

BAB III

LANDASAN TEORI

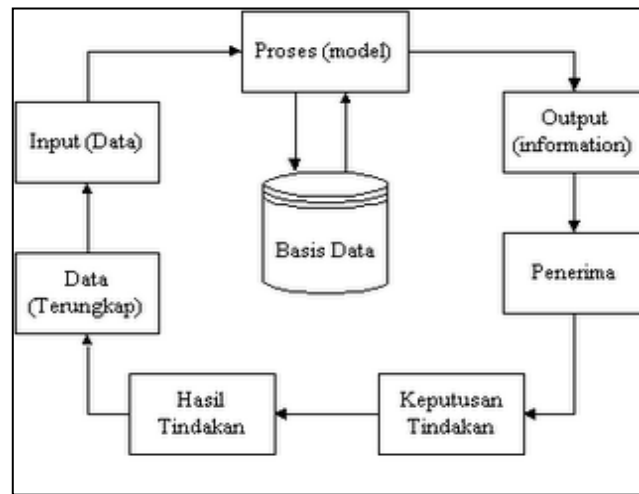
III.1 Sistem Informasi

III.1.1 Definisi Sistem Informasi

Definisi sistem informasi adalah kumpulan informasi didalam sebuah basis data menggunakan model dan media teknologi informasi digunakan didalam pengambilan keputusan bisnis sebuah organisasi. Di dalam suatu organisasi, informasi merupakan sesuatu yang penting didalam mendukung proses pengambilan keputusan oleh pihak manajemen. Menurut Raymond Mcleod, "Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang". Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan

kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

Data yang diolah melalui suatu model menjadi informasi, penerima 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 akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi ini dapat digambarkan sebagai berikut:



Gambar 3.1 Siklus Sistem Informasi

III.1.2 Komponen Sistem Informasi

Komponen-komponen dari sistem informasi adalah:

1. Komponen *Input*

Input merupakan data yang masuk ke dalam sistem informasi.

2. Komponen Model

Model merupakan kombinasi prosedur, logika, dan model matematik yang memproses data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *Output*

Output merupakan hasil dari sistem informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Komponen Teknologi

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

5. Komponen Basis Data

Basis data merupakan kumpulan data yang saling berhubungan yang tersimpan didalam komputer dengan menggunakan perangkat lunak *database*.

6. Komponen Kontrol

Kontrol merupakan pengendalian yang dirancang untuk menanggulangi gangguan terhadap sistem informasi.

III.2 Sistem Informasi Berbasis Web

III.2.1 Web Server

Web server adalah perangkat lunak yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *browser* web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Cara kerja web server merupakan mesin dimana tempat aplikasi atau perangkat lunak beroperasi dalam mendistribusikan web *page* ke *user*, tentu saja sesuai dengan permintaan *user*.

Hubungan antara web server dan *browser* internet merupakan gabungan atau jaringan komputer yang ada di seluruh dunia. Setelah terhubung secara fisik, *protocol* TCP/IP (*networking protocol*) yang memungkinkan semua komputer dapat berkomunikasi satu dengan yang lainnya. Pada saat *browser* meminta data web *page* ke *server* maka instruksi permintaan data oleh *browser* tersebut di kemas di dalam TCP yang merupakan *protocol* transport dan dikirim ke alamat yang dalam hal ini merupakan *protocol* berikutnya yaitu *Hyper Text Transfer Protocol* (HTTP). HTTP ini merupakan *protocol* yang digunakan dalam *World Wide*

Web (WWW) antar komputer yang terhubung dalam jaringan di dunia ini. Untuk mengenal *protocol* ini jelas sangat mudah sekali dimana setiap kali anda mengetik `http://` anda telah menggunakannya, dan membawa anda ke dunia internet. Data yang di *passing* dari *browser* ke *web server* disebut sebagai *HTTP request* yang meminta *web page* dan kemudian *web server* akan mencari data *HTML* yang ada dan di kemas dalam *protocol* *TCP* dan di kirim kembali ke *browser*. Data yang dikirim dari *server* ke *browser* disebut sebagai *HTTP response*. Jika data yang diminta oleh *browser* tidak ditemukan oleh *web server* maka akan menimbulkan *error* yang sering anda lihat di *web page* yaitu *Error: 404 Page Not Found*.

III.2.2 Web Browser

Web browser adalah sebuah aplikasi perangkat lunak yang memungkinkan pengguna untuk menayangkan dan berinteraksi dengan tulisan, gambar, video, musik dan berbagai informasi lainnya yang terdapat pada halaman web di sebuah situs di *World Wide Web* atau di jaringan *LAN* lokal. Tulisan dan gambar dihalaman web dapat mempunyai *hyperlinks* ke halaman web lain di mesin yang sama atau di situs

web lainnya. Web *browser* memungkinkan pengguna secara cepat dan mudah mengakses informasi yang diberikan oleh banyak situs web dengan cara menjelajahi *link* tersebut. Web *browser* memformat informasi HTML untuk ditayangkan, oleh karena itu penampakan halaman web akan agak berbeda dari satu *browser* ke *browser* yang lain.

III.2.3 Web Statis

Web Statis adalah web yang *content* atau isinya tidak berubah-ubah. Maksudnya adalah isi dari dokumen yang ada di web tersebut tidak dapat diubah secara mudah. Ini dikarenakan karena *script* yang digunakan untuk membuat web statis tidak mendukung untuk mengubah isi dokumen. Karena *script* yang digunakan untuk membuat web statis ini seperti HTML dan *Cascading Style Sheet* atau biasa disebut dengan CSS. Maka dari itu untuk perubahan isi dokumen pada web statis harus mengubah isi file HTML atau CSS tersebut. Tentunya bagi orang awam atau orang yang tidak mengerti soal program atau *script* HTML dan CSS akan sangat sulit sekali dilakukan. Untuk itu jika ingin mengubah dokumen web statis harus mengerti benar *script* HTML atau

CSS atau sering disebut *programmer* yang dapat melakukan perubahan isi dokumen sebuah web statis. Web statis ini biasanya digunakan oleh *website* yang menggunakan HTML, *Web Search Engine*, atau web *Company Profile*.

III.2.4 Web Dinamis

Web dinamis adalah web yang *content* atau isinya dapat berubah-ubah setiap saat. Karena dalam teknologi pembuatan web dinamis sudah dirancang semudah mungkin bagi user yang menggunakan web dinamis tersebut. Untuk perubahan *content* atau isi dokumen dalam sebuah web dinamis terbilang mudah ketimbang web statis yang harus memiliki keahlian khusus pada bagian *scripting* web tersebut. Ketika kita akan mengubah *content* atau isi dari sebuah web dinamis kita hanya perlu masuk kebagian *control panel* atau bagian *administrator* web yang telah disediakan oleh *script* web dinamis.

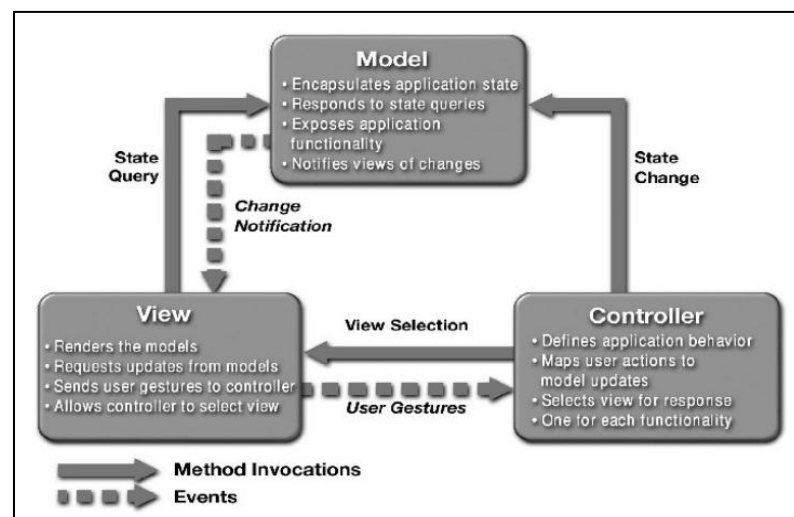
Jadi untuk pengubahan *content* atau isi dokumen dalam sebuah web dinamis tidak perlu memiliki keahlian *programming* atau seorang *programmer* saja yang dapat mengubah isi dokumen pada web dinamis. Untuk membuat web dinamis diperlukan beberapa

komponen yaitu *client side scripting* (HTML, *javascript*, *Casing Style Sheet* atau CSS) dan *server side scripting* seperti PHP dan program basis data seperti *database* MySQL untuk menyimpan data-data yang ada di web dinamis.

III.3 Arsitektur MVC (Model View Control)

Arsitektur *Model-View-Controller* adalah sebuah pola yang terbukti membangun proyek secara lebih efektif. Hal itu dilakukan dengan memilah komponen antara model, *view* dan *controller* pada bagian-bagian dalam proyek. *Layer* model yang merepresentasikan data yang digunakan oleh aplikasi sebagaimana proses bisnis yang diasosiasikan terhadapnya. Dengan memilahnya sebagai bagian terpisah, seperti penampungan data, *persistence*, serta proses manipulasi, terpisah dari bagian lain aplikasi. *Layer view* mengandung keseluruhan detail dari implementasi *user interface*. Disini komponen grafis menyediakan representasi proses internal aplikasi dan menuntun alur interaksi *user* terhadap aplikasi. Tidak ada *layer* lain kecuali *layer view* yang berinteraksi dengan *user*. *Layer Controller*

menyediakan detail alur program dan transisi *layer*, dan juga bertanggung jawab akan penampungan *events* yang dibuat oleh *user* dari *view* dan melakukan *update* terhadap komponen model menggunakan data yang dimasukkan oleh *user*. Diagram dibawah ini menunjukkan tiga komponen yang terdapat dalam pola MVC dan interaksi yang terjadi.



Gambar 3.2 Arsitektur (MVC) Model View Control

III.4 *Scripts* dan PHP

III.4.1 *Script*

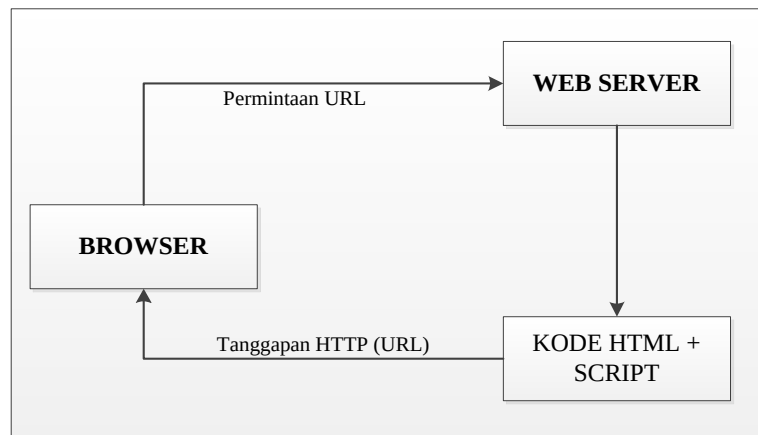
Script adalah sekumpulan instruksi pada halaman web baik pada web browser (*client-side scripting*) maupun pada server (*server-side scripting*). *Scripts* biasanya menyediakan fungsi untuk perpindahan

halaman atau tampilan halaman web yang kita akses secara interaktif ataupun secara otomatis, sehingga dapat memberikan tampilan dan fungsionalitas yang lebih.

Semua fungsi *login* sistem, pilihan menu, dan hampir semua fitur *slideshow* foto atau gambar yang ada pada halaman web menggunakan *scripts*. *Script* merupakan hal penting pada pemrograman web untuk keamanan, membuat tampilan yang interaktif dan menarik, serta mempermudah penggunaan web.

III.4.2 Client-side Scripting

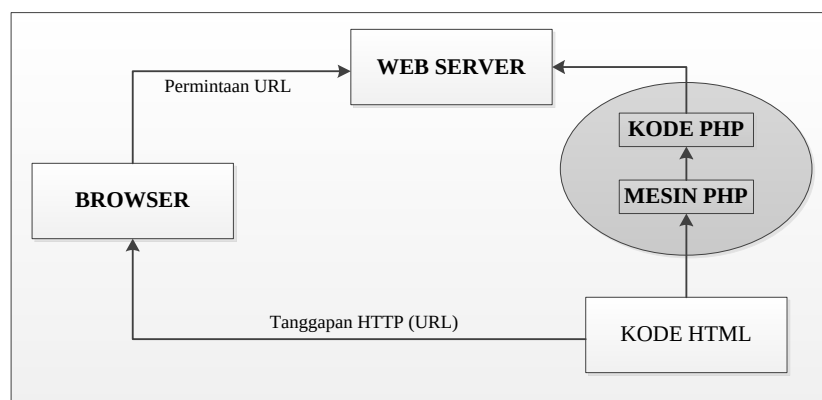
Client Side Scripting merupakan *script* yang berjalan di sisi *client/browser*, contohnya adalah HTML, XHTML, JavaScript, CSS, dan VBScript. Cara kerja *Client Side Scripting* itu sangat sederhana, yaitu berawal dari *client* yang memanggil berdasarkan URL (*Uniform Resource Locator*) melalui *browser*, kemudian *browser* mendapat alamat dari web *server*, yang nantinya akan memberikan segala informasi yang dibutuhkan web *browser*. Web *browser* yang sudah mendapat informasi segera melakukan proses penterjemahan kode HTML (*Client Side Scripting*) dan menampilkannya ke layar pemakai.



Gambar 3.3 Cara Kerja *Client-side Scripting*

III.4.3 *Server-side Scripting*

Server Side Scripting hanya perlu penterjemahan khusus untuk kode-kode, seperti PHP yang nantinya akan diterjemahkan oleh mesin PHP ke kode HTML terlebih dahulu sebelum diterjemahkan browser untuk ditampilkan dilayar *client*. Untuk lebih jelasnya lihat skema yang ada untuk membandingkan konsep kerja HTML dengan PHP.



Gambar 3.4 Cara Kerja *Server-side Scripting*

III.4.4 Bahasa Pemrograman PHP

Bahasa pemrograman PHP adalah contoh dari *server-side scripting*. PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML, sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server-side* berarti pengerjaan skrip akan dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser*. Keunggulannya dari sifatnya yang *server-side* tersebut antara lain:

Tidak diperlukan kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena *server*-lah yang akan mengerjakan *script* PHP. Hasil yang dikirimkan kembali ke *browser* umumnya bersifat teks atau gambar saja, sehingga pasti dikenal oleh *browser* apapun. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, misalnya koneksi ke *database*. *Script* tidak dapat "diintip" dengan menggunakan fasilitas *view HTML source*.

III.5 Codeigniter dan JQuery

III.5.1 Framework

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau 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. *Framework* juga dapat diartikan sebagai sebuah *tools* atau alat yang berfungsi untuk membantu sebuah pekerjaan. Karena PHP adalah *script* untuk membuat *website*, maka *framework* PHP dalam kasus ini bisa diartikan sebagai alat yang berfungsi untuk mempermudah proses pembuatan sebuah *website*.

III.5.2 Framework Codeigniter

Codeigniter adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model*, *View*, *Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. Codeigniter memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. Codeigniter dirilis pertama kali pada 28 Februari

2006. Versi terakhir adalah 2.1.2 dapat dilihat di *website* resminya (diakses pada 2 Februari 2013).

III.5.3 Kelebihan Codeigniter

Ada beberapa kelebihan codeigniter (CI) dibandingkan dengan *Framework* PHP adalah:

1. Performa sangat cepat

Salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scratch*, tapi codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding *framework* yang lain.

2. Konfigurasi yang sangat minim (*nearly zero configuration*)

Tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan codeigniter dengan *setting* standard, anda hanya perlu merubah sedikit saja *file* pada *folder config*.

3. Banyak komunitas

Dengan banyaknya komunitas codeigniter ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.

4. Dokumentasi yang sangat lengkap

Setiap paket instalasi codeigniter sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

III.5.4 Framework JQuery

Jquery adalah *framework javascript* siap pakai, sehingga mempermudah dan mempercepat kita membuat kode *javascript*. Secara umum, apabila kita membuat kode *javascript* maka diperlukan kode yang sangat panjang, bahkan sangat sulit dipahami. Dengan bantuan jquery kita bisa langsung memanggil fungsi *javascript* yang terdapat dalam *library* jquery yang sesuai dengan kebutuhan. Inti dari jquery adalah menyederhanakan kode *javascript*. Hal ini sesuai dengan selogannya "*Write Less, do More*", cukup menulis sedikit dapat melakukan banyak hal.

Jquery pertama kali dirilis pada tahun 2006 oleh John Resig, jquery telah banyak mencuri

perhatian pada pengembang web. Buktinya pada tahun-tahun berikutnya setelah jquery dirilis telah banyak digunakan oleh *website-website* terkemuka di dunia seperti Google, Microsoft, Intel, Facebook, Amazon, youtube dan masih banyak lagi.

III.6 Database Management System (DBMS)

Sistem manajemen *database* atau *database management system* (DBMS) adalah merupakan suatu sistem perangkat lunak yang memungkinkan seorang *user* dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. *Database* sendiri adalah sekumpulan data yang berhubungan dengan secara logika dan memiliki beberapa arti yang saling berpautan. Contoh SDBD adalah Oracle, SQL server 2000/2003, MS Access, MySQL dan sebagainya. DBMS yang digunakan oleh penulis adalah MySql, dengan GUI yang digunakan untuk membantu adalah MySql Workbench.

III.7 Eclipse IDE

Integrated Development Environment (IDE) adalah perangkat lunak komputer yang memiliki beberapa

fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan disemua *platform* (*platform-independent*). Berikut ini adalah sifat-sifat dari Eclipse antara lain:

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan MacOSX.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse dapat digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, uji perangkat lunak, pengembangan web dan sebagainya.

Eclipse pada saat ini merupakan salah satu IDE populer dikarenakan gratis dan *opensource*, yang berarti setiap orang dapat melihat dan memodifikasi *source code* perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plugin*.

Eclipse awalnya dikembangkan oleh perusahaan IBM (*International Business Machines*) untuk menggantikan perangkat lunak IBM *Visual Age for Java 4.0*. Produk ini diluncurkan oleh IBM pada tanggal 15 November 2001, yang menginvestasikan sebanyak 40 juta *Dollar* Amerika untuk pengembangannya. Semenjak itu konsorsium *Eclipse Foundation* mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya. Sejak tahun 2006 *Eclipse Foundation* secara rutin merilis versi Eclipse setiap tahun. Setiap rilis kemudian diberi kode sesuai dengan berbagai nama bintang dari planet Jupiter. Diantaranya adalah sebagai berikut:

1. Callisto (26 Juni 2006)
2. Europa (27 Juni 2007)
3. Ganymede (25 Juni 2008)
4. Galileo (24 Juni 2009)

5. Helios (23 Juni 2010)

6. Indigo (26 Juni 2012)

7. Juno (27 Juni 2012)

8. Kepler (27 juni 2013)

Disamping itu Eclipse juga didistribusikan dalam beberapa proyek sesuai kebutuhan spesifik:

1. Sebagai Java IDE (fungsi utama)

2. C++ IDE

3. Java *mobile/embedded device* IDE

4. Web *development*, dan sebagainya