BAB II
LANDASAN TEORI

2.1 Sistem Informasi

Sistem adalah sekelompok elemen–elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan. (Yakub, 2012).

Sistem yang dikemukakan oleh Azhar Susanto (2013) adalah sebagai berikut: Sistem adalah kumpulan atau group dari sistem atau bagian atau komponen apapun baik phisik atau pun non phisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.

Sedangkan menurut Yakub (2012), yang mengartikan bahwa, Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu.

2.2 CRM

Customer relationship management (CRM) adalah suatu pendekatan pelayanan kepada konsumen yang berfokus pada pembangunan jangka panjang dan hubungan konsumen yang berkelanjutan yang dapat memberikan nilai tambah bagi pelanggan maupun perusahaan (Jurnal Informatika, Vol. 12, No. 2, Desember 2012).

2.3 *Supply Chain Management*

Supply Chain Management (SCM) dapat mengintegrasikan praktek pengelolaan lingkungan ke dalam seluruh manajemen rantai pasokan dalam rangka mencapai greener supply chain.

management dan mempertahankan keunggulan yang kompetitif dan juga untuk meningkatkan keuntungan bisnis dan tujuan pangsa pasar. (Seman 2012).

2.4 Pemesanan

Pemesanan adalah suatu aktifitas yang dilakukan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik. Menurur Kamus Besar Bahasa Indonesia yang dimaksud pemesanan adalah "proses, perbuatan, cara memesan (tempat,barang) kepada orang lain" Menurut Edwin dan Chris (2011) Pemesanan dalam arti umum adalah perjanjian pemesanan tempat antara 2 (dua) pihak atau lebih, perjanjian pemesanan tempat tersebut dapat berupa perjanjian atas pemesanan suatu ruangan, kamar, tempat duduk dan lainnya, pada waktu tertentu dan disertai dengan produk jasanya. Produk jasa yang dimaksud adalah jasa yang ditawarkan pada perjanjian pemesanan tempat tersebut, seperti pada perusahaan penerbangan atau perusahaan pelayaran adalah perpindahan manusia atau benda dari satu titik (kota) ketitik (kota) lainnya.

2.5 Penjadwalan

Menurut Arifin dan Rudyanto (2010) Penjadwalan adalah salah satu hal yang penting dalam perusahaan manufaktur. Oleh karena itu masalah penjadawalan menjadi perhatian yang serius di perusahaan.

2.6 E-CRM

Electronic Customer Relationship Management (e-CRM) merupakan aplikasi CRM yang berusaha memanfaatkan internet dan teknologinya untuk mengintegrasikan perusahaan dengan seluruh pelanggannya.(Jurnal Telematika Juli 2011).

2.7 Metode Pengembangan Sistem

Metode pengembangan sistem sangat dibutuhkan dalam perancangan sebuah sistem karena sebelum memulai pembuatan koding – koding hendaknya merancang terlebih dahulu metode pemodelan seperti apa yang harus digunakan dengan memprioritaskan ketepatan waktu selesai dan efektifitas dalam perancangan sebuah sistem.

(Pressman, 2012) Metode air terjun atau yang sering disebut metode waterfall sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan.

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu:

1. *Requirement Analisis*

   Tahap ini pengembang sistem memerlukan komunikasi yang bertujuan untuk memahami sistem yang diharapkan oleh pengguna dan batasan sistem tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *System Design*

   Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras*(hardware)* dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

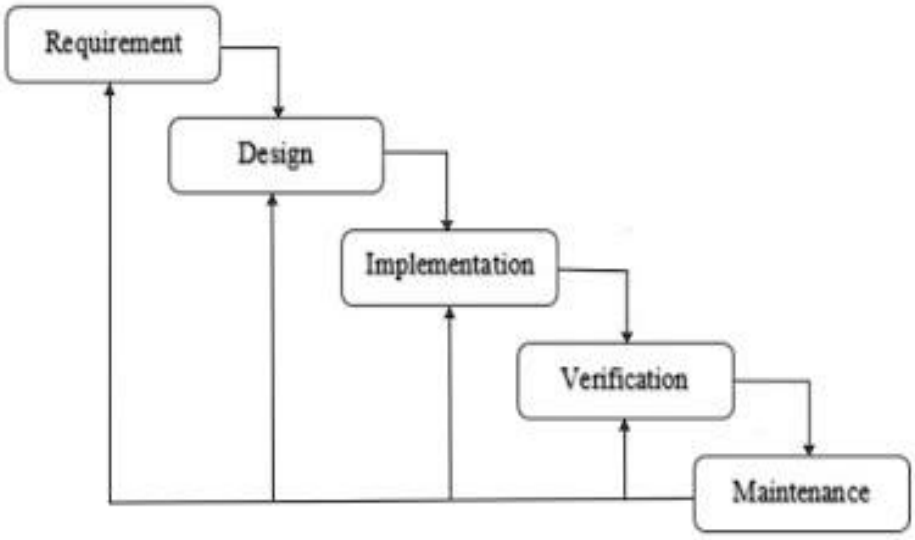
   Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

4. *Integration & Testing*

   Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. *Operation & Maintenance*

   Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.



Gambar 2.1 Tahapan waterfall

2.8 Usecase Diagram

(Analisis dan Desain Berorientasi Objek, 2016) Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yangdiberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistemdalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar

oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

Komponen-komponen yang ada dalam usecase diagram antara lain:

1. Actor

   Mempresentasikan seseorang atau sesuatu(seperti perangkat,sistem lain) yang berinteraksi dengan sistem.*Actor* hanya berinteraksi dengan use case tetapi tidak memiliki kontrol atas use case.

2. Use Case

   Adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

3. Relasi

   Merupakan hubungan yang terjadi pada sistem baik antar aktor maupun antar usecase maupun anatara usecase dan aktor. Relasi yang digunakan dalam diagram usecase antara lain:

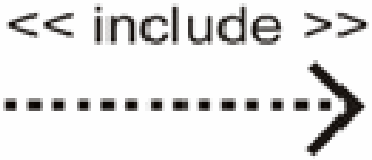
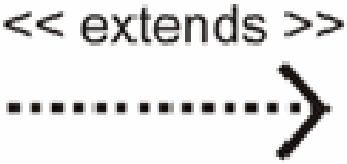
   a. Assosiation. Merupakan rlasi yang digunakan untuk menggambarkan interaksi antara usecase dan aktor. Asosiasi juga menggambarkan berapa banyak objek lain yang bisa berinteraksi dengan suatu objek atau disebut multiplicity (Multiplicity dapat dilihat pada postingan Class Diagram). Generalization. Merupakan relasi yang menggambarkan inheritance baik aktor maupun usecase.

   b. Generalization. Merupakan relasi yang menggambarkan inheritance baik aktor maupun usecase.

   c. Dependency. Merupakan relasi yang menggambarkan ketergantungan antara usecase yang satu dengan usecase yang lain. Ada dua macam dependency yaitu include dan extends. Include menggambarkan bahwa jalannya suatu usecase memicu

jalannya usecase lain. Misalnya usecase login diinclude oleh usecase memilih menu, artinya usecase memilih menu akan memicu dijalankannya usecase login. Sebelum aktor menjalankan usecase memilih menu, aktor harus menjalankan usecase login dulu. Dalam penggambaran diagram usecase, parah mengarah kepada usecase yang diinclude. Sedangkan extends menggambarkan bahwa suatu usecase dijalankan karena ada persyaratan tertentu dari usecase lain. Misal, dalam sebuah sistem user tidak bisa menjalankan login sebelum dia mendaftar akun. Dalam diagram usecase, usecase daftar akun mengextends usecase login. Artinya aktor harus menjalankan usecase daftar akun dulu sebelum menjalankan usecase login karena usecase login memiliki syarat aktor yang melakukan login harus sudah melakukan pendaftaran akun. Arah panah dependency mengarah pada usecase yang memiliki syarat.

Gambar 2.2 komponen usecase diagram

Nama Simbol	Simbol
Aktor	
Use Case	
Association Relationship	
Include Relationship	
Extend Relationship	
Generalisasi Relationship	

2.9 Class Diagram

(Analisis dan Desain Berorientasi Objek, 2016) Menggambarkan struktur statis *class* di dalam sistem. *Clas*s merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*.

Komponen-komponen yang ada pada class diagram antara lain:

1. Class

   Class adalah blok - blok pembangun pada pemrograman berorientasi obyek.Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan methodmethod dari sebuah class.

2. Association

   Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship.

3. Composition

   Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.

4. Dependency

   Kadangkala sebuah class menggunakan class yang lain. Hal ini disebut dependency. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah dependency dilambangkan sebagai sebuah panah bertitik-titik.

5. Aggregation

   Aggregation mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi.

Gsmbar 3.3 Komponen class diagram

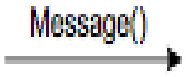
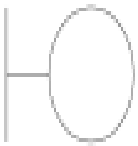
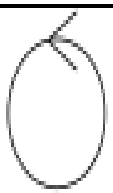
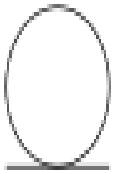
asosiasi / *association*	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
asosiasi berarah / *directed association* 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
kebergantungan / *dependency* 	relasi antar kelas dengan makna kebergantungan antar kelas
agregasi / *aggregation* 	relasi antar kelas dengan makna semua-bagian (*whole-part*)

2.10 Sequence Diagram

(Analisis dan Desain Berorientasi Objek, 2016) Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Simbol-simbol yang ada pada sequence diagram antara lain: actor, entity class, boundary class, control class, life line dan line message.

Gambar 2.4 simbol sequence diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
		*Actor*	Digunakan untuk menggambarkan user / pemgguna.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		*Boundary*	Digunakan untuk menggambarkan sebuah form.
4		*Control Class*	Digunakan untuk menghubungkan *boundary* dengan tabel.
5		*Entity Clas*	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

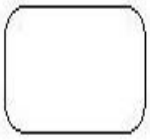
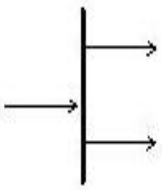
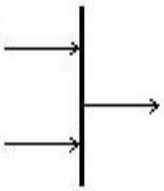
2.11 Activity Diagram

(Analisis dan Desain Berorientasi Objek, 2016) Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Simbol-simbol yang ada pada activity diagram antara lain:

1. Swimlanes, menunjukkan siapa yang bertanggung jawab melakukan aktivitas dalam suatu diagram.
2. Activities State, adalah kegiatan dalam aliran kerja.
3. Action State, adalah langkah-langkah dalam sebuah activity. Action bisa terjadi saat memasuki activity, meningggalkan activity, saat di dalam activity, atau pada event yang spesifik.
4. Business object, adalah entitas-entitas yang digunakan dalam aliran kerja.
5. Transition, menunjukkan bagaimana aliran kerja itu berjalan dari satu aktivitas ke aktivitas lainnya.
6. Decision point, menunjukkan dimana sebuah keputusan perlu dibuat dalam aliran kerja.
7. Start state, menunjukkan dimana aliran kerja itu dimulai.
8. End state, menunjukkan dimana aliran kerja itu berakhir.

Gambar 2.5 simbol Activity Diagram

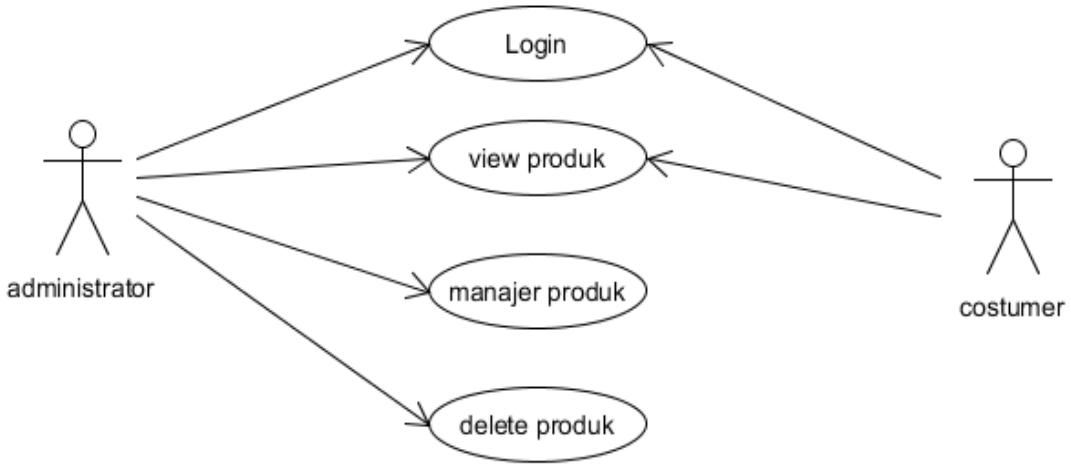
Simbol	Keterangan
	Start Point
	End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

2.12 Bahasa Pemrograman dan Perangkat Lunak Pendukung

Bahasa pemograman yang digunakan dalam penulisan skripsi ini adalah *HTML, PHP*dan *CMS* sedangkan perangkat lunak pendukung yang digunakan adalah *MySql, MyLiveChat* dan Xampp.

2.12.1 Konsep Website

World Wide Web (WWW) yang lebih dikenal dengan *website*,merupakan salah satu layanan yang dapat digunakan oleh pemakaikomputer yang terhubung pada internet. *Website* pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext* pemakai dituntut untuk menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen *website* yang ditampilkan pada *web browser*. Internet identik dengan *website*, karena popularitasnya sebagai penyedia informasi dan tampilan antar muka (*interface*) yang dibutuhkan oleh pengguna internet, dari masalah informasi sampai komunikasi.*Website* memudahkan penguna komputer untuk berinteraksi dengan pelaku internet lainnya dan menelusuri informasi. Website juga telah banyak digunakan oleh perusahaan sebagai bagian dari strategi teknologi informasinya, hal ini tidak lepas dari kelebihannya yaitu memiliki akses informasi yang mudah, *set up server* lebih mudah, informasi lebih mudah didistribusikan dan bebas *platform*. Bebas *platform* yang dimaksud adalah informasi dapat ditampilkan oleh *web browser* pada sistem operasi apa saja hal ini dikarenakan telah adanya standar dokumen berbagai tipe data dapat disajikan. *Web browser* dan *server* berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini, yaitu HTTP (*Hypertext Transfer Protocol*) bertugas menangani permintaan-permintaan dari browser untuk mengambil dokumen-dokumen website.



Gambar di atas mengambarkan transaksi antara admin dengan costumer mengunakan caseperancangan *system use case.*

2.12.2 HTML 5

HyperText Markup Language (HTML5) adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi *World Wide Web*, sebuah teknologi inti dari Internet. HTML5 adalah revisi kelima dari HTML, yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997 dan hingga bulan Juni 2011 masih dalam pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan salah satu karya Konsortium Waring Wera Wanua (*World Wide Web Consortium*, W3C) untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat *web*.

Secara khusus, HTML5 menambahkan banyak fitur baru sintaksis. Ini termasuk <*video*> baru, <*audio*> dan elemen <*canvas*>, serta integrasi *scalable vector* grafis konten (SVG) (yang menggantikan penggunaan tag generic <*object*> dan MathML untuk rumus matematika. Fitur-fitur ini dirancang untuk memudahkan untuk memasukkan dan menangani konten multimedia dan grafis di *web* tanpa harus resor untuk *plugin proprietary* dan API. Elemen baru lainnya, seperti <*section*>, <*article*>, <*header*> *dan* <*nav*>, dirancang untuk memperkaya isi semantik dokumen. Atribut baru telah diperkenalkan untuk tujuan yang sama, sementara beberapa elemen dan atribut telah dihapus. Beberapa elemen, seperti <*a*>, <*cite*> dan <*menu*> telah diubah, atau didefinisikan ulang standar. API dan dokumen model obyek (DOM) tidak *afterthoughts* lagi, tetapi bagian *fundamental* dari spesifikasi HTML5. HTML5 juga mendefinisikan secara rinci proses yang diperlukan untuk dokumen yang tidak *valid* sehingga kesalahan sintaks akan diperlakukan secara seragam oleh semua *browser conforming* dan agen pengguna lainnya.

HTML juga akan berkaitan erat dengan CSS (*Cascading Style Sheet*). CSS menentukan *format* dari konten. Dengan adanya CSS, konten *website* bisa lebih seragam sehingga hasilnya tampak cantik, dan *format* ini dapat diatur dengan cara yang berbeda (Zaki dkk, 2012). Berbicara Mengenai HTML tidak terlepas dari javascript, yang merupakan bahasa pemprograman *client-side* untuk memberikan efek dinamis. Javascript mampu membuat *web* terlihat lebih dinamis dan interaktif. Maka dari itu HTML tidak berdiri sendiri, Sebuah halaman *web* biasanya hasil kolaborasi antara HTML, CSS dan Javascript.

HTML5 sendiri sampai saat ini masih terus dikembangkan, tapi mayoritas *browser modern* sudah bisa mendukung HTML5. Ada beberapa aturan yang diterapkan untuk HTML5 (Zaki dkk, 2012),
seperti :

a. Fitur-fitur baru harus berbasis HTML, CSS, DOM, dan Javascript.
b. Mengurangi kebutuhan *plugin* eksternal (contohnya Flash).
c. *Error handling* yang lebih baik.
d. *Markup* tambahan untuk menggantikan *scripting*.
e. HTML5 harus bisa diakses dari peranti manapun

Sintaks untuk HMTL5 sangat fleksibel, namun bukan berarti anda bisa menulis sembarangan. HTML5 mendukung pemisahan yang tegas antara tahapan presentasi dan konten.
Beberapa fitur baru di HTML5 (Zaki dkk, 2012) antara lain :

a. Elemen *canvas* untuk menggambar.
b. Elemen *audio* dan *video* untuk memutar *audio* dan *video*.
c. Dukungan untuk penyimpanan *local* atau *offline*
a. Elemen yang spesifik terhadap konten, seperti *article, footer, header, nav* dan *section.*
b. Kontrol *form* baru, seperti *calendar, date, time, email, url* dan *search*.

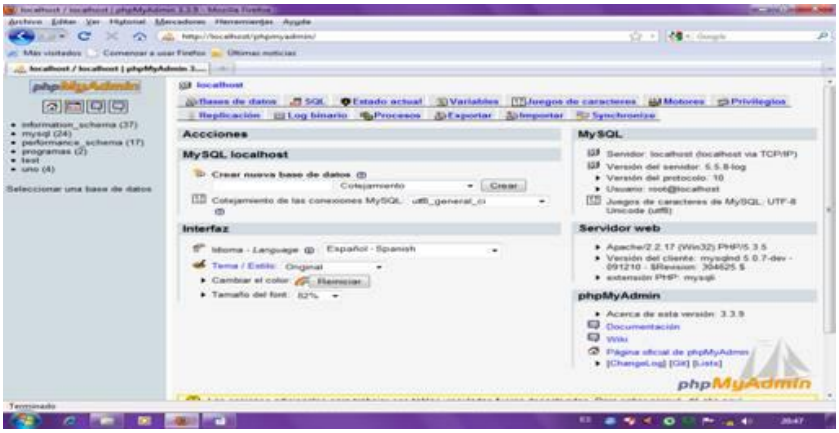
2.12.3 PHP

Pengertian PHP

PHP merupakan kependekan dari kata *Hypertext Preproessor.* PHP tergolong sebagai perangkat lunak open sourceyang diatur dalam aturan *General Purpose Licences* (GPL), serta dapat di-download bebas dari situs resminya (**http://www.php.net**).
Pemrograman php sangat cocok dikembangkan dalam lingkungan web, karena PHP dilekatkan pada *script* HTML atau sebaliknya.PHP dikhususkan untuk pengembangan 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(contohnya penggunaan browser Mozilla Firefox, Opera, Internet Explorer dan lain-lain). Pada umumnya, pembuatan web dinamis berhubungan erat dengan *database* sebagai sumber data yang akan ditampilkan.

PHP tergolong juga sebagai bahasa pemrograman yang berbasis *server* (*Server Side Scripting*). Ini berarti bahwa semua script PHP diletakkan diserver dan diterjemahkan oleh *web server* terlebih dahulu, kemudian hasil terjemahan itu dikirim ke browser *client*. Tentu hal tersebut berbeda dengan *java script*. (**Dodit, 2008**).



Gambar 2.6 , tampilan dari Php mysql.

2.12.4 Kelebihan PHP

Diantara maraknya pemrograman server web saat ini, adalah ASP yang berkembang menjadi ASP.NET, JSP, CFML, dan PHP. Jika dibandingkan di antara tiga terbesar pemrograman server web di atas, terdapat kelebihan dari PHP itu sendiri, yaitu :

1. PHP merupakan sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaanya. Tidak seperti hanya bahasa pemrograman aplikasi seperti Visual Basic dan sebagainya.

2. PHP dapat berjalan pada web server yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada apache yang bersifat *open source*.

3. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis dan *developer* membantu pengembangannya.

4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.

5. PHP dapat berjalan pada tiga sistem operasi, yaitu : Linux, Unix, dan Windows, dan juga dapat dijalankan secara runtime pada suatu *console*.

2.12.5 Sintaks PHP

Sintaks program/script PHP ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok script PHP:

<?php ... ?>
<script language = ”PHP”> ... </script>
< ? ... ?> <% … %>

2.12.6 CSS 3

DES *Cascading Style Sheets* (CSS) merupakan *feature* yang sangat penting dalam membuat Dynamic HTML. Meskipun bukan merupakan suatu keharusan dalam membuat web, akan tetapi penggunaan *cascading style sheets* merupakan kelebihan tersendiri. Suatu cascading style sheet merupakan tempat dimana mengontrol dan mengatur style-style yang ada. *Cascading Style sheet* mendeskripsikan bagaimana tampilan dokumen HTML di layar. Dalam pemakaian umumnya sering disebut juga sebagai template dari dokumen HTML yang menggunakanya. *Cascading Style Sheet* (CSS) teknologi yang support pada hampir semua

web Browser, hal ini disebabkan CSS telah di standartkan oleh *World Wide Web Consortium* (W3C) untuk digunakan web browser. (Nurhasyim,2003).

2.12.7 MySQL

MySQL adalah *Relational Databases Manajemen System* (RDBMS) yang didistribusikan gratis dibawah lisensi GPL (*General Public Licence*). Tidak sama dengan proyek-proyek seperti Apache dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya dimiliki oleh kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson dan Michael "*Monty*" Widenius. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *databases* sejak lama yaitu SQL (Structural Query Language), SQL adalah sebuah konsep pengoperasian databases terutama untuk pemilihan atau seleksi pemasukkan data seleksi dari pemasukkan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. adalah bahasa standar yang digunakan untuk mengakses server database . (http://www.mysql.com/about)

(Arief, 2011) MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya.

(Wicaksono, 2008) menjelaskan bahwa "XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MYSQL di komputer lokal". XAMPP berperan sebagai *server web* pada komputer lokal.XAMPP juga dapat disebut sebuah *Cpanel server virtual,* yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

2.12.8 Analisis

Pada tahap ini yang dilakukan adalah menganalisis kebutuhan sistem. Proses pengumpulan kebutuhan dilakukan untuk menspesifikasian kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user.*

2.12.9 Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

2.12.10 Pengkodean

Hasil dari tahap ini adalah translasi desain ke dalam bentuk program perangkat lunak yang didapatkan dari pembuatan program di komputer.

2.12.11 Pengujian

Tahap ini berfokus pada perangkat lunak secara logic dan fungsional. Selain itu, memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan diinginkan.

2.12.12 Pemeliharaan

Sebuah perangkat lunak memungkinkan mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.