BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Dalam konsep dasar sistem akan membahas mengenai pengertian dari sistem, yaitu :

2.1.1. Pengertian Sistem

Pada saat ini banyak pakar yang telah mendalami masalah sistem untuk kebutuhannya, sehingga definisi sistem menjadi beragam. Pengertian sistem lainnya yaitu :

Menurut Abdul Kadir (2003:54) sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan.

Menurut Budi Sutedjo Dharma Oetomo (2006:168) sistem merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan.

Dari pengertian sistem menurut beberapa pakar tersebut dapat disimpulkan bahwa sistem adalah kumpulan dari elemen-elemen atau sub-sub sistem yang saling berhubungan dan bekerja sama secara harmonis untuk mencapai suatu tujuan.

2.2. Konsep Dasar Informasi

Dalam konsep dasar informasi akan membahas mengenai pengertian dari data dan informasi, yaitu :

2.2.1. Pengertian Data

Menurut Humadiana dan Evi (2006:20), data merupakan fakta-fakta, simbol/karakter, data mentah, atau observasi yang menggambarkan suatu fenomena tertentu.

Sedangkan menurut Edhy Sutanta (2003:9), data merupakan bahan keterangan tentang kejadian-kejadian nyata atau fakta-fakta yang dirumuskan dalam sekelompok tertentu yang tidak acak yang menunjukkan jumlah, tindakan, atau hal.

Dari beberapa pandangan mengenai data yang telah dipaparkan diatas dapat disimpulkan bahwa, data merupakan kejadian nyata, fakta, bilangan, ataupun karakter yang belum memiliki makna.

2.2.2. Pengertian Informasi

Adapun beberapa definisi informasi menurut beberapa pakar, antara lain :

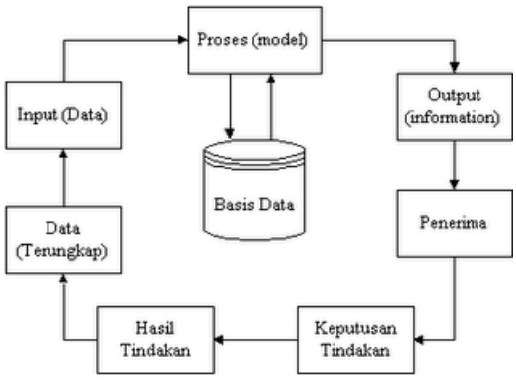
Menurut McFadden, dkk dalam Abdul Kadir (2003:31) mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

Sedangkan menurut Davis dalam Abdul Kadir (2003:31) informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Dari beberapa definisi informasi menurut beberapa pakar tersebut dapat disimpulkan bahwa informasi adalah data yang sudah diolah yang memiliki arti sehingga berguna bagi penggunannya.

2.2.2.1. Siklus Informasi

Informasi merupakan suatu proses perubahan dunia menjadi informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu. Data yang diolah melalui suatu model menjadi informasi, *user* kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut kemudian akan ditangkap sebagai *input*, diproses kembali melalui suatu model dan seterusnya membentuk suatu siklus. Adapun siklus informasi sebagai berikut:



Gambar 2.1 Siklus Informasi

Sumber : Edhy Sutanta “Sistem Informasi Manajemen (2003:32)

2.3. Konsep Dasar Sistem Informasi

Dalam konsep dasar sistem informasi akan membahas mengenai pengertian dari sistem informasi dan komponen dari sistem informasi, yaitu :

2.3.1. Pengertian Sistem Informasi

Menurut Abdul Kadir (2003:10), sistem informasi adalah mencakup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu proses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

Sedangkan menurut Alter (1992), sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Abdul Kadir (2003:31)

Berdasarkan beberapa definisi diatas, sistem informasi adalah suatu sistem yang terdiri dari komponen-komponen dalam organisasi yang saling berinteraksi untuk melakukan suatu proses dan bekerja sama untuk memberikan informasi bagi pengambil keputusan serta untuk mencapai suatu tujuan.

2.3.2. Komponen Sistem Informasi

Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok. Blok tersebut masing-masing saling berinteraksi untuk membentuk satu kesatuan untuk mencapai sasaran. Komponen atau blok tersebut yaitu :

1. Komponen Masukan (*Input*)

   Input mewakili data yang masuk kedalam sistem informasi.

2. Komponen Model

Terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan pada basis data dengan cara yang sudah ada untuk menghasilkan keluaran yang diinginkan.

3. Komponen Keluaran (*Output*)

Produk atau hasil akhir dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajeman serta semua pemakai sistem.

4. Komponen Teknologi

Teknologi merupakan alat dalam sistem informasi, teknologi yang digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* dan membantu dalam pengendalian sistem.

5. Komponen Basis Data

Basis data merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras (*hardware*) komputer dan menggunakan perangkat lunak (*software*) untuk memanipulasinya.

6. Komponen Kontrol

Komponen kontrol merupakan pengendalian yang dirancang untuk menanggulangi gangguan terhadap sistem informasi.

Dari semua komponen diatas harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada maka sistem informasi tidak bisa menjalankan fungsinya.

2.4. Konsep dasar Sistem Akademik

Dalam konsep dasar sistem akademik akan membahas mengenai akademik dan sistem informasi akademik, yaitu :

2.4.1. Akademik

Akademik adalah seluruh lembaga pendidikan formal baik pendidikan anak usia dini, pendidikan dasar, pendidikan menengah, pendidikan kejuruan maupun perguruan tinggi yang menyelenggarakan pendidikan vokasi dalam satu cabang atau sebagian cabang ilmu pengetahuan, teknologi, dan/atau seni tertentu. (http://id.wikipedia.org/wiki/Akademik 23/05/2014)

2.4.2. Sistem Informasi Akademik

Sistem informasi akademik merupakan salah satu sistem informasi untuk pengelolaan manajemen akademik mulai dari proses penjadwalan belajar mengajar hingga pada penilaian siswa, yang dapat diakses secara *online* baik Siswa, Wali Kelas, dan Kepala sekolah.

2.5. *Unified Modeling Language (UML)*

Menurut Adi Nugroho (2005:3), *Unified Modeling Language (UML)* adalah alat bantu analis serta perancangan perangkat lunak berbasis objek.

Menurut Martin Fowler (2005:1), *Unified Modeling Language (UML)* yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (*Object Oriented*).

Berdasarkan beberapa pendapat diatas yang dikemukakan dapat ditarik kesimpulan bahwa *Unified Modeling Language (UML)* adalah alat bantu analisis

untuk membantu mendokumentasikan, menspesifikasi, dan membangun sistem perangkat lunak.

Berikut adalah macam-macam diagram dalam *Unified Modeling Language (UML)*, yaitu :

1. ***Use Case Diagram***

Menurut Martin Fowler (2005:141), *Use Case* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan member sebuah narasi tentang bagaimana sistem tersebut digunakan. Ini mirip dengan diagram konteks yang digunakan dalam metode terstruktur, karena menampilkan batasan sistem dan interaksi dengan dunia lain. *Use Case* menampilkan *actor*, *use case*, dan hubungan antara mereka :

a. Aktor mana yang menggunakan *use case* mana.

b. *Use case* mana yang memasukkan *use case* lain.

2. ***Activity Diagram***

Menurut Martin Fowler (2005:163), *Activity diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan alur kerja. Dalam beberapa hal, diagram ini mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung *behavior* paralel.

3. ***Sequence Diagram***

Menurut Martin Fowler (2005:81), Sebuah *sequence diagram*, secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan

sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *use case*.

4. ***Collaboration Diagram***

Menurut Sholiq (2006:12), *Collaboration diagram* menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap *message* memiliki *sequence number*, dimana *message* dari *level* tertinggi memiliki nomor 1.

5. ***Class Diagram***

Menurut Martin Fowler (2005:53), *Class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara mereka. *Class diagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. *UML* menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah *class.*

6. ***Component Diagram***

Menurut Sholiq (2006:16), *Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tetapi dapat juga dari komponen-komponen yang lebih kecil.

7. ***Deployment Diagram***

Menurut Sholiq (2006:18), *Deployment diagram* menggambarkan detail bagaimana komponen pada *deployment* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server, atau piranti keras lainnya), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server* dan hal-hal lain yang bersifat fisikal.

2.6. Konsep Dasar Sistem Basis Data (*Database*)

Dalam konsep dasar sistem basis data (*database*) akan membahas mengenai tentang pengertian dari basis data (*database*) dan sistem basis data (*database*), yaitu :

2.6.1. Basis Data (*Database*)

Menurut Fathansyah, Ir (2007:2), basis data terdiri dari 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar atau kombinasinya.

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang, seperti :

a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansasi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan atau kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan.

Pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan seperti berikut ini :

1. Kecepatan dan Kemudahan (*Speed*)

Memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika menyimpan data secara manual.

2. Efisiensi Ruang Penyimpanan (*Space*)

Efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi antar kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

Pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data, domain data, keunikan data, dan sebagainya, yang secara

ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

4. Ketersediaan (*Availability*)

Pertumbuhan data sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Data yang sudah jarang atau bahkan tidak pernah lagi digunakan, dapat diatur untuk dilepaskan dari sistem basis data yang sedang aktif baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan *off-line*.

5. Kelengkapan (*Completeness*)

Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka tidak hanya dapat menambahkan *record-record* data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan *field-field* pada suatu tabel.

6. Keamanan (*Security*)

Dapat menentukan pemakai yang boleh menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.

7. Kebersamaan Pemakaian (*Sharability*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja, atau di satu lokasi saja atau oleh satu sistem/aplikasi saja. Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan multiuser, akan dapat memenuhi

kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data atau kondisi *deadlock*.

2.6.2. Sistem Basis Data (*Database*)

Menurut Fathansyah, Ir (2007:9), sistem basis data merupakan sistem yang terdiri atas kumpulan *file* (tabel) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi *file-file* (tabel-tabel) tersebut.

Dalam sebuah sistem basis data terdapat komponen-komponen utama, yaitu:

a. Perangkat Keras (*Hardware*)
b. Sistem Operasi (*Operating System*)
c. Basis Data (*Database*)
d. Sistem Pengelola Basis Data (*Database Management System/DBMS*)
e. Pemakai (*User*)

Berdasarkan pendapat diatas dapat disimpulkan bahwa, sistem basis data merupakan sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah menjadi informasi, dan membuat informasi tersebut tersedia saat dibutuhkan dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai (*user*) lain untuk mengakses dan memanipulasi tabel-tabel tersebut.

2.7. *Hypertext Preprocessor* (*PHP*)

Menurut Abdul Kadir (2009:2), *Hypertext Preprocessor* (*PHP*) merupakan bahasa berbentuk *script* yang ditempatkan dalam *server* dan diproses di *server*.

Menurut Suprianto (2008:15), *PHP* tergolong sebagai perangkat lunak *open source* yang diatur dalam aturan *general purpose licences* (GPL). Pemrograman *PHP* sangat cocok dikembangkan dalam lingkungan web, karena *PHP* bisa dilekatkan pada script HTML (*HyperText Markup Language*) atau sebaliknya. Secara khusus *PHP* dirancang untuk membentuk aplikasi web dinamis. Maksudnya, *PHP* mampu menghasilkan *website* yang secara terus-menerus hasilnya bisa berubah-ubah sesuai dengan pola yang diberikan. Hal tersebut tergantung pada permintaan *client browser*-nya (bisa menggunakan *browser Opera*, *Internet Explorer*, *Mozzila*, dan lainnya). Pada umumnya, pembuatan web dinamis berhubungan erat dengan *database* sebagai sumber data yang akan ditampilkan.

2.8. *Framework*

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/ prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau *class* dari awal.

Ada beberapa alasan mengapa menggunakan *Framework* :

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *Framework* (dengan syarat programmer mengikuti pola standar yang ada).

3. Umumnya *Framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga tidak perlu membangun dari awal.
4. Lebih bebas dalam pengembangan jika dibandingkan dengan CMS.

2.8.1. *CodeIgniter*

Menurut Hakim (2010:8), *CodeIgniter* adalah sebuah *framework PHP* yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis *PHP* dibanding jika menulis semua kode program dari awal.



Gambar 2.2 Logo CodeIgniter

Sumber : Hakim "Membangun Web Berbasis *PHP* dengan Framework CodeIgniter" (2010:9)

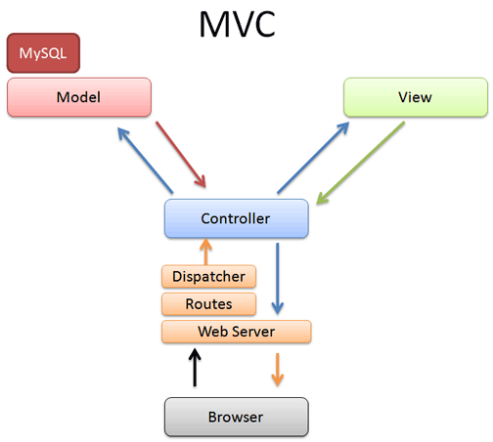
CodeIgniter merupakan *PHP framework* yang menerapkan sistem berbasis MVC (*Model-View-Controller*) yang secara sederhana dapat diartikan bahwa *CodeIgniter* memisahkan komponen-komponen didalam pengkodean aplikasi berbasis web, sehingga diharapkan nantinya lebih mudah untuk dikelola.

2.8.2. *Design Patern* : MVC (*Model, View, Controller*)

CodeIgniter adalah *framework PHP* yang dibuat berdasarkan kaidah *Model-View-Controller* (MVC). Dengan MVC, maka memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Sehingga dalam sebuah pengembangan web, seorang *programmer* bisa berkonsentrasi pada *core-system*, sedangkan web *designer* bisa berkonsentrasi pada tampilan web. Menariknya, *script PHP*, *query*

MySQL, Javascript dan CSS bisa saling terpisah, tidak dibuat dalam satu *script* berukuran besar yang membutuhkan *resource* besar pula untuk mengesekusinya.

Adapun alur program aplikasi berbasis *framework CodeIgniter* dapat dilihat pada gambar dibawah ini :



Gambar 2.3 Arsitektur MVC

Sumber : Hakim "Membangun Web Berbasis *PHP* dengan Framework CodeIgniter" (2010:4)

Gambar diatas menerangkan bahwa ketika datang sebuah *user request*, maka akan ditangani oleh *Controller*, kemudian *Controller* akan memanggil *Model* jika memang diperlukan operasi *database*. Hasil dari *query* oleh *Model* kemudian akan dikembalikan ke *Controller*. Selanjutnya *Controller* akan memanggil *View* yang tepat dan mengkombinasikannya dengan hasil *query Model*. Hasil akhir dari operasi ini akan ditampilkan dibrowser.

Dalam konteks *CodeIgniter* dan aplikasi berbasis web, maka penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori, yaitu :

1. *Model*

   Berisi kode program (berupa OOP *class*) yang digunakan untuk memanipulasi *database*.

2. *View*

   Berisi *template* HTML/XML atau *PHP* untuk menampilkan data pada *browser*.

3. *Controller*

   Berisi kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol *Model* dan *View*).

2.8.3. *Smarty*

Smarty merupakan salah satu *PHP template engine* yang memiliki fungsi untuk memisahkan antara *file-file* proses (.php) dan *file-file* tampilan (.html atau .css).

Beberapa tujuan dari *smarty* :

1. Memisahkan *file* presentasi dari *file* kode aplikasi.
2. Perkembangan cepat / penyebaran untuk *programmer* dan desainer.
3. Cepat dan mudah dalam perawatannya.
4. Sintaks mudah dimengerti.
5. Fleksibel untuk pengembangan khusus.
6. Gratis dan *open source*. (http://www.smarty.net/)

2.8.4. *CiSmart*

CiSmart merupakan modifikasi dari *framework CodeIgniter* yang digabungkan bersama dengan *Smarty template engine*. Dengan demikian bahwa

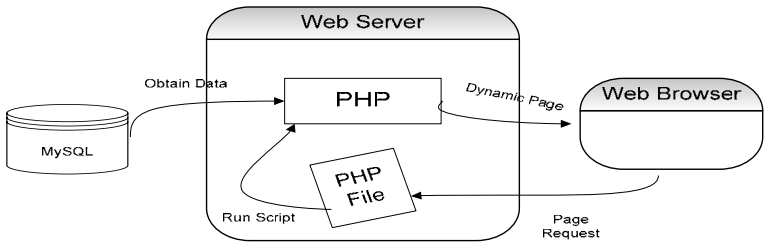
dasar *framework* yang digunakan tetap menggunakan *CodeIgniter* yaitu dengan konsep MVC (*Model-View-Controller*), sedangkan untuk pengelolaan *View* digantikan dengan menggunakan *Smarty template engine*.

2.9. *Database MySQL*

Menurut Abdul Kadir (2009:14), *MySQL* merupakan *software* yang tergolong *database server* dan bersifat *open source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja untuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di Internet secara gratis. *MySQL* juga bersifat *multiplatform*. *MySQL* dapat dijalankan pada berbagai sistem operasi.

Pengaksesan data dalam *database* dapat dilakukan dengan mudah melalui *SQL* (*Structured Query Language*). Data dalam *database* bisa diakses melalui aplikasi non-web (misalnya dengan *Visual Basic*) maupun aplikasi web (misalnya dengan *PHP*).

2.9.1. *PHP* dan *MySQL*



Gambar 2.4 Konsep Kerja *Database MySQL* dan *PHP*

Gambar diatas merupakan konsep kerja antara *database MySQL*, web *server* (*Apache*), *PHP*, dan *browser client* yang menjelaskan :

a. Pengunjung web browser melakukan permintaan pengiriman halaman web (*web request*) dengan menggunakan URL standar.
b. Web *server*, dalam hal ini adalah *Apache*, mengenali permintaan halaman web tersebut, kemudian menerjemahkan *file* yang mengandung *script PHP*.
c. Halaman web yang terkirim harus mengandung *script PHP* berisi perintah untuk menciptakan koneksi ke *database MySQL* berupa *SQL query*.
d. *Database MySQL* memberikan tanggapan atas permintaan sambungan ke *MySQL* dan *SQL query* sesuai dengan isi *script PHP* yang dikirim.
e. *Script PHP* menyimpan hasil *query database MySQL* menjadi satu atau lebih variable *PHP*, dengan pernyataan *echo*, maka hasil *request* dikeluarkan.
f. Selanjutnya, *PHP* meng*copy* hasilnya dalam bentuk HTML yang telah diciptakan oleh web *server*.
g. Web *server* mengirim HTML yang sudah jadi ke web *browser client*.

2.10. *CodeLobster PHP Edition*

CodeLobster PHP Edition merupakan salah satu *PHP* IDE (*PHP code editor*) yang dapat memudahkan dalam melakukan aktivitas *coding PHP*. *CodeLobster PHP Edition* ini bisa di jadikan alternatif *code editor*, selain memiliki ukuran instalasi yang kecil, *CodeLobster PHP Edition* ini juga memiliki *plugin* untuk memudahkan kita dalam melakukan *coding* untuk membuat *plugin*, *themes* CMS seperti *wordpress, joomla, drupal*. Sedangkan untuk *framework*nya mendukung *CakePHP, CodeIgniter, Smarty, Symfony* dan *Yii*.

CodeLobster PHP Edition ini juga tidak hanya mendukung *PHP* saja, tetapi bisa juga digunakan untuk edit *file* lainnya seperti HTML, CSS, *JavaScript*, dan XML. (http://www.codelobster.com)

2.11. *XAMPP*

Menurut Riyanto (2011:1), XAMPP merupakan paket *PHP* dan *MySQL* berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis *PHP*. XAMPP mengkombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket. Beberapa paket yang dibundel adalah *Apache*, HTTPD, *mod_autoindex_color module, FileZilla, OpenSLL*, *MySQL*, *PHP*, FTP *Server*, PEAR, *PHPMyAdmin*, dan lainnya.

Riyanto (2011:5), Setiap *web server* memiliki struktur direktori kerja (*webroot* atau *document root*) yang berbeda. Direktori ini difungsikan untuk menyimpan seluruh *file* aplikasi *web*, termasuk dokumen *PHP*.

Pada *web server* lokal, seperti dalam paket XAMPP, tersedia sebuah *folder* kerja dengan nama *htdocs*. Pada paket ini, *folder* kerja tersebut dapat di temukan pada *folder* "xampp". Dengan meletakkan *folder* kerja pada direktori *htdocs*, maka dapat mengaksesnya dengan alamat *http://localhost/namafolder* atau dengan menggunakan nomor *IP Address* pada komputer.

2.12. *Web Server Apache*

Apache merupakan turunan dari *web server* yang dikeluarkan oleh NSCA, yaitu NSCA HTTPD sekitar tahun 1995-an. Pada dasarnya *apache* adalah

"*Apatchy(patch)*" dan pengganti dari *NSCA HTTPD. Apache web server* paling banyak digunakan pada *Word Wide Web* (www). *Web server* menunggu permintaan dari klien yang menggunakan *browser* seperti *Netscape, Mozilla*, dan lain-lain. *Web server* dalam berkomunikasi dengan kliennya, menggunakan protokol HTTP (*Hyper Text Transfer Protocol*).

Apache berada di bawah GNU *general public license* yang bersifat gratis sehingga *apache* dapat di *download* garis. Saatu ini *apache* banyak digunakan sebagai *web server* untuk portal-portal besar.

Apache merupakan produk yang istimewa karena berisi hal-hal yang diperlukan oleh sebuah web, dapat diakses, dan gratis. *Apache* juga merupakan *web server* yang *open source*. Ini berarti setiap yang menggunakan produk ini mempunyai akses ke sumber kode (*source code*). Jika ingin menambahkan sesuatu tinggal dituliskan ke dalam sumber kode dan menjalankannya. Apabila ditemukan kesalahan (*bug*) dapat segera diperbaiki.

2.13. Topologi Jaringan

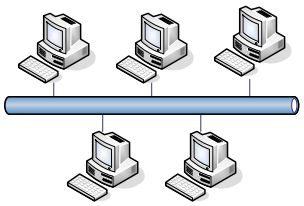
Topologi jaringan adalah tipe jaringan yang berdasarkan susunan arsitektur. Ada beberapa topologi jaringan :

a. Topologi *Bus*

Topologi *Bus* juga sering disebut dengan *Ethernet*. Topologi *Bus* menyediakan satu saluran (berupa kabel sebagai pemberi sinyal) untuk komunikasi semua perangkat, sehingga seluruh perangkat harus menggunakan saluran tersebut. Oleh karena itu hanya ada dua perangkat yang saling

berkomunikasi dalam suatu saat. Masing-masing komputer melakukan kominakasi data melalui *Bus* tersebut dengan cara mengirim dan mengambil data yang dimaksud.

Kelemahan dari topologi ini adalah jika terjadi gangguan atau kerusakan pada salah satu komputer, maka jaringan secara keseluruhan akan terganggu bahkan komunikasi dalam jaringan tersebut akan putus sama sekali. Hal ini karena seluruh proses komunikasi data hanya menggunakan satu jalur saja. Hal lain yang mengganggu adalah ketika kondisi lalu-lintas data sedang padat, dapat memungkinkan terjadinya tabrakan komunikasi (*collision*) antar data, sehingga kecepatan transfer data akan menurun.



Gambar 2.5 Topologi *Bus*

Karakteristik jaringan dengan topologi *Bus* adalah :

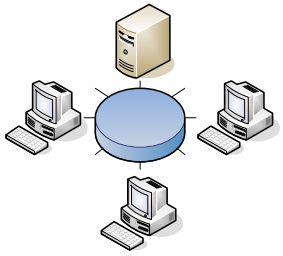
1. Biaya instalasi sangat murah.
2. Kecepatan rata-rata transfer informasi untuk setiap perangkat sangat lambat karena harus bergantian menggunakan saluran.
3. Sulit untuk manajemen jaringan.
4. Sulit untuk *expand* (menambah) jaringan.

b. Topologi Ring

Sesuai dengan namanya, yaitu *ring* atau cincin. Pada topologi ini, seluruh komputer dihubungkan sedemikian rupa sehingga membentuk seluruh cincin.

Komunikasi data melalui topologi ini cukup unik. Karena ada tidak langsung ditransfer ke komputer yang dituju, akan tetapi data tersebut diedarkan secara berkeliling sepanjang jaringan melalui media yang disebut dengan *token*.

Komputer yang hendak mentransfer data ke komputer lain meletakkan data tersebut ke dalam *token*, kemudian *token* berkeliling dari satu komputer ke komputer lain. Komputer yang lain kemudian akan melihat apakah data yang dibawa oleh *token* merupakan data yang ditujukan untuk dirinya atau bukan. Bila ditujukan untuk dirinya, data tersebut akan disimpan. Apabila tidak, maka data tersebut akan diteruskan ke komputer selanjutnya. Seperti halnya dengan topologi *Bus*, topologi ini hanya menggunakan satu jalur komunkasi sehingga mempunyai kelemahan bila salah satu komputer mengalami gangguan, maka jaringan secara keseluruhan juga akan terganggu. Akan tetapi *transfer* data melalui topologi ini relatif lebih cepat daripada topologi Bus. Sekemanya ditunjukkan pada gambar berikut ini :



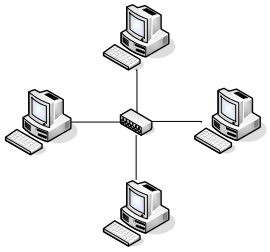
Gambar 2.6 Topologi Ring

c. Topologi Star

Susunan fisik dari topologi ini mirip sebuah bintang atau *star*. Berbeda dengan kedua topologi sebelumnya (*Bus* dan *ring*) yang hanya menggunakan satu

jalur transmisi, pada topologi *star* masing-masing *client* terhubung ke *server* dengan jalur yang berbeda. Komunikasi pada jaringan diatur di sentral jaringan,

pada *server*-nya. Topologi ini mempunyai kelebihan dibanding kedua topologi sebelumnya, yaitu bila terjadi kerusakan pada salah satu *client*, tidak akan membawa dampak bagi bagian jaringan yang lain. Jaringan hanya akan terganggu apabila terjadi kerusakan atau gangguan pada *server*-nya. Selain itu, kecepatan komunikasi datanya relatif lebih baik daripada kedua topologi yang lain.



Gambar 2.7 Topologi Star