga tetap dapat dipergunakan sebagaimana mestinya sesua dengan baik.

3. Migrasi Perangkat Lunak

   Merupakan proses-proses yang memindahkan sistem yang ada kesistem yang baru dikarenakan perkembangan dari kebutuhan perangkat lunak tersebut [8].

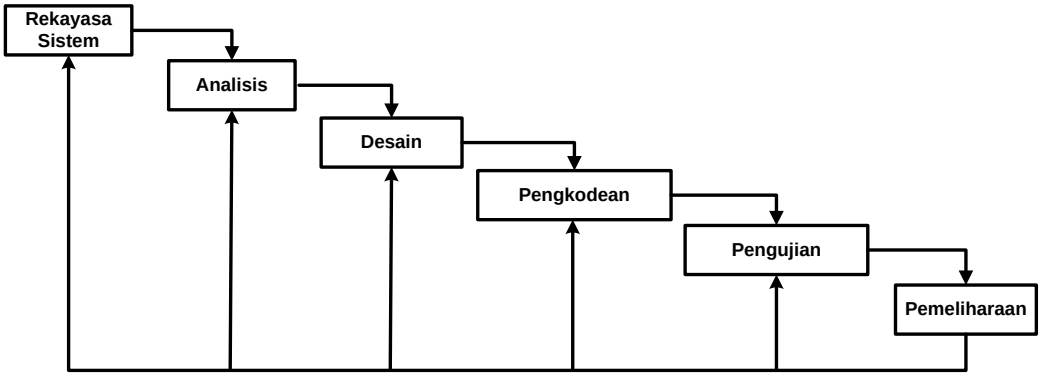
2.2.1 Model Proses Perangkat Lunak

Model proses perangkat lunak merupakan representasi abstrak dari proses perangkat lunak. Seting model proses merepresentasikan suatu proses dari sudut

pandang tertentu sehingga hanya memberikan informasi parsial mengenai proses tersebut. Model-model proses perangkat lunak terdiri dari: model *waterfall* (model air terjun atau model sekuensial linier), model prototype (*prototyping paradigma*), model RAD (*Rapid Aplication Development*), model *evolusioner*, model formal, dan model teknik generasi ke empat. Paradigm atau model yang digunakan penulis dalam membangun perangkat lunak adalah model *waterfall* [8].

2.2.1.1 Model *Waterfall* (Model Air Terjun atau Model Sekuensial Linier)

Menurut S. Pressman Roger, Ph.D. (1999) Model *waterfall* adalah paradigm rekayasa perangkat lunak yang paling luas dipakai dan paling tua. Model *waterfall* mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Setelah setiap tahap didefinisikan, tahap tersebut diakhiri (*signed off*) dan pengembangan berlanjut ke tahap berikutnya [8].



Gambar 2.1 Model *Waterfall.* "Sumber : S. Pressman Roger, Ph.D. (1999)"

Sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan

kemajuan sistem pada seluruh analisis, perancangan, pengkodean, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut (Pressman, Rekayasa Perangkat Lunak) [8]:

1. Rekayasa dan pemodelan sistem/informasi

   Karena perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain seperti perangkat lunak, manusia, dan database. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan perangkat lunak

   Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perekayasa perangkat lunak (analisis) muka (interface) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

   Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi interface. Proses desain menerjemahkan syarat/kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

4. *Generate Code*

   Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian

   Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada external fungsional yaitu mengarahkan pengujian untuk enemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

   Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah perangkat lunak yang

dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasikan perubahan-perubahan di dalam lingkungan eksternalnya, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja.

2.3 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.

2.3.1 Pengertian Basis Data

Basis data atau *database* dapat didefinisikan sebagai : kumpulan dari data-data yang berhubungan antara satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai [4].

Tujuan dari desain basis data adalah untuk menentukan data-data yang dibutuhkan dalam sistem, sehingga informasi yang dihasilkan dapat terpenuhi dengan baik. Perancangan *database* yang digunakan adalah untuk memudahkan

dalam mengetahui *file-file database* yang digunakan dalam perancangan sistem, sekaligus untuk mengetahui hubungan antara file dari *database* tersebut.

Ada beberapa istilah atau definisi yang digunakan dalam sistem manajemen basis data, antara lain :

a. Entitas (*entity*) : Orang, tempat, kejadian (konsep) yang informasinya direkam.

b. Atribut : Setiap entitas mempunyai atribut atau sebutan untuk mewakili suatu entitas.

c. *Data Value* (Nilai / isi data) : Data aktual informasi yang disimpan pada tiap data.

d. *Record / Tuple :* Kumpulan elemen-elemen yang saling berkaitan, yang menginformasikan suatu entitas secara lengkap.

e. *File :* Kumpulan *record* yang mempunyai panjang elemen yang sama, atribut sama, namun berbeda isi datanya.

f. *Database :* Kumpulan *file-file* yang mempunyai kaitan antara *file-file* dengan *file* lainnya sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan atau instansi dalam batasan tertentu.

g. *Database Management Sistem* (DBMS) : Kumpulan *file* yang saling berkaitan bersama dengan program untuk pengelolaannya. Database adalah kumpulan datanya, sedangkan program pengelolanya berdiri sendiri

dalam sutu paket program komersial untuk membaca data, mengisi data, menghapus data, melaporkan data, dalam database.

2.3.2 *Database Management System* (DBMS)

Definisi DBMS pada sejumlah literartur sangatlah bervariasi. Secara umum, DBMS diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi, dan memperoleh data atau informasi dengan praktis dan efisien. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam field dan menulis kode aplikasi yang spesifik untuk pengaturannya [4].

Diperlukan suatu sistem untuk diintegrasikan data file kedalam suatu file sehingga bisa melayani berbagai user yang berbeda. Perangkat keras dan lunak serta prosedur yang mengelola database merupakan suatu database manajemen sistem. DBMS memungkinkan untuk membentuk dan meremajakan file-file, memilih, mendatakan dan menyortir data, dan untuk menghasilkan laporan-laporan. Fungsi yang penting dari DBMS adalah sebagai berikut :

1. Menyediakan sistem access cepat.
2. Mengurangi kerangkapan data atau *redudancy* data.
3. Memungkinkan adanya updating secara bersama.
4. Menyediakan sistem yang memungkinkan dilakukannya pengembangan terhadap *database.*
5. Memberikan perlindungan dari pihak pemakai yang tidak berhak.

2.3.3 Keunggulan DBMS

Dibandingkan dengan sistem yang berbasis kertas, DBMS memiliki 4 keunggulan :

1. Kepraktisan : Sistem yang berbasis kertas akan menggunakan kertas yang sangat banyak untuk menyimpan informasi, sedangkan DBMS menggunakan media penyimpan sekunder yang berukuran kecil tetapi padat informasi.
2. Kecepatan : Mesin dapat mengambil atau mengubah data jauh lebih cepat daripada manusia.
3. Mengurangi kejemuan : Orang cenderung menjadi bosan kalau melakukan tindakan-tindakan berulang yang menggunakan tangan (misalnya harus mengganti suatu informasi).
4. Kekinian : Informasi yang tersedia pada DBMS akan bersifat mutakhir dan akurat setiap saat.

2.3.4 Komponen DBMS:

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

1. File Manager, yang mengelola ruang dalam disk dan struktur data yang dipakai untuk merepresentasikan informasi yang tersimpan dalam disk.
2. Database Manager, yang menyediakan interface antara data low-level yang ada di basis data dengan program aplikasi dan query yang diberikan ke sistem.

3. Query Processor, yang menterjemahkan perintah-perintah dalam query language ke perintah low-level yang dapat dimengerti oleh database manager.
4. DML Precompiler, yang mengkonversi perintah DML yang ditambahkan dalam sebuah program aplikasi ke prosedur normal dalam bahasa induk.
5. DDL Compiler, yang mengkonversi perintah-perintah DDL ke dalam sekumpulan tabel yang mengandung metadata. Tabel-tabel ini kemudian disimpan dalam kamus data.

2.4 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].

2.4.1 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah suatu model diagram yang menyatakan keterhubungan suatu *entity* dengan *entity* yang lain. Atau juga dapat dikatakan sebagai sebuah teknik untuk menggambarkan informasi yang dibutuhkan dalam sistem dan hubungan antar data-data tersebut.

Secara terjemahan dalam bahasa Indonesia, *Entity Relationship Diagram* adalah diagram relasi atau keterhubungan entitas. Dari model *Entity Relationship Diagram* akan didapatkan data-data yang dibutuhkan sistem. Dengan begitu maka akan didapatkan pula kejelasan aktivitas yang dilakukan dalam sistem [1].

Didalam *Entity Relationship Diagram* (ERD) dikenal beberapa komponen, yaitu sebagai berikut :

1. Entitas

   Entitas adalah suatu objek yang memiliki hubungan dengan objek lain. Didalam ERD digambarkan dengan bentuk persegi panjang.

2. Hubungan

   Dimana entitas dapat berhubungan dengan entitas lain, hubungan ini disebut dengan *entity relationship* yang digambarkan dengan garis.

   Ada empat bentuk relasi dasar pada *database*, yaitu :

   a. *One-to-One* artinya satu data memiliki satu data pasangan.

b. *One-to Many* artinya satu data memiliki beberapa data pasangan.

c. *Many-to-One* artinya beberapa data memiliki satu data pasangan.

d. *Many-to-Many* artinya beberapa data memiliki beberapa data pasangan.

3. Atribut

Atribut adalah elemen dari entitas yang berfungsi sebagai deskripsi karakter entitas dan digambarkan dengan bentuk *elips*.

2.4.2 *Data Flow Diagram* (DFD)

Data Flow Diagram adalah suatu jaringan dari proses dengan tempat penyimpanan data serta dihubungkan satu dengan lainnya, atau kumpulan simbol-simbol yang menggambarkan jalannya aliran data dari sistem atau suatu diagram yang mudah dimengerti dan merupakan suatu gambaran mengenai tata letak lokasi dan semua kegiatan-kegiatan dan aktivitas yang terjadi di dalam proses aktivitas tersebut [1].

Beberapa simbol yang di pergunakan dalam *Data Flow Diagram* (DFD) antara lain:

1. *External entity* (kesatuan luar) atau *boundary* (batas sistem)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat

berupa orang, organisasi, atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. *Data Flow* (arus data)

   Data Flow (arus data) di DFD diberi simbol panah. Arus data ini mengalir diantara proses, simpanan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. *Process* (proses)

   Suatu proses adalah suatu kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk menghasilkan arus data yang akan keluar dari proses. Setiap proses harus diberi penjelasan yang lengkap meliputi identifikasi proses, nama proses dan pemroses.

4. *Data Store* (simpanan data)

   Data Store (simpanan data) merupakan simpanan dari data yang dapat berupa, suatu file atau database di sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data dimeja seseorang, suatu tabel acuan manual, dan suatu agenda atau buku.

2.4.3 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari

sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada store dalam diagram konteks.

Diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi "siapa saja yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem." Jadi, yang dibutuhkan adalah Siapa saja pihak yang akan memberikan data ke sistem, data apa saja yang diberikannya ke sistem, kepada siapa sistem harus memberi informasi atau laporan, dan apa saja isi atau jenis laporan yang harus dihasilkan sistem. Kata "Siapa" di atas dilambangkan dengan kotak persegi (disebut dengan *terminator*), dan kata "apa" di atas dilambangkan dengan aliran data (disebut dengan *data flow*), dan kata "sistem" dilambangkan dengan lingkaran (disebut dengan *process*) [1].

2.4.4 Kamus Data

Kamus data adalah katalog fakta tentang data kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan kamus data analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap analisis sistem, kamus data digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir ke sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input*, merancang laporan dan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD.

Kamus data tidak menggunakan notasi grafik sebagaimana halnya DFD. Kamus data berfungsi membantu pelaku sistem untuk mengerti aplikasi secara presisi, sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses [1].

Cara mendefinisikan kamus data adalah:

1. Menggambarkan arti aliran data atau penyimpanan yang ditunjukan dalam *Diagram Flow Datagram.*

2. Menggambarkan komponen dari kumpulan data yang mengalir, yaitu kumpulan komponen yang mengkin bisa dipecahkan menjadi data elementer.

3. Menggambarkan data yang tersimpan.

4. Menentukan nilai di bagian elementer dari informasi yang relevan *Diagram Flow Datagram* dan *file*.

2.5 *Client Server*

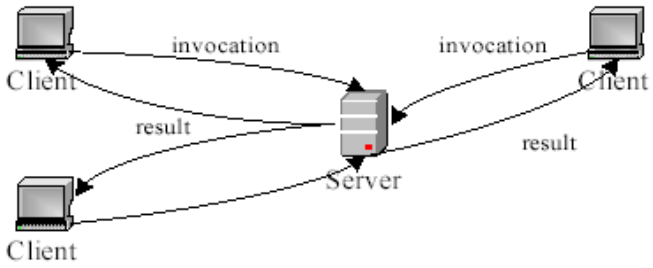
Dalam jaringan komputer terdapat dua tipe jaringan, yaitu *model Peer to Peer* dan *model client-server.* Tipe jaringan yang diterapkan dalam sistem ini adalah model jaringan *client-server.*

Dalam buku “Jaringan Komputer, Irawan, Budhi (2005)”, sistem *client - server* mempunyai dua komponen utama yaitu komputer *client* dan komputer *server*. *Server* merupakan komputer induk yang melakukan pemrosesan terbanyak

untuk memenuhi permintaan- permintaan dari komputer *client* dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu komputer atau *workstation* yang melakukan pengiriman permintaan- permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya. Selain itu *client* juga mempunyai kemampuan untuk mengubah atau menghapus data itu.

Sistem *client* - *server* merupakan suatu sistem *client* komputer yang melibatkan proses - proses *client* yang meminta suatu pelayanan data kepada komputer *server* yang menyediakan layanan data tersebut. Sehingga *client* maupun *server* sama - sama melakukan pekerjaan. Dengan adanya kombinasi *client (Frontend)* dan *server (back-end)* ini maka kumpulan dari modul - modul program tidak dieksekusi dalam memori yang sama namun terbagi dalam komputer *client - server.*

Hal ini menjadikan konfigurasi bagi komputer *client* dan komputer *server* bias berbeda seperti kapasitas memori, kecepatan prosesor atau alat masukan dan keluaran yang disesuaikan dengan fungsi kerja dari elemen - elemen tersebut. Bagi *server* yang menjalankan tugas pengelolaan suatu *database* digunakan suatu konfigurasi yang khusus menangani tugasnya tersebut dengan sistem operasi yang dikhususkan bagi *server* seperti windows NT *server, windows 2000 server,* sedangkan komputer *client* menggunakan konfigurasi yang umum bagi sebuah komputer desktop yang terhubung ke jaringan dengan sistem operasi seperti *windows 98, windows ME, windows Xp* dan lain - lain [5].



Gambar 2.2 *Model Client-server*. "Sumber : Irawan, Budhi (2005)"

2.6 PHP (*Personal Home Page*)

PHP adalah salah satu bahasa *sever-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 menngunakan 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 [7].

2.7 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 soket (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* [2].