

## II TINJAUAN PUSTAKA

### A. Bahasa Provinsi Lampung

Aksara Lampung yang disebut dengan Had Lampung adalah bentuk tulisan yang memiliki hubungan dengan aksara Pallawa dari India Selatan. Had Lampung diciptakan oleh Para Saibatin di Paksi Pak Sekala Brak pada awal abad ke-9. Macam tulisannya fonetik berjenis suku kata yang merupakan huruf hidup seperti dalam Huruf Arab dengan menggunakan tanda tanda fathah di baris atas dan tanda tanda kasrah di baris bawah tapi tidak menggunakan tanda dammah di baris depan melainkan menggunakan tanda di belakang, masing-masing tanda mempunyai nama tersendiri.

Artinya Had Lampung dipengaruhi dua unsur yaitu Aksara Pallawa dan Huruf Arab. Had Lampung memiliki bentuk kekerabatan dengan Aksara Rencong Aceh, Aksara Rejang Bengkulu dan Aksara Bugis. Had Lampung terdiri atas huruf induk, anak huruf, anak huruf ganda dan gugus konsonan, juga terdapat lambing, angka dan tanda baca. Had Lampung disebut dengan istilah KaGaNga ditulis dan dibaca dari kiri ke kanan dengan Huruf Induk berjumlah 20 buah.

Dr Van Royen mengklasifikasikan Bahasa Lampung dalam Dua Sub Dialek yaitu Dialek Belalau atau Dialek Api, yang dipertuturkan oleh sebagian besar Etnis Lampung yang masih memegang teguh Garis Adat dan Aturan Saibatin dan Dialek Nyow, yang dipertuturkan oleh orang Abung dan Tulang Bawang yang mengenal kenaikan Pangkat Adat dengan Kompensasi Tertentu yang berkembang setelah Seba yang dilakukan oleh Orang Abung ke Banten.

- a. Dialek Belalau (Dialek Api / Dialek A), terbagi menjadi:
  1. Bahasa Lampung Logat Belalau dengan tambahan spesifikasi Logat Kembahang dan Logat Sukau, Dipertuturkan oleh Etnis Lampung yang berdomisili di Kabupaten Lampung Barat yaitu Kecamatan Balik Bukit, Batu Brak, Belalau, Suoh, Sukau, Ranau, Sekincau, Gedung Surian, Way Tenong dan Sumber Jaya. Kabupaten Lampung Selatan di Kecamatan Kalianda, Penengahan, Palas, Pedada, Katibung, Way Lima, Padangcermin, Kedondong dan Gedongtataan. Kabupaten Tanggamus di Kecamatan Kotaagung, Semaka, Talangpadang, Pagelaran, Pardasuka, Hulu Semuong, Cukuhbalak dan Pulau Panggung. Penututr dialek ini di kota Bandar Lampung berada di Teluk Betung Barat, Teluk Betung Selatan, Teluk Betung Utara, Panjang, Kemiling dan Raja Basa. Penutur yang sama di Banten berada di Cikoneng, Bojong, Salatuhur dan Tegal dalam Kecamatan Anyer, Serang.
  2. Bahasa Lampung Logat Krui dipertuturkan oleh Etnis Lampung di Pesisir Barat Lampung Barat yaitu Kecamatan Pesisir Tengah,

Pesisir Utara, Pesisir Selatan, Karya Penggawa, Lemong, Bengkuntan dan Ngaras.

3. Bahasa Lampung Logat Melinting dipertuturkan Masyarakat Etnis Lampung yang bertempat tinggal di Kabupaten Lampung Timur di Kecamatan Labuhan Maringgai, Kecamatan Jabung dan Kecamatan Way Jepara.
  4. Bahasa Lampung Logat Way Kanan dipertuturkan Masyarakat Etnis Lampung yang bertempat tinggal di Kabupaten Way Kanan yakni di Kecamatan Blambangan Umpu, Baradatu, Bahuga dan Pakuan Ratu.
  5. Bahasa Lampung Logat Pubian dipertuturkan oleh Etnis Lampung yang berdomosili di Kabupaten Lampung Selatan yaitu di Natar, Gedung Tataan dan Tegineneng. Penutur di Lampung Tengah berada di Kecamatan Pubian dan Kecamatan Padangratu. Penutur di kota Bandar Lampung berada di Kecamatan Kedaton, Sukarame dan Tanjung Karang Barat.
  6. Bahasa Lampung Logat Sungkay dipertuturkan Etnis Lampung yang berdomisili di Kabupaten Lampung Utara meliputi Kecamatan Sungkay Selatan, Sungkai Utara dan Sungkay Jaya.
  7. Bahasa Lampung Logat Jelema Daya atau Logat Komring dipertuturkan oleh Masyarakat Etnis Lampung yang berada di Muara Dua, Martapura, Komring, Tanjung Raja dan Kayuagung di Propinsi Sumatera Selatan.
- b. Dialek Abung (Dialek Nyow / Dialek O), terbagi menjadi:

1. Bahasa Lampung Logat Abung Dipertuturkan Etnis Lampung yang yang berdomisili di Kabupaten Lampung Utara meliputi Kecamatan Kotabumi, Abung Barat, Abung Timur dan Abung Selatan. Lampung Tengah di Kecamatan Gunung Sugih, Punggur, Terbanggi Besar, Seputih Raman, Seputih Banyak, Seputih Mataram dan Rumbia. Penutur di Lampung Timur berada di Kecamatan Sukadana, Metro Kibang, Batanghari, Sekampung dan Way Jepara. Penutur di Kota Metro berada di Kecamatan Metro Raya dan Bantul. Kota Bandar Lampung di Gedongmeneng dan Labuhan Ratu.
2. Bahasa Lampung Logat Menggala Dipertuturkan Masyarakat Etnis Lampung yang bertempat tinggal di Kabupaten Tulang Bawang meliputi Kecamatan Menggala, Tulang Bawang Udik, Tulang Bawang Tengah, Gunung Terang dan Gedung Aji.[2]

## **B. Sistem**

### **1. Pengertian sistem**

Sistem adalah sehimpunan unsur yang melakukan suatu kegiatan atau menyusun skema atau melaksanakan suatu skema atau melakukan suatu kegiatan pemrosesan untuk mencapai suatu atau beberapa tujuan, dalam hal ini dilakukan dengan cara mengolah data dan energi atau barang dan benda dalam jangka tertentu guna menghasilkan informasi atau energi atau benda atau barang. Sedangkan menurut Jogianto (2001), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama

untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

Pada dasarnya ada dua jenis sistem didunia ini yaitu :

- a. Sistem alami yaitu sistem yang adanya secara alami, misalnya sistem matahari, sistem tatasurya, sistem reproduksi, sistem tubuh dan lain-lain.
- b. Sistem buatan manusia yaitu sistem yang adanya karena dibuat oleh manusia. Misalnya sistem hukum, sistem perpustakaan, sistem transportasi, dan lain-lain.

## **2. Pengertian Sistem Informasi**

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 serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2001).

### **a. Komponen Sistem Informasi**

Komponen-komponen sistem informasi adalah sebagai berikut :

#### **1. Blok Masukan**

Blok masukan terdiri atas metode-metode dan media untuk menangkap data yang dimasukkan, yang dapat berupa dokumen-dokumen dasar.

## 2. Blok Model

Blok model terdiri atas kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan dalam basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

## 3. Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumen yang berguna untuk tingkat manajemen dan semua pemakai sistem.

## 4. Blok Teknologi

Teknologi merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, serta membantu pengendalian sistem secara keseluruhan.

## 5. Blok Basis Data

Basis data merupakan kumpulan data yang saling berhubungan antara yang satu dengan yang lain. Basis data tersimpan dalam perangkat keras komputer dan untuk menggunakannya diperlukan perangkat lunak untuk memanipulasi.

## 6. Blok Kendali

Blok kendali berguna untuk mengendalikan sistem informasi agar dapat berjalan dengan baik sesuai dengan keinginan.[6]

## b. Perangkat Pemodelan Sistem

Langkah- langkah yang dilakukan dalam menganalisa sistem adalah membuat pemecahan masalah secara logika.

Alat bantu yang digunakan adalah :

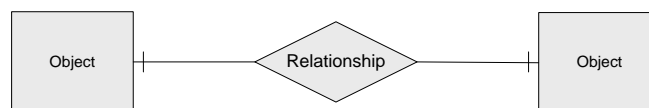
1. *Entity Relationship Diagram (ERD)*
2. *Data Context Diagram (DCD)*
3. *Data flow diagram (DFD)*

### 1. *Entitiy Relationship Diagram (ERD)*

ERD merepresentasikan pemodelan data di mana pemodelan ini diperlukan untuk meminimalkan kebergantungan objek data terhadap proses, memfokuskan pada pengekplorasian domain data, membuat suatu model yang memudahkan *customer* memahaminya dan mengindikasikan keterhubungan antara suatu objek data dengan objek data lainnya.[7]

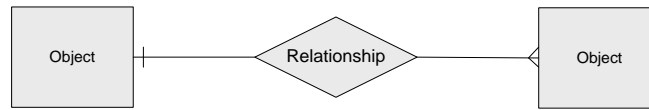
Notasi ERD digambarkan sebagai berikut:

#### a. *One to one*



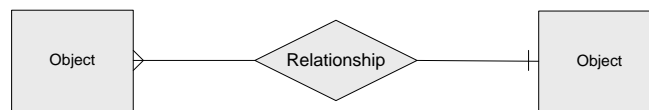
Gambar 1. ERD *One to one*

b. *One to many*



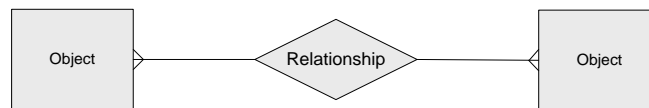
Gambar 2. ERD *One to many*

c. *Many to one*



Gambar 3. ERD *Many to one*

d. *Many to many*



Gambar 4. ERD *Many to many*

Untuk kasus di mana data yang terlibat dalam sistem tidak memerlukan penyimpanan (basis data) atau jika objek-objek data tersebut tidak memiliki keterhubungan satu sama lain, ERD tidak perlu digambarkan.

## 2. *Data Context Diagram (DCD)*

Diagram konteks merupakan diagram level pertama yang memperlihatkan sistem sebagai suatu proses yang berinteraksi dengan lingkungannya. Ada pihak luar yang memasukkan informasi ke dalam sistem dan ada yang

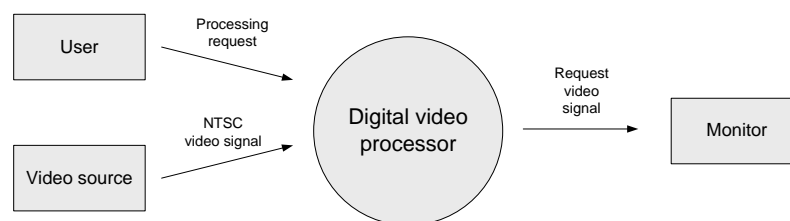


menerima informasi dari sistem. Pihak luar bisa berupa sistem lain, perangkat keras, orang atau organisasi.[8]

Pembuatan diagram konteks dilakukan dengan tahapan sebagai berikut :

- a. Menentukan entitas eksternal.
- b. Menentukan informasi yang mengalir dari entitas luar ke sistem dan sebaliknya.
- c. Menggambarkan diagram konteks.

Gambar di bawah memperlihatkan contoh diagram konteks

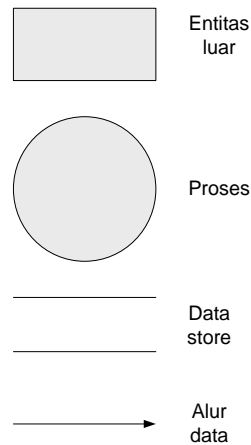


Gambar 5. Contoh DCD

### 3. Data Flow Diagram (DFD)

DFD merepresentasikan pemodelan fungsi di mana pemodelan ini diperlukan untuk memperlihatkan proses-proses yang dimiliki aplikasi dan bagaimana proses tersebut mentransformasikan data menjadi informasi. DFD digunakan untuk menggambarkan aliran data yang mengalir dalam sistem atau perangkat lunak tersebut.[8]

Notasi DFD dapat dilihat dari gambar di bawah ini



Gambar 6. Notasi DFD

Entitas luar adalah entitas di luar sistem yang berkomunikasi atau berhubungan langsung dengan sistem. Entitas luar dapat berupa orang, sekelompok orang, organisasi, perusahaan atau departemen yang berada di luar sistem yang akan dibuat, diberi nama yang berhubungan dengan sistem tersebut dan biasanya menggunakan kata benda.

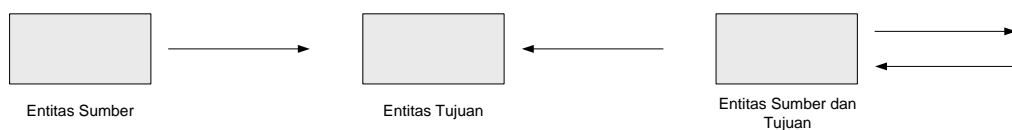
Terdapat dua jenis entitas luar:

#### 1. Entitas luar sumber

Merupakan entitas luar yang menjadi sumber.

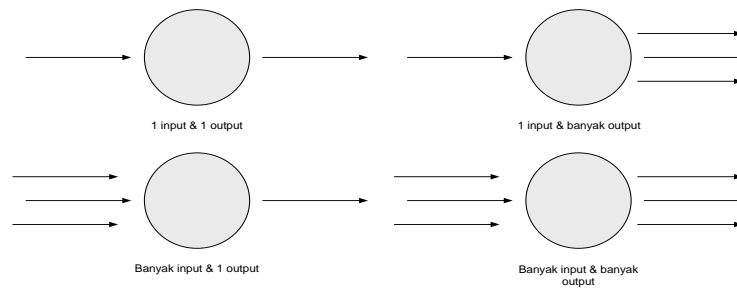
#### 2. Entitas luar tujuan

Merupakan entitas luar yang menjadi tujuan data atau informasi sistem.



Gambar 7. Entitas luar

Proses menggambarkan transformasi *input* menjadi *output*. Penamaan proses disesuaikan dengan proses atau kegiatan yang sedang dilakukan. Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan *input* dan *output*:

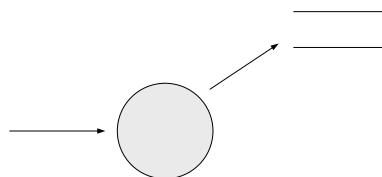


Gambar 8. Proses

*Data store* digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda bersifat jamak. *Data store* dapat berupa *file* atau *database* yang tersimpan dalam disket, *hardisk* atau bersifat manual seperti buku alamat, *file folder*.

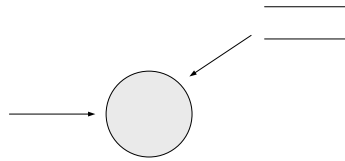
Yang perlu diperhatikan tentang *data store*:

1. Alur data dari proses menuju *data store*, hal ini berarti *data store* berfungsi sebagai tujuan atau tempat penyimpanan dari suatu proses (misalnya proses *write*).



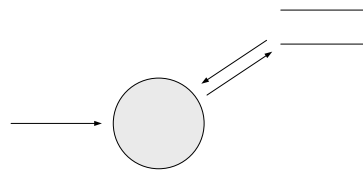
Gambar 9. Proses *write*

2. Alur data dari *store* ke proses, hal ini berarti *data store* berfungsi sebagai sumber atau proses memerlukan data (misalnya proses *read*)



Gambar 10. Proses *read*

3. Alur data dari proses menuju *data store* dan sebaliknya berarti berfungsi sebagai sumber dan tujuan (misalnya proses *read and write*).



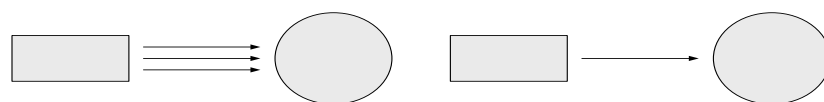
Gambar 11. Proses *read and write*

Alur data digunakan untuk menerangkan perpindahan data atau paket data dari satu bagian ke bagian lainnya. Alur data dapat berupa kata, pesan, formulir atau informasi.

Ada empat konsep tentang alur data:

#### 1. *Packets of data*

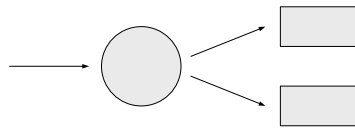
Apabila ada dua data atau lebih yang mengalir dari satu sumber yang sama menuju pada tujuan yang sama dan mempunyai hubungan digambarkan dengan satu alur data.



Gambar 12. *Packets of data*

## 2. *Diverging data flow*

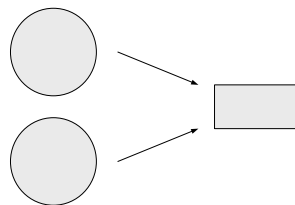
Apabila ada sejumlah paket data yang berasal dari sumber yang sama menuju pada tujuan yang berbeda atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda.



Gambar 13. *Diverging data flow*

## 3. *Converging data flow*

Apabila ada beberapa alur data yang berbeda sumber menuju ke tujuan yang sama.

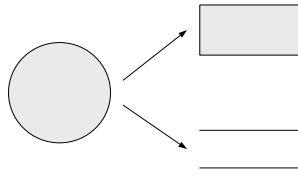


Gambar 14. *Converging data flow*

## 4. Sumber dan tujuan

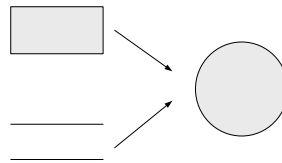
Arus data harus dihubungkan pada proses, baik dari maupun yang menuju proses.

Dari proses ke bukan proses



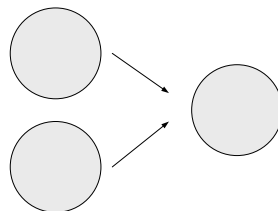
Gambar 15. Dari proses ke bukan proses

Dari bukan proses menuju proses



Gambar 16. Dari bukan proses ke proses

Dari proses ke proses



Gambar 17. Dari proses ke proses

#### **4. *Specification Control (CSPEC)***

CSPEC merepresentasikan tingkah laku sistem (pada tingkat di mana dia direferensi) di dalam dua cara yang berbeda. CSPEC berisi sebuah diagram transisi keadaan (STD) yang merupakan suatu spesifikasi sekuensial dari tingkah laku. Dia juga dapat berisi suatu tabel aktivasi proses (PAT)- sebuah spesifikasi kombinatorial dari tingkah laku.

#### **5. *Process Specification (PSPEC)***



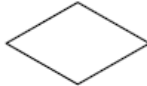

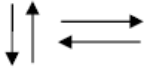



Spesifikasi proses (PSPEC) digunakan untuk menggambarkan semua proses model aliran yang nampak pada tingkat akhir penyaringan. Kandungan dari spesifikasi proses dapat termasuk teks naratif, gambaran bahasa desain program dari algoritma proses, persamaan matematika, tabel, diagram atau bagan.

#### **6. *Diagram Alur (Flowchart)***

Suatu program adalah sederetan instruksi (dalam bahasa yang dimengerti komputer yang bersangkutan) yang mengatur apa – apa yang harus dikerjakan komputer, untuk mendapatkan suatu hasil atau keluaran yang kita harapkan. Oleh karena itu, sebelum suatu program dibuat, disarankan untuk membuat logika atau urutan – urutan instruksi program tersebut kedalam suatu diagram yang disebut diagram alur (flowchart). Suatu diagram alur memberi gambaran dua dimensi berupa symbol-simbol grafis. Masing – masing simbol telah

ditetapkan terlebih dahulu fungsi dan artinya. Simbol tersebut dipakai untuk menunjukkan berbagai kegiatan operasi dan jalur pengendalian.[8]

Tabel 1. Simbol Diagram Alir

Simbol	Fungsi
 Proses	Satu atau beberapa himpunan penugasan yang dilaksanakan secara berurutan.
 Masukkan serta keluaran	Data yang akan dibaca dan dimasukkan kedalam memori komputer dari suatu alat input atau data harus melewati memori untuk dikeluarkan dari alat output.
 Kotak keputusan	Jalur yang harus dipilih pada saat pelaksanaan
 Kotak mulai serta berhenti	Simbol untuk menyatakan mulai ataupun berhenti
 Arah	Menunjukkan bagan instruksi selanjutnya
 Tampilan ( di layar atau monitor )	Menampilkan suatu data ke layar atau monitor
 menyimpan data	Digunakan untuk menyimpan data
 Penghubung di halaman yang sama	Untuk menghubungkan suatu instruksi selanjutnya di halaman yang sama



### C. Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan. (Sommerville, 2003). Pada definisi ini, ada dua istilah kunci yaitu:

1. Disiplin rekayasa, perekayasa membuat suatu alat bekerja. Menerapkan teori, metode, dan alat bantu yang sesuai, selain itu mereka menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. Perekayasa juga menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan-batasan ini.
2. Semua aspek produksi perangkat lunak, rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode, dan teori untuk mendukung produksi perangkat lunak.

Secara umum, rekayasa perangkat lunak memakai pendekatan sistematis dan terorganisasi terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun demikian, rekayasa ini sebenarnya mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan. Pengembangan

informal sangat cocok untuk pengembangan sistem *e-commerce web* membutuhkan gabungan keahlian perangkat lunak dan perancangan grafis.

### **1. Proses Perangkat Lunak**

Proses perangkat lunak adalah serangkaian kegiatan-kegiatan dan hasil-hasil relevannya yang menghasilkan perangkat lunak. Kegiatan-kegiatan ini sebagian besar dilakukan perencana perangkat lunak. Ada empat kegiatan proses dasar yang umum bagi seluruh kegiatan proses perangkat lunak. Kegiatan-kegiatan ini adalah :

1. Spesifikasi perangkat lunak, fungsionalitas perangkat lunak dan batasan kemampuan operasinya harus didefinisikan
2. Pengembangan perangkat lunak, perangkat lunak yang memenuhi spesifikasi tersebut harus diproduksi.
3. Validasi perangkat lunak, perangkat lunak harus divalidasi untuk menjamin bahwa perangkat lunak melakukan apa yang diinginkan oleh pelanggan.
4. Evolusi perangkat lunak, perangkat lunak harus berkembang untuk memenuhi kebutuhan pelanggan yang berubah-ubah.

Proses perangkat lunak yang berbeda mengatur kegiatan ini dengan cara berbeda dan dijelaskan dengan tingkat kerincian yang berbeda pula. Waktu kegiatan bervariasi, sebagaimana hasilnya. Pengaturan yang berbeda dapat menggunakan proses yang berbeda untuk menghasilkan produk dengan jenis yang sama. Namun demikian, untuk beberapa jenis aplikasi tertentu, beberapa proses lebih sesuai dari yang lainnya jika digunakan proses yang tidak sesuai,

maka kualitas penggunaan produk perangkat lunak yang akan dikembangkan tersebut mungkin berkurang.

## **2. Model Proses Perangkat Lunak**

Model proses pengembangan perangkat lunak adalah sebagai berikut :

1. Model air terjun (*waterfall*). Model ini mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi dan evolusi, dan merepresentasikannya sebagai *fase-fase* proses yang berbeda seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya.
2. Pengembangan evolusioner. Pendekatan ini berhimpitan dengan kegiatan spesifikasi, pengembangan, dan validasi. Suatu sistem awal dikembangkan dengan cepat dari spesifikasi abstrak. Sistem ini kemudian diperbaiki dengan masukan dari pelanggan untuk menghasilkan sistem yang memuaskan bagi kebutuhan pelanggan.
3. Pengembangan sistem formal. Pendekatan ini didasarkan atas pembuatan spesifik sistem matematis dan pentransformasian spesifikasi, dengan memakai metode matematis untuk membangun program. Verifikasi komponen sistem dilakukan dengan membuat argumen matematis yang disesuaikan dengan spesifikasi.

Pengembangan berdasarkan pemakaian ulang. Pendekatan ini didasarkan atas adanya komponen yang dapat dipakai untuk jumlah yang signifikan . Proses pengembangan sistem terfokus pada integrasi komponen-komponen ini ke dalam suatu sistem dan bukan mengembangkan dari awal.[10]

#### **D. Metode *Waterfall* (Model air terjun)**

Model sekuensial linier untuk *Software engineering*, sering disebut juga dengan siklus kehidupan klasik atau model air terjun. Model ini mengusulkan sebuah pendekatan kepada perkembangan *Software* yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas – aktivitas sebagai berikut :

##### **1. Rekayasa dan pemodelan sistem/informasi**

Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen yang lain seperti *software*, 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 *Software***

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada *software*. Untuk memahami sifat program yang dibangun, analis harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan. Kebutuhan baik untuk sistem maupun *software* didokumentasikan dan dilihat lagi dengan pelanggan.

### **3. Desain**

Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.

### **4. Generasi Kode**

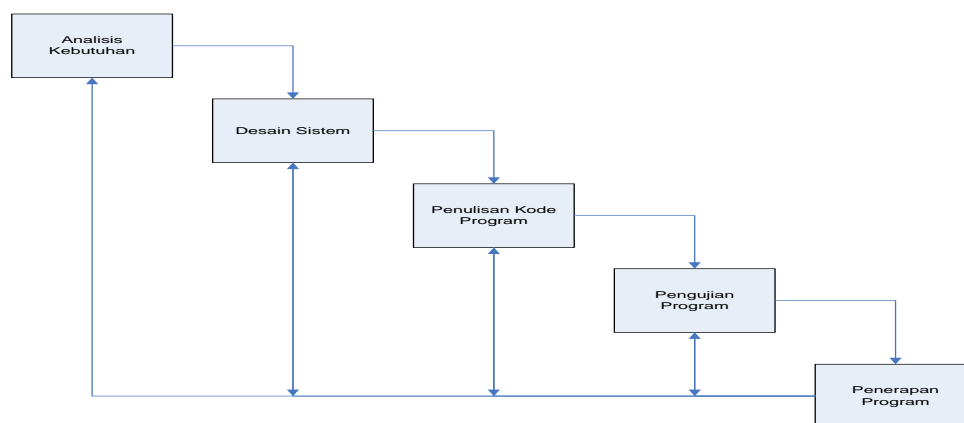
Desain harus diterjemahkan ke dalam bentuk mesin yang biasa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

### **5. Pengujian**

Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan – kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

## 6. Pemeliharaan

*Software* akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan terjadi karena kesalahan – kesalahan ditentukan, karena *software* harus disesuaikan untuk mengakomodasi perubahan – perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan *Software* mengaplikasikan lagi setiap *fase* program sebelumnya dan tidak membuat yang baru lagi.[10]



Gambar 18. Model *Waterfall*

## E. PHP ( *Personal Home Page* )

### 1. Sejarah PHP

PHP adalah teknologi yang diperkenalkan tahun 1994 oleh Rasmus Lerdorf. Beberapa versi awal yang tidak dipublikasikan digunakan pada situs pribadinya untuk mencatat siapa saja yang mengakses daftar riwayat hidup

*online*-nya. Versi pertama digunakan oleh pihak lain pada awal tahun 1995 dan dikenal sebagai *Personal Home Page Tools*. Terkandung di dalamnya sebuah *parser engine* (mesin pengurai) yang sangat disederhanakan, yang hanya mampu mengolah *macro* khusus dan beberapa utilitas yang sering digunakan pada pembuatan *homepage*, seperti buku tamu, pencacah, dan hal semacamnya. *Parser* tersebut ditulis ulang pada pertengahan 1995 dan dinamakan PHP/FI Versi 2. FI (*Form Interpreter*) sendiri berasal dari kode lain yang ditulis juga oleh Rasmus, yang menterjemahkan HTML dari data. Ia menggabungkan *script Personal Home Page Tools* dengan *Form Interpreter* dan menambahkan dukungan terhadap *server database* yang menggunakan format *mSQL*, sehingga lahirlah PHP/FI. PHP/FI tumbuh dengan pesat, dan orang-orang mulai menyiapkan kode-kode programnya supaya bisa didukung oleh PHP.

Sulit memberikan data statistik yang akurat, namun diperkirakan pada akhir 1996 PHP/FI sudah digunakan sedikitnya pada 15.000 situs *web* di seluruh dunia. Pada pertengahan 1997, angka tersebut berubah menjadi 50.000. pada saat itu juga terdapat perubahan di dalam pengembangan PHP. PHP berubah dari proyek pribadi Rasmus menjadi sebuah tim yang lebih terorganisasi. *Parser*-nya ditulis ulang dari bentuk rancangan awal oleh Zeev Suraski dan Andi Gutmans, dan *parser* baru ini adalah sebagai dasar PHP Version 3. Banyak kode utilitas yang berasal dari PHP/FI di *port* ke PHP3, dan banyak di antaranya sudah selesai ditulis ulang secara lengkap.

Pada pertengahan 1998, baik PHP/FI maupun PHP3 dikemas bersama produk – produk komersial seperti *server web* StrongHold buatan C2 dan

Linux RedHat, dan menurut survei yang dilakukan oleh NetCraft, kemungkinan PHP digunakan pada lebih dari 150.000 situs *web* di seluruh dunia. Sebagai pembandingan, angka tersebut lebih banyak dari pada pengguna *server web Enterprise Server* buatan Netscape di internet.

PHP singkatan dari *Personal Home Page Tools*, adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi *server*. Sebagian besar perintahnya berasal dari bahasa C, Perl dan Java dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi *web* menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan *server*. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI (*Common Gateway Interface*).

Pengertian dari PHP itu sendiri adalah sebuah bahasa *script server-side* yang bisa digunakan dengan bahasa HTML atau dokumennya secara bersamaan untuk membangun sebuah aplikasi di *web* yang sangat banyak kegunaannya. PHP adalah merupakan bahasa yang disertakan dalam dokumen HTML, bekerja di sisi *server* (*server side HTML-embedded scripting*). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada halaman HTML biasa, sehingga *scriptnya* tak tampak di sisi *client*.

PHP dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses *database* menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi – aplikasi yang dijalankan di atas teknologi



*web browser*, tetapi prosesnya secara keseluruhan dijalankan di atas *web server*.<sup>[4]</sup>

## 2. Cara Kerja PHP

Karena PHP merupakan aplikasi sisi *server*, maka proses dilakukan di *server*. Pada saat *browser* meminta dokumen PHP, *web server* langsung menggunakan modul PHP untuk mengolah dokumen tersebut. Jika pada dokumen terkandung fungsi yang mengakses *database* maka modul PHP akan menghubungi *database server* yang bersangkutan. Dokumen berformat PHP tersebut akan dikembalikan *web server* dalam format HTML, sehingga *source code* PHP tidak tampak di sisi *browser*. Untuk dapat menjalankan *script* PHP, dibutuhkan sebuah interpreter PHP yang bertugas menterjemahkan *script* yang ditulis, sebuah *web server* seperti Apache dan sebuah *text editor* sederhana misalnya Notepad (Windows) atau Vi (Unix).

Agar sebuah *script* PHP dianggap sebagai sebuah *script*, bukanlah dokumen biasa pada *browser*, maka *web server* harus diaktifkan. Kemudian untuk menjalankannya pada *browser* masukkan alamat *web server*, misalnya `http://localhost/`.

## 3. Alasan Menggunakan PHP

Ada beberapa alasan yang mendasari penggunaan PHP, antara lain :

### a. *Open Source*

PHP merupakan aplikasi bahasa *web* yang bisa diperoleh secara gratis.

Karena tujuan dari *open source* adalah untuk menjamin bahwa

perangkat lunak yang didistribusikan dapat ditinjau secara bebas dan berkesinambungan mengalami perbaikan secara evolusi dan seleksi, sehingga sangat dimungkinkan PHP akan terus berkembang menjadi suatu bahasa *script* yang handal.

**b. PHP *embedded***

Penulisan *script* PHP menyatu dengan HTML. Untuk membedakannya dengan sintaks HTML, *script* PHP harus diawali dengan tag `<?` dan diakhiri dengan tag `?>`. Pada saat client meminta suatu halaman *web*, *web server* mencari *section-section* PHP dan mengubahnya ke format HTML.

**c. Multi *Platform***

Aplikasi *web* yang dibangun dengan PHP berlaku secara universal. PHP dapat dijalankan di bawah sistem operasi Windows, Linux dan Unix. PHP kompatibel dengan 3 *web server* teratas, yaitu Apache HTTP *Server*, Microsoft Internet Information *Server*, Netscape Enterprise *Server* (iPlanet *Server*). Ketersediaan kode sumber dalam bahasa C membuat PHP sangat universal dan mudah disesuaikan dengan platform yang digunakan.

**d. Stabil**

PHP bersifat stabil, karena *server* tidak memerlukan boot ulang dan *software* tidak berubah secara radikal dan kompatibel.

**e. Cepat**

PHP memerlukan waktu yang sedikit untuk mengeksekusinya, terutama saat dikompilasi sebagai modul Apache di sisi Unix.

**f. PHP Mudah Berkomunikasi dengan Aplikasi Lain**

PHP mudah untuk berkomunikasi dengan program dan protokol lain. PHP menyediakan fleksibilitas maksimum untuk sejumlah *user*. Konektifitas *database* yang cukup kuat dengan dukungan *native-driver* untuk sekitar 15 *database* paling populer termasuk ODBC. PHP mendukung sejumlah protokol seperti POP3, IMAP dan LDAP. PHP5 juga mempunyai dukungan baru untuk Java dan arsitektur pendistribusian objek (COM dan CORBA).

#### **4. Kemampuan PHP**

Pada PHP, data dimasukkan dalam suatu form HTML akan secara otomatis divariabelkan dan dapat digunakan langsung sehingga tidak perlu memarsing apa yang disebut query string. Kemampuan dan kelebihan PHP antara lain :

**a. Pengaksesan *file* GIF dengan PHP**

PHP tidak terbatas hanya untuk menghasilkan output halaman HTML. PHP juga dapat digunakan untuk menghasilkan *file* gambar GIF atau bahkan agar lebih menghasilkan *file* gambar yang bersifat *stream* dari *server* ke *browser*.

**b. File Upload**

PHP mampu menerima *file upload* dari setiap *browser* yang RFC-1867 compliant. Kemampuan ini memungkinkan *user* untuk meng-*upload file* teks atau binary.

**c. HTTP Cookie**

PHP secara transparan mendukung HTTP *Cookie*. *Cookie* adalah mekanisme untuk menyimpan data pada *browser* sehingga *user* dapat diidentifikasi.

**d. Koneksi ke Database**

PHP mendukung banyak *database* dalam model native dan melalui ODBC

**e. HTTP Authentication**

Kelebihan ini hanya tersedia jika PHP berjalan sebagai modul *web server* Apache.

**f. Error Handling**

Seluruh ekspresi PHP dapat dipanggil dengan prefiks '@' yang akan mematikan *error reporting* untuk ekspresi tersebut.

**5. Dasar-dasar PHP****a. Tag-tag PHP**

Semua *script* PHP harus ditandai secara khusus dengan tag tertentu agar oleh *server* tidak dianggap sebagai *client-side script* yang akan diteruskan

ke *web browser* tanpa diolah oleh *web server*. Untuk itu, PHP mempunyai empat buah cara untuk membedakannya dari *script* HTML dan mendefinisikannya sebagai *script* PHP :

Tabel 2. *script* HTML dan mendefinisikannya sebagai *script* PHP

Cara	Keterangan
Cara I (Short Tag) <? Echo " <i>Script</i> PHP"; ?>	cara yang paling sederhana untuk memproses instruksi.
Cara II ( Default Tag ) : <?php echo " <i>Script</i> PHP"; ?>	cara penulisan seperti dokumen XML.
Cara III ( model Tag ASP): <% echo " <i>Script</i> PHP"; %>	menggunakan gaya penulisan seperti ASP.
Cara IV( <i>Script</i> Tag) : < <i>script</i> language="php"> echo " <i>Script</i> PHP"; </ <i>script</i> >	menggunakan gaya <i>Javascript</i> .

### b. Variabel pada PHP

Secara sederhana variabel dapat didefenisikan sebagai ruang memori yang disediakan sebagai tempat menampung atau menyimpan data yang ditugaskan pada identifier tertentu. Dalam PHP, variabel diawali dengan tanda dollar(\$). Secara normalnya variabel dapat dibuat dengan :

```
$a = "satu"; // terbentuk variabel $a berisi "satu"
```

Selain itu dalam PHP nilai dari suatu variabel dapat digunakan menjadi nama variabel, seperti :

```
$a = "satu"; //terbentuk variabel $a berisi "satu"
```

```
$$a = "dua"; //terbentuk variabel $$a berisi "dua"
```

Pada PHP tidak dibutuhkan deklarasi variabel secara eksplisit, tipe variable ditentukan berdasarkan konteks pemakaiannya. Dengan kata lain, jika memberi nilai string ke suatu variabel X, maka X menjadi suatu variabel bertipe string. Sebaliknya jika memberi nilai integer ke X, maka otomatis berubah menjadi tipe integer.

Suatu contoh dari konversi tipe data secara otomatis pada PHP adalah operator penjumlahan. Jika salah satu operand-nya bertipe *double*, maka semua operand lainnya dievaluasi sebagai *double* dengan hasilnya adalah *double*.

### **c. Tipe Data di PHP**

Tipe data dari suatu variabel dalam PHP tidak perlu dideklarasikan, nilai dari suatu variabel dapat langsung diberikan. Tipe data yang dapat diolah oleh PHP adalah Integer, Double, String, Boolean, Array dan Object.

#### **1. Integer**

Tipe data ini mencakup seluruh bilangan bulat dengan kisaran - 2.147.483.648 sampai 2.147.483.647. Integer dapat dibaca dalam 3 format sesuai dengan bilangan berbasis desimal, oktal (basis 8) dan hexa (basis 16). Format desimal sebagai default, octal dispesifikasikan

dengan diawali '0' dan hexa diawali dengan '0x'. Format tersebut dapat didahului tanda '-' untuk integer negatif. \$integer\_10 = 1000;  
 //Akan menghasilkan output \$integer\_10 = 1000 \$integer\_8 = -0100;  
 //Akan menghasilkan output \$integer\_8 = -512 \$integer\_16 = 0x1000;//Akan menghasilkan output \$integer\_16 = 4096.

## 2. *Double*

Tipe data ini akan merepresentasikan pecahan atau desimal. Kisaran data ini adalah 1,7E-308 sampai 1,7E+308.

## 3. *String*

String merupakan kumpulan karakter. String dapat ditutup dalam tanda petik tunggal maupun petik ganda. Pada petik tunggal akan dicetak seperti literalnya, sedangkan petik ganda menggantikan variabel dengan nilainya.

## 4. *Array*

Tipe array pada PHP mempermudah cara pengelompokan nilai-nilai yang berbeda dan mengindeksnya dengan numerik ataupun nama. Elemen array dituliskan dengan indeks dalam tanda kurung ( [ n ] ).

## 5. *Object*

*Object* adalah sebuah tipe data yang dapat berupa sebuah bilangan, variabel atau bahkan sebuah fungsi.

## F. MySQL

### 1. Pengenalan MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh menjadikannya produk turunan yang bersifat *closed source* atau komersial.

MySQL sebenarnya merupakan turunan dari SQL. SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukkan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* dapat diketahui dari cara kerja optimizer-nya dalam melakukan perintah-perintah SQL, yang dibuat oleh pemakai 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 dapat sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*. [3]

### 2. Sejarah MySQL

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang *software* dan konsultan *database* bernama MySQL AB yang bertempat di Swedia. Pada saat itu, perusahaan tersebut masih bernama



TcX DataKonsult AB, dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis *web* pada *client*.

Awalnya Michael Widenius "Monty", pengembang satu-satunya di TcX memiliki sebuah aplikasi UNREG dan rutin ISAM buatannya sendiri dan sedang mencari antarmuka SQL yang cocok untuk diimplementasikan di dalamnya. Mula-mula Monty memakai miniSQL (mSQL) pada eksperimennya itu, namun mSQL dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan.

Akhirnya Monty menghubungi David Hughes, pembuat mSQL yang sedang merilis versi kedua dari mSQL. Kemudian Monty mencoba membuat sendiri mesin SQL yang memiliki antarmuka mirip dengan SQL, tetapi dengan kemampuan yang lebih sesuai dan lahirlah MySQL.

### **3. Keistimewaan MySQL**

#### **a. Portabilitas**

MySQL dapat berjalan stabil pada berbagai sistem operasi diantaranya adalah Windows, Linux, FreeBSD, Mac OS X Servr, Solaris, Amiga, HP-UX dan masih banyak lagi.

#### **b. Open Source**

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

**c. Multi User**

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

**d. Performance Tuning**

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

**e. Tipe Data**

MySQL memiliki tipe data/kolom yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text blob, date, time, datetime, timestamp, year, set serta enum.

**f. Command dan Function**

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

**g. Keamanan**

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

**h. Skalabilitas dan Batas**

MySQL mampu menandai *database* dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel. Selain itu, batas indeks yang dapat ditampung mencapai 2 indeks pada setiap tabelnya.

**i. Konektifitas**

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

**j. Lokalisasi**

MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari 20 bahasa.

**k. Antarmuka**

MySQL memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman yang menggunakan fungsi API (*Application Programming Interface*).

**l. Client dan Tools**

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

**m. Struktur Tabel**

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

**4. Perintah Dasar SQL**

SQL merupakan bahasa ANSI (American National Standard Input) yang digunakan untuk melakukan *query* pada *database*. Semua pengoperasian

data dapat dilakukan secara mudah dengan menggunakan bahasa ini, terutama dalam pemasukkan dan seleksi data.

Hampir semua *software database* mengimplementasikan bahasa ini sebagai komponen utama dari produknya termasuk MySQL. Perintah-perintah SQL dikelompokkan menjadi lima macam :

**a. *Data Definition Language (DDL)***

DDL adalah perintah SQL yang digunakan untuk menjelaskan objek dari *database*. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka *database*. Perintah-perintah yang termasuk DDL antara lain:

1. Create : untuk membuat objek *database*
2. Alter : untuk memodifikasi objek *database*
3. Drop : untuk menghapus objek *database*

Dengan objek *database* yang dapat berupa *database*, table, index dan view

**b. *Data Manipulation Language (DML)***

DML adalah perintah yang digunakan untuk mengoperasikan atau memanipulasi isi *database*. SQL menyediakan 4 perintah DML, yaitu :

1. Select : untuk menampilkan data dari *database*
2. Delete : untuk menghapus data pada *database*
3. Insert : untuk menambahkan data ke *database*
4. Update : untuk memodifikasi data pada *database*

**c. *Security***

*Security* adalah perintah-perintah yang digunakan untuk menjamin keamanan data. Antara lain terdiri atas :

1. Grant : memberi akses ke pemakai tertentu untuk akses ke *database*
2. Revoke : untuk mencabut hak akses dari pemakai

**d. *Integrity***

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

**e. *Auxilliary***

*Auxilliary* adalah perintah-perintah pelengkap atau tambahan seperti *unload* dan *rename*.

## **5. Arsitektur Tabel MySQL**

MySQL memiliki enam tipe tabel yang masing-masing mempunyai kelebihan tersendiri. Dengan beragamnya tipe tabel ini, pengguna MySQL dapat menentukan tipe tabel yang sesuai dengan kebutuhannya.

**a. MyISAM**

MyISAM merupakan tipe tabel default pada MySQL. Tabel tipe ini lebih cepat dibandingkan jenis ISAM, karena pola aksesnya telah disesuaikan dan dioptimasi untuk pola akses SQL. Selain itu MyISAM mendukung index pada kolom bertipe TEXT dan BLOB.

**b. MERGE**

Tipe tabel MERGE baru ditambahkan pada MySQL versi 3.23.25. Kelebihan tabel ini antara lain sangat efisien dalam pencarian data, memudahkan pengaturan log tables serta mudah diperbaiki. Namun tabel ini juga memiliki banyak kekurangan, di antaranya tidak dapatnya melakukan perintah INSERT.

**c. ISAM**

Pada jenis tabel ini, data disimpan pada format mesin sehingga dapat meningkatkan kecepatan, namun bergantung pada operasi sistem yang digunakan. Kekurangan tabel ini adalah kurang sesuai untuk setiap sistem operasi, tidak dapat menampung lebih dari 4G tabel serta batas *key* yang sangat sedikit.

**d. HEAP**

Heap adalah tabel yang menggunakan *index hashed* yang terletak di dalam memori. Tabel tipe ini merupakan tabel yang tercepat pada MySQL, akan tetapi jika MySQL mengalami *crash* maka semua data didalamnya tidak dapat diselamatkan.

**e. BDB**

BDB memberikan dukungan penuh terhadap transaksi. Sejak memiliki handler BDB, MySQL mengalami kenaikan status menjadi database yang ACIDcompliant, yakni suatu *database* yang memiliki dukungan terhadap *database*.

#### f. InnoDB

InnoDB merupakan penyempurnaan dari BDB.

### 6. Tipe Data MySQL

Tipe data MySQL dapat dikatakan paling lengkap dibandingkan dengan *database server* lainnya. Beberapa tipe data MySQL adalah TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, FLOAT, DOUBLE, DATE, CHAR, VARCHAR, TEXT, LONGTEXT, dan masih banyak lagi.

### 7. Konektifitas PHP dan MySQL

Untuk menghubungkan dan menjalankan perintah-perintah MySQL dari dalam *script* PHP dibutuhkan fungsi-fungsi koneksi tersendiri, beberapa di antaranya :

#### a. `mysql_connect`

PHP menyediakan fungsi ini untuk membuka hubungan ke MySQL *server*. Fungsi ini membutuhkan tiga buah argumen : *hostname*, *username*, dan *database user password*. Dengan penggunaan :

```
$connect = mysql_connect ( $localhost, $user, $password);
```

#### b. `mysql_close`

Fungsi ini digunakan untuk menutup hubungan dengan MySQL *server*.

Dengan penggunaan : `mysql_close( );`

**c. mysql\_create\_db**

Berfungsi membuat sebuah *database* baru pada *server database* MySQL. Dengan penggunaan : `mysql_create_db($database);`

**d. mysql\_select\_db**

Berfungsi memilih sebuah *database* sebagai *database* aktif.

**e. mysql\_query**

Berfungsi mengirim perintah SQL ke *server* MySQL untuk dieksekusi.

**f. mysql\_result**

Berfungsi menampilkan data hasil query.

**g. mysql\_error**

Berfungsi menampilkan pesan kesalahan dari operasi yang dilakukan terakhir kali.

**h. mysql\_fetch\_row**

Berfungsi menampilkan baris hasil array secara satu persatu.

**G. Macromedia Dreamweaver**

Dreamweaver adalah *software* aplikasi desain *web* visual yang biasa dikenal dengan istilah *WYSIWYG—What You See Is What You Get*—intinya Anda tidak harus berurusan dengan tag-tag HTML untuk membuat sebuah situs. Macromedia belum lama ini telah mengeluarkan rilis terbaru dari Dreamweaver yaitu Dreamweaver MX, dengan penambahan beberapa fasilitas baru di dalamnya. Dreamweaver tidak hanya dapat digunakan oleh para desainer *web*, namun juga dapat digunakan oleh programmer untuk membangun halaman interaktif karena



Dreamweaver MX mendukung pula PHP, ColdFusion, ASP.NET dan lain-lain (Haerman, 2002).

## **H. XAMPP**

XAMPP merupakan salah satu paket instalasi Apache, PHP dan MySQL *instant* yang dapat kita gunakan untuk membantu proses instalasi ketiga produk tersebut. Selain paket instalasi *instant* XAMPP versi 1.6.4 juga memberikan fasilitas pilihan penggunaan PHP4 atau PHP5 Untuk berpindah versi PHP yang ingin digunakan juga sangat mudah dilakukan dengan menggunakan bantuan PHP Switch yang telah disertakan oleh XAMPP, dan yang terpenting XAMPP bersifat *free* atau gratis untuk digunakan (kadir, 2002).

## **I. Apache**

Menurut Kadir (2002) *apache* adalah perangkat lunak yang dapat digunakan sebagai *web server*. *Hypertext Transfer Protocol Daemon (HTTPD) server* yang menterjemahkan perintah-perintah tag dari *Hypertext Transfer Protocol (HTTP)*.

### **a. HTTP (*Hypertext Transfer Protokol*)**

HTTP (*Hypertext Transfer Protokol*) merupakan protokol dasar dari *World Wide Web (WWW)* yang dapat diterapkan dalam aplikasi *client-server* yang melibatkan *hypertext* (standar dokumen dalam pembuatan aplikasi *web*).

Penggunaan http yang khas adalah antara *web browser* dengan *web server*. Untuk menyediakan kehandalan dalam kinerjanya http menggunakan koneksi TCP (*Transmission Control Protokol*), akan tetapi http merupakan suatu protokol yang

tidak jelas, karena setiap transaksi yang yang ditangani diperlukan secara mandiri (*independent*), dimana setiap transaksi yang dilakukan akan menciptakan suatu koneksi TCP baru antara *client* dengan *server*, kemudian memutuskan koneksi tersebut setelah transaksi dilengkapi.

Fungsi dari *http* bukanlah hanya sebagai *protokol* yang digunakan untuk *Mentransfer hypertext*, melainkan *protokol* yang digunakan mentransmisikan informasi dengan tingkat efisiensi yang diperlukan untuk membuat lompatan *hypertext*. Data-data yang ditransmisikan oleh *protokol* ini dapat merupakan *text* asli, *hypertext*, audio, gambar atau informasi. Informasi lainnya yang bisa diakses melalui internet. Keandalan lain dari *http* adalah sifatnya yang sangat *flexibel* dalam format-format yang dapat ditangani, ketika seorang *Client* mengajukan permintaan kepada *server*, maka *client* bisa memasukkan daftar prioritas format-format yang ditanganinya (Haerman. 2007).

#### **b. HTML ( *Hypertext Markup Language* )**

HTML merupakan standar yang digunakan dalam *platform* tampilan dan isi dari sebuah dokumen pada halaman *web*. Dalam penggunaannya, *html* didefinisikan dalam *tag-tag* yang dipergunakan untuk mengatur tampilan dan isi dari dokumen *hypertext* tersebut seperti ukuran huruf, warna, *header* dokumen, pembuatan link, pengaturan paragraf, serta pembuatan tabel (Haerman, 2007).

Yang bisa dilakukan dengan HTML yaitu:

- a) Mengontrol tampilan dari *web page* dan *content*-nya.

- b) Mempublikasikan document secara *online* sehingga bisa di akses dari seluruh dunia.
- c) Membuat *online form* yang bisa di gunakan untuk menangani pendaftaran, transaksi secara online.
- d) Menambahkan *object-object* seperti *image*, *audio*, *video* dan juga java applet dalam dokumen HTML.

Contoh penulisan kode HTML :

```
<html>
<head>
<title>judul</title>
</head>
<body>
    kode-kode HTML
</body>
</html>
```

Pasangan tag <html> dan </html> menandakan bahwa kode yang terdapat didalamnya adalah kode yang diterjemahkan oleh *browser* sebagai dokumen HTML.[3]

Bagian yang terdapat dalam <html> dan </html> terbagi atas :

- 1) Kepala (*head*)
- 2) Badan (*body*)

## **J. DATABASE**

### **1. Database Management System(DBMS)**

*Database Management System(DBMS)* merupakan paket program (*Software*) yang dibuat agar memudahkan dan mengefisienkan pemasukan, pengeditan, penghapusan dan pengambilan informasi terhadap *database*. Software yang tergolong kedalam DBMS antara lain, Microsoft SQL, MySQL, Oracle, MS.

Access, dan lain-lain. Model *database* adalah suatu konsep yang terintegrasi dalam menggambarkan hubungan (*relationships*) antar data dan batasan-batasan (*constraint*) data dalam suatu sistem *database*. Model data yang paling umum, berdasarkan pada bagaimana hubungan antar *record* dalam *database* (*Record Based Data Models*), terdapat tiga jenis, yaitu :

a. Model *Database* Hirarki (*Hierarchical Database Model*)

Model *database* hirarki disebut juga model pohon, karena hubungan antar simpul digambarkan seperti struktur pohon (*tree-structured*) yang dibalik dengan pola hubungan orang tua – anak (*parent – child*). Simpul yang paling atas disebut akar (*root*) dan paling bawah disebut daun. Setiap simpul digambarkan dengan lingkaran atau kotak. Simpul yang berada di atas simpul lainnya disebut orang tua, sedangkan yang berada di bawahnya disebut anak, dimana seorang orang tua bisa mempunyai satu anak (jenis hubungan satu ke satu, *one to one*) atau mempunyai beberapa anak (jenis hubungan satu ke banyak, *one to many*). Tapi satu anak hanya boleh punya satu orang tua (jenis hubungan satu ke satu, *one to one*).

b. Model *Database* Jaringan (*Network Database Model*)

Model *database* jaringan merupakan pengembangan dari model *database* hirarki, Dimana kelemahan yang ada pada model *database* hirarki yaitu ketidakmampuannya dalam mengelola hubungan banyak ke banyak (*Many to Many*) telah dapat diatasi dengan model *database* jaringan ini. Dalam model ini, data di representasikan sebagai koleksi *record* dan hubungan antar *record*

direpresentasikan sebagai pointer. Oleh karena itu, model *database* jaringan mampu menyatakan hubungan :

1. Satu ke Satu (*One To One*, 1 : 1), satu orang tua punya satu anak.
2. Satu ke Banyak (*One To Many*, 1 : M) Satu orang tua punya beberapa anak,
3. Banyak ke Banyak (*Many To Many*, N : M), beberapa anak punya beberapa orang tua.

### c. Model *Database* Relasi (*Relational Database Model*)

Model *database* relasi merupakan model *database* yang paling banyak digunakan saat ini, karena paling sederhana dan mudah digunakan serta yang paling penting adalah kemampuannya dalam mengakomodasi berbagai kebutuhan pengelolaan *database*. Sebuah *database* dalam model ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan kolom (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*), tabel-tabel yang ada dihubungkan (*relationship*) sedemikian rupa menggunakan *field-field* kunci (*Key field*) sehingga dapat meminimalkan duplikasi data. Model *database* relasi ini dikemukakan pertama kali oleh E.F. Codd, salah seorang pakar dalam bidang *database*. Sering juga model ini disebut *Database* relasi.

### 2. Tingkatan Data Dalam *Database* Relasi

Dalam suatu sistem *database* relasi, data yang tersimpan dalam DBMS mempunyai tingkatan-tingkatan, sebagai berikut :

a. Karakter (*Characters*)

Merupakan bagian terkecil dalam *database*, dapat berupa karakter numerik (angka 0 s.d 9), huruf ( A - Z, a - z) ataupun karakter-karakter khusus, seperti \*, &. %, # dan lain-lain.

b. *Field* atau *Attribute*

Merupakan bagian dari *record* yang menunjukkan suatu item data yang sejenis, Misalnya : *field* nama, *file* NIM dan lain sebagainya. Setiap *field* harus mempunyai nama dan tipe data tertentu. Isi dari *field* di sebut Data Value. Dalam *table database*, *field* ini disebut juga kolom.

c. *Record* atau *Tuple*

*Tuple/Record* adalah kumpulan data value dari attribute yang berkaitan sehingga dapat menjelaskan sebuah *entity* secara lengkap. Misal : *Record entity* mahasiswa adalah kumpulan data value dari *field* nobp, nama, jurusan dan alamat per-barisnya. Dalam tabel *database*, *Record* disebut juga baris.

d. *Table/Entity*

*Entity* merupakan sesuatu yang dapat diidentifikasi dari suatu sistem *database*, bias berupa objek, orang, tempat, kejadian atau konsep yang informasinya akan disimpan dalam *database*. Misal. Pada sistem *database* akademik, yang menjadi *entity* adalah, mahasiswa, dosen, matakuliah dan lain-lain. Dalam aplikasi, penggunaan istilah *Entity* sering di samakan dengan istilah Tabel. (*Entity = table*). Disebut tabel, karena dalam merepresentasikan datanya di atur dalam bentuk baris dan kolom. Baris mewakili 1 *record* dan kolom mewakili 1 *field*. Dalam sistem *database* tradisional, *entity/table* ini disebut juga dengan *file*.

#### e. Database

Kumpulan dari tabel-tabel yang saling berelasi, disusun secara logis, sehingga menghasilkan informasi yang bernilai guna dalam proses pengambilan keputusan.[11]

### K. Codeigniter

*CodeIgniter* (CI) ([www.CodeIgniter.com](http://www.CodeIgniter.com)) adalah salah satu *framework* php yang tangguh dan populer. *CodeIgniter* tergolong *framework* dengan ukuran kecil dan cukup mudah di kuasai. CI juga datang dengan manual yang tergolong lengkap. *CodeIgniter* adalah sebuah *framework* PHP. *Framework* itu sendiri adalah suatu kerangka kerja yang berupa sekumpulan *folder* yang memuat *file-file* php yang menyediakan *class libraries*, *helpers*, *plugins* dan lainnya. *Framework* menyediakan konfigurasi dan teknik coding tertentu. *code Igniter* dapat di peroleh secara gratis, dengan mendownload nya di [www.CodeIgniter.com](http://www.CodeIgniter.com). Versi saat ini adalah versi 1.7.2.[5]

#### 1. Instalasi *code Igniter*

Karena berupa *folder* php, maka instalasi *CodeIgniter* , cukup dengan mengekstrak *file* kompresi yang di dapat dari situs [www.CodeIgniter.com](http://www.CodeIgniter.com) dan menempatkannya di dalam *folder directory web server* anda. Kemudian *edit file* config.php yang ada di application/config/config.php dan set base URL anda, misal : <http://localhost/ci>. Jika anda hendak menggunakan *database*, *edit file* database.php yang ada di application/config/database.php. Setting *database* anda di *file* tsb. Setelah di lakukan instalasi dan setting, test lah *code Igniter* anda

dengan membuka *browser* anda dan menetikkan url yang anda taruh di *file* *config.php* di atas, misal : <http://localhost/ci>. Jika muncul halaman *welcome* ala *code* Igniter, maka *code* Igniter telah siap di gunakan.

## 2. Konsep Model-View-Controller (MVC)

Konsep MVC adalah konsep pemisahan antara *logic* dengan tampilan dan *database*. Manfaat konsep ini adalah, membuat *coding logic* lebih *simple*, karena sudah di pisah dengan *code* untuk tampilan dan membuat programmer dapat bekerja secara terpisah dengan *designer*. *Programmer* mengerjakan *logic*, sedangkan *designer* berkuat dengan *design* dan tampilan.

Model → Merupakan *code* struktur data. Model berisi fungsi di dalam pengolahan *database*. Script *Sql* masuk di sini.

View → Merupakan *code* untuk menampilkan tampilan suta program. Tampilan dapat berupa *web page*, *header*, *footer* dan apa saja yang berjenis tampilan.

Controller → Merupakan *code* untuk *logic*, algoritma dan sebagai penghubung antara model, *View*, dan sumber lain yang di perlukan untuk mengolah *HTTP request* dan *generate web page*.

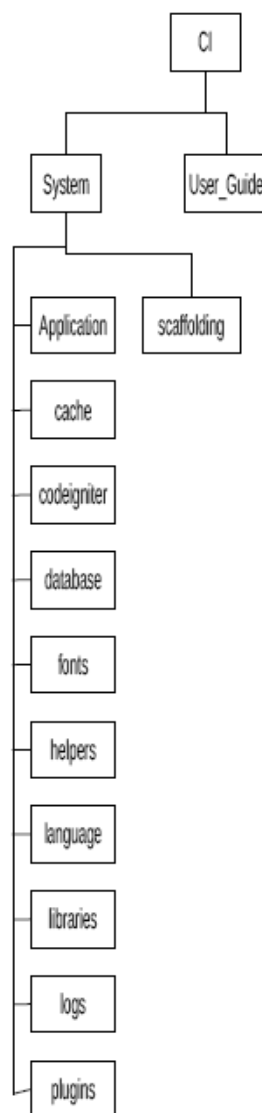
CI menerapkan pola MVC yang *flexible*, karena model dapat tidak di gunakan.

Anda dapat hanya menggunakan *Controller* dan *View* saja dalam menggunakan CI tanpa Model. Jika anda tidak memerlukan pemisahan di dalam struktur data dan *database* atau menganggap penggunaan model hanya menambah kompleks aplikasi dengan keuntungan yang kurang sebanding, maka anda dapat tidak menggunakan model.



### 3. Struktur *file* CI

CI adalah sebuah *php framework* yang berupa kumpulan *folder* dan *file* *php*, *java script*, *css*, *txt* dan *file* berbasis web lainnya dengan setting tertentu untuk menggunakannya dan menyediakan *library* dan *helper* yang dapat di manfaatkan di dalam pemrograman *php*. CI di jalankan *under web* dan harus dengan *web server*. Program CI cukup di letakkan di bawah *folder directory web server* anda. Berikut adalah struktur *file* CI.[9]



Gambar 19. Struktur *file* CI

## **L. Pemrograman Berorientasi Objek**

*Object-Oriented Programming* (OOP) adalah sebuah pendekatan untuk pengembangan / development suatu software dimana dalam struktur software tersebut didasarkan kepada interaksi object dalam penyelesaian suatu proses/tugas. Merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak dalam skala besar.

### **1. Objek**

Dalam ilmu komputer, sebuah obyek adalah entitas apapun yang dapat dimanipulasi oleh perintah dari sebuah bahasa pemrograman, seperti nilai (ilmu komputer), variabel, fungsi, atau struktur data. (Dengan kemudian pengenalan pemrograman berorientasi objek kata yang sama, "objek", mengacu pada contoh tertentu dari suatu kelas Konsep modern "objek" dan pendekatan berorientasi objek untuk pemrograman diperkenalkan oleh bahasa pemrograman Simula awalnya dirilis pada 1967, yang dipopulerkan oleh Smalltalk merilis dua tahun kemudian pada tahun 1969, dan menjadi alat standar perdagangan dengan penyebaran C++ awalnya dirilis pada tahun 1983.

### **2. Class**

Dalam pemrograman berorientasi obyek, sebuah kelas adalah suatu konstruksi yang digunakan sebagai cetak biru (atau template) untuk membuat objek dari

kelas itu. Kumpulan atas definisi data dalam suatu unit untuk tujuan tertentu. Cetak biru ini menjelaskan negara dan perilaku bahwa objek dari kelas semua berbagi. Objek dari suatu kelas tertentu disebut sebuah instance dari kelas. Kelas yang mengandung (dan digunakan untuk menciptakan) yang misalnya dapat dianggap sebagai jenis objek, misalnya contoh objek dari "Buah" kelas akan menjadi tipe "Buah".

Bahasa pemrograman yang mendukung kelas berbeda dalam dukungan mereka untuk berbagai kelas-fitur terkait. Kebanyakan mendukung berbagai bentuk warisan kelas. Banyak bahasa juga mendukung enkapsulasi menyediakan fitur, seperti akses specifiers. Kelas dapat mempercepat pembangunan dengan mengurangi mubazir kode program, *testing* dan *bug fixing*. Jika sebuah kelas telah benar-benar teruji dan dikenal sebagai 'padat karya', biasanya benar bahwa dengan menggunakan atau memperluas kelas diuji dengan baik akan mengurangi jumlah bug dibandingkan dengan penggunaan baru yang dikembangkan atau ad hoc kode di final *output*. Selain itu, menggunakan kembali kelas efisien berarti bahwa banyak bug yang perlu diperbaiki dan hanya satu tempat ketika masalah yang ditemukan.

### **3. Enkapsulasi**

Enkapsulasi adalah menyembunyikan detail implementasi sementara memusatkan pada antarmuka. Tujuannya adalah membuat sebuah abstraksi yang memaksa programmer berpikir secara konseptual. Biasanya, anggota-anggota data dari sebuah kelas terlihat oleh penggunanya. Jika sebuah anggota data harus dibuat agar dapat diakses oleh client kelas, maka kelas tersebut menyediakan sebuah

metode yang memeriksanya dan mengembalikan nilainya. Saat sebuah kelas mengekspos anggota data, ia dikatakan memecahkan enkapsulasi.

#### **4. Inheritansi**

Dalam Pemrograman Berbasis Objek, inheritance atau pewarisan adalah suatu keadaan dimana suatu kelas baru mewarisi seluruh variabel atau data dan method yang dimiliki oleh kelas yang menjadi induknya (parents). Suatu kelas bisa dikatakan mewarisi sebuah kelas lain apabila kelas tersebut memiliki semua variabel dan method yang dimiliki kelas induknya dan dia sendiri memiliki variabel dan atau method sendiri yang tidak dimiliki oleh kelas induk.

#### **5. Polimorfisme**

Polimorfisme, yang berarti satu objek dengan banyak bentuk, adalah konsep sederhana yang memperbolehkan method memiliki beberapa implementasi yang dipilih berdasarkan tipe objek yang dilewatkan pada pengerjaan metode. Ini dikenal sebagai *overloading method*. Ini akan memungkinkan method yang sama untuk Anjing, misalkan memperlihatkan perilaku yang benar-benar berbeda. Jadi secara objek, Polimorfisme adalah suatu bentuk fungsi dalam orientasi objek yang digunakan secara bersama-sama untuk berbagai objek dan berbagai tujuan. Contoh polimorfisme yang digunakan oleh berbagai objek adalah fungsi penjumlahan. Fungsi penjumlahan dapat digunakan oleh objek integer maupun objek real.[13]