



BAB II TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Mulyanto (2009:44), “Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan terorganisasi agar secara otomatis menerima dan menyimpan input, memproses, dan menghasilkan output di bawah pengawasan dari instruksi –instruksi program yang tersimpan dalam memori”.

Pangera (2010:1), “Komputer adalah perangkat elektronik, beroperasi dibawah perintah pengendali yang disimpan pada memori komputer, dapat menerima data, memproses data berdasarkan aturan tertentu, mencetak hasilnya dan menyimpan data untuk penggunaan dimasa depan.

2.1.2 Pengertian Aplikasi

Phoenix (2009:81), “ Aplikasi adalah program komputer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu ”.

2.1.3 Pengertian Pengolahan

Phoenix (2009:107), “Pengolahan adalah proses, perbuatan, cara mengolah”.

2.1.4 Pengertian Data

Kristanto (2008:7) “Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata”.

Phoenix (2009:296), “Data adalah keterangan yang benar dan nyata”.

2.1.5 Pengertian Aplikasi Pengolahan Data

Kristanto (2008:8), “Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

Ladjamudin (2013:9), “ Pengolahan data adalah waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

2.1.6 Pengertian Sistem

Hartono (2013:9), “Sistem adalah Hubungan fungsional yang terorganisasi atau teratur, yang berlangsung diantara bagian-bagian atau elemen-elemen”.

Kristanto (2008:1), “Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.

2.1.7 Karakteristik Sistem

Al Fatta (2011:5), “Suatu sistem mempunyai beberapa karakteristik, yaitu, komponen atau elemen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), sasaran (*objective*) atau tujuan (*goal*)”.

1. Komponen Sistem

Suatu sistem tidak berada dalam lingkungan yang kosong, tetapi sebuah sistem berada dan berfungsi di dalam lingkungan yang berisi sistem lainnya.

2. Batas Sistem (*boundary*)

Batas sistem merupakan pembatas atau pemisah antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*environment*)

Lingkungan luar adalah apa pun di luar batas dari sistem yang dapat memengaruhi operasi sistem, baik pengaruh yang menguntungkan ataupun yang merugikan.

4. Penghubung Sistem (*interface*)

Penghubung merupakan hal yang sangat penting, sebab tanpa adanya penghubung, sistem akan berisi kumpulan subsistem yang berdiri sendiri dan tidak saling berkaitan.

5. Masukan Sistem (*input*)

Masukan atau input merupakan energi yang dimasukkan ke dalam sistem.

6. Keluaran Sistem (*output*)

Keluaran (*output*) merupakan hasil dari pemrosesan.

7. Pengolah Sistem (*process*)

Pengolahan sistem (*process*) merupakan bagian yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan.

8. Sasaran Sistem

Suatu sistem pasti memiliki sasaran (*objective*) atau tujuan (*goal*).

2.1.8 Klasifikasi Sistem

Al Fatta (2011:5), “Sistem dapat diklasifikasikan dari berbagai sudut pandang”, di antaranya adalah sebagai berikut:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak (*abstract system*) adalah sistem yang berupa pemikiran atau gagasan yang tidak tampak secara fisik.

2. Sistem Alamiah dan Sistem Buatan

Sistem alamiah (*natural system*) adalah sistem yang terjadi karena proses alam, bukan buatan manusia.

3. Sistem Tertentu dan Sistem Tak Tentu

Sistem tertentu (*deterministic system*) yaitu sistem yang operasinya dapat diprediksi secara cepat dan interaksi di antara bagian-bagiannya dapat dideteksi dengan pasti.

4. Sistem Tertutup dan Sistem Terbuka

Sistem tertutup (*closed system*) yaitu sistem yang tidak berhubungan dengan lingkungan di luar sistem.

2.1.9 Pengertian *Internet*

Sibero (2013:10), “*Internet* adalah jaringan komputer yang menghubungkan antar jaringan secara *global*, *internet* dapat juga disebut jaringan dalam suatu jaringan yang luas.

Phoenix (2009:543), “*Internet* adalah jaringan komunikasi elektronik yang menghubungkan jaringan komputer dan fasilitas komputer yang terorganisasi di seluruh dunia melalui telepon atau satelit”.

2.2 Teori Judul

2.2.1 Pengertian Sistem

Hartono (2013:9), “Sistem adalah Hubungan fungsional yang terorganisasi atau teratur, yang berlangsung diantara bagian-bagian atau elemen-elemen”.

Kristanto (2011:1), “Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.

2.2.2 Pengertian Informasi

Sibero (2013:10), “Informasi adalah Sekumpulan data yang memiliki maksud dan tujuan serta dapat memberikan keterangan akurat yang diperlukan dalam pengambilan keputusan”.

Hartono (2013:15), “Informasi adalah Data yang telah diolah menjadi suatu bentuk yang berguna bagi penerimanya dan memiliki nilai bagi pengambilan keputusan saat ini atau di masa yang akan datang”.

2.2.3 Pengertian Sistem Informasi Geografis

Budiyanto (2005:3), “Sistem Informasi *Geografis* adalah sebuah rangkaian sistem yang memanfaatkan teknologi digital untuk melakukan analisis spasial.



2.2.4 Pengertian Data

Mulyanto (2009:15), “Data merupakan representasi dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya”.

Phoenix (2009:296), “Data adalah keterangan yang benar dan nyata”.

2.2.5 Pengertian Kependudukan

Menurut Undang-Undang Republik Indonesia No 10 Tahun 1992 tentang Perkembangan Kependudukan dan Pembangunan Keluarga Sejahtera “Kependudukan adalah hal ihwal yang berkaitan dengan jumlah, ciri utama, pertumbuhan, persebaran, mobilitas, penyebaran, kualitas, kondisi kesejahteraan yang menyangkut politik, ekonomi, sosial, budaya, agama serta lingkungan penduduk tersebut”.

2.2.6 Pengertian Warga Negara

Menurut Kamus Besar Bahasa Indoensia (2009:1268) “Warga adalah anggota (keluarga, perkumpulan, dan sebagainya)

2.2.7 Pengertian Warga Negara Indonesia

Menurut Undang-Undang Republik Indonesia Nomor 23 Tahun 2006 tentang Administrasi Kependudukan “Warga Negara Indonesia adalah orang-orang basang indonesia asli dan bangsa lain yang disahkan undang-undang sebagai warga negara”.

2.2.8 Pengertian Sistem Informasi Geografis Warga Negara Indonesia (WNI) pada Dinas Kependudukan dan Pencatatan Sipil Kabupaten Ogan Komering Ulu

Pengertian Sistem Informasi *Geografis* Warga Negara Indonesia (WNI) pada Dinas Kependudukan dan Pencatatan Sipil Kabupaten Ogan Komering Ulu adalah suatu sistem informasi yang dibuat untuk mempermudah *user* dalam proses pengolahan data kependudukan Warga Negara Indonesia (WNI) Kabupaten Ogan Komering Ulu dengan menggunakan pemrograman *PHP* dan *database MySQL*.

2.2.9 Pengertian Web

Sibero (2013:11), “Web adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan *internet*”.

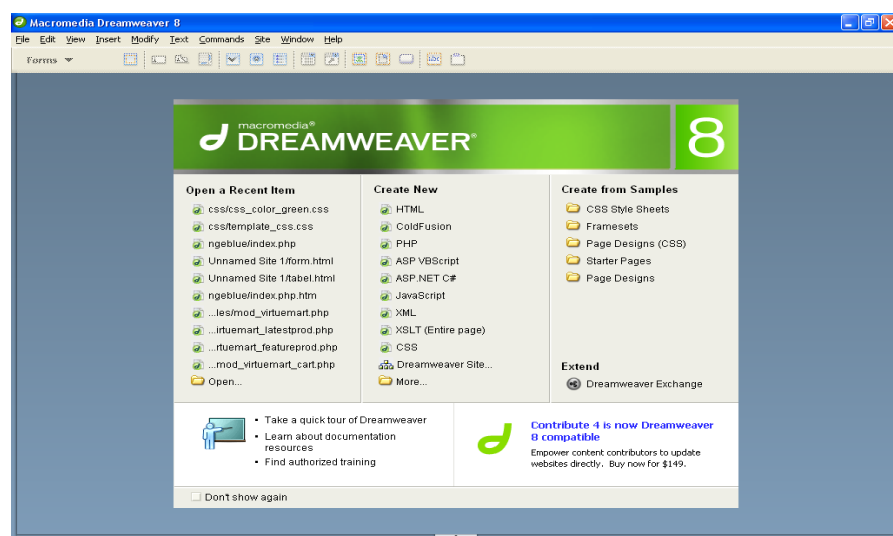
Phoenix (2009:1560), “Web adalah sistem untuk mengakses, memanipulasi, dan mengunduh hipertaut yang terdapat di komputer yang di hubungkan melalui internet, jejaring, jaringan”.

2.3 Teori Program

2.3.1 Pengenalan Dreamweaver 8

Nugroho (2009:1) *Dreamweaver 8* merupakan suatu bentuk program editor web yang dibuat oleh Macromedia. *Dreamweaver 8* selain sebagai editor yang komplet juga dapat digunakan untuk membuat animasi sederhana yang berbentuk layer dengan bantuan *JavaScript* yang didukungnya.

Sebagai editor, *Dreamweaver 8* mempunyai sifat WYSIWYG dibaca (wai-si-wig) yang artinya apa yang kita lihat pada halaman desain, maka semuanya itu akan kita peroleh pada *browser*. Dengan kelebihan ini sehingga seorang *programmer* (pembuat program) atau *desainer* (pembuat desain web) dapat langsung melihat hasil buatannya tanpa harus membuka pada *browser* (aplikasi pengakses web seperti *Internet Explorer*, *Mozilla*, dll). Adapun halaman awal dari *dreamweaver 8*, sebagai berikut :



Gambar 2.1 Halaman awal *dreamweaver 8*

Pada halaman awal *Dreamweaver 8* terdapat beberapa *menu* yang dapat dipilih :

a. *Open a Recent Item*

Pada *menu* ini akan ditampilkan beberapa *file* yang sebelumnya pernah kita buka dengan menggunakan *Dreamweaver 8*. Atau di paling bawah ada *Open* yang dapat digunakan untuk membuka *file* yang lain.

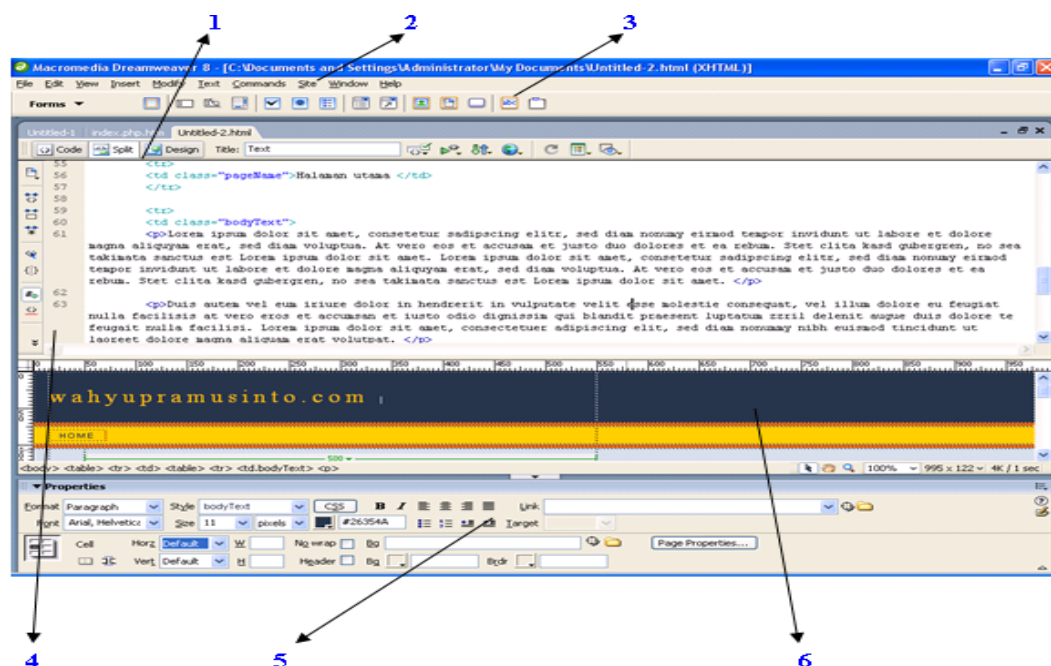
b. *Create New*

Pada *menu* ini kita dapat memilih dokumen baru apa yang akan kita buat dengan menggunakan *Dreamweaver 8*. Ada banyak pilihan, diantaranya *HTML*, *ColdFusion*, *PHP*, *ASP*, *JavaScript*, *CSS*.

c. *Create From Samples*

Pada *menu* ini kita dapat membuat *file* berdasarkan contoh yang sudah diberikan oleh *Dreamweaver*.

2.3.1.1 Menu Utama pada *Dreamweaver 8*



Gambar 2.2 Halaman *menu* utama *Dreamweaver 8*

Keterangan:

1. *Toolbar* Dokumen

Toolbar dokumen digunakan untuk mengubah tampilan dan mengakses fungsi-fungsi penting secara cepat dan mudah. Pada *toolbar* dokumen terdapat *menu* untuk berpindah antar dokumen kerja *window* dan mengatur tampilan area kerja. Untuk mengatur tampilan kita bisa memilih *Code*, *Split* dan *Design*.

2. *Menu* Utama

Menu Utama berisi semua perintah yang dapat digunakan untuk bekerja pada *Dreamweaver*.

3. *Insert Bar*

Insert bar merupakan tempat semua perangkat kerja (tombol) yang digunakan untuk membuat halaman *web*. *Insert bar* mempunyai dua jenis tampilan, yaitu tampilan sebagai *menu* dan tampilan sebagai *tab*.



Gambar 2.3 *Insert Bar* pada *Dreamweaver 8*

a. *Tab Common*

Berisi semua tombol yang sering atau umum digunakan untuk membuat halaman *web*. Tombol yang ada di *Tab Common* antara lain *Hyperlink*, *Email Link*, *Named Anchor*, *Table*, *Images*.

b. *Tab Layout*

Tab layout digunakan untuk membuat *layout* halaman *web*. Terdapat tiga jenis *layout* yang dapat dipilih, yaitu *Standard*, *Expanded* dan *Layout*.

c. *Tab Form*

Tab form digunakan untuk membuat elemen dalam *form*, misalnya saja *textarea*, *textfield*, *radio button*, *checkbox*.

d. *Tab Text*



Tab text digunakan untuk membuat pengaturan teks. Misalnya saja membuat *text italic*, *strong*, *underline*, *h1*.

e. *Tab HTML*

Tab HTML digunakan untuk membuat garis *horizontal*, menambahkan meta tag dalam tag `<head>`, dan *frame*.

f. *Tab Application*

Tab application digunakan jika aplikasi kita sudah berhubungan dengan suatu bahasa pemrograman dan sebuah *database*.

g. *Tab Flash elements*

Tab flash elements digunakan untuk memasukkan elemen *flash* dalam dokumen yang kita buat.

4. *Code View*

Code View digunakan untuk melihat kode *HTML* dari halaman *web* yang sedang kita buat.

5. *Panel Properties*

Panel properties merupakan panel yang digunakan untuk melihat dan mengubah *property* dari semua objek yang ada di area kerja. Masing-masing objek mempunyai *property* yang berbeda. Untuk melihat *property* dari objek yang diinginkan, seleksi dulu objek tersebut.

6. *Design View*

Design View digunakan untuk melihat tampilan *web* dari kode *HTML* yang kita buat.

2.3.2 *PHP (Page Hypertext Preprocessor)*

2.3.2.1 *Sejarah PHP*

Anhar (2010:2), “PHP pertama kali dibuat oleh Rasmus Lerdroft, yang beri nama FI (*Form Interpreted*), dan digunakan untuk mengolah *form* dari *web*, pada perkembangannya kode tersebut dirilis keumum sehingga mulai banyak di kembangkan oleh *programmer* diseluruh dunia”.

2.3.2.2 Pengenalan PHP

Nugroho (2009:114), “*PHP* adalah bahasa program yang berbentuk skrip yang diletakkan didalam *server web* “.

Adapun kelebihan-kelebihan dari *PHP* yaitu :

- PHP* mudah dibuat dan kecepatan akses tinggi.
- PHP* dapat berjalan dalam *web server* yang berbeda dan dalam sistem operasi yang berbeda pula.*PHP* dapat berjalan disistem operasi *UNIX*, *Windows 98*, *Windows NT* dan *Macintosh*.
- PHP* diterbitkan secara gratis.
- PHP* juga dapat berjalan pada *web server Microsoft Personal*, *Web Server*, *Apache*, *IIS*, *Xitami* dan sebagainya.
- PHP* adalah termasuk bahasa yang *embedded* (bisa ditempel atau diletakkan dalam *tag HTML*)
- PHP* termasuk *server-side programming*.

2.3.2.3 Teknik Penulisan Skrip

Nugroho (2009:114-115), Dalam membuat program *PHP*, maka yang dibutuhkan adalah memulai dengan *start tag* dan *end tag*, yaitu perintah awal dan perintah akhir.

Ada beberapa cara bagaimana untuk menuliskan *start tag* dan *end tag*:

```
<? Skrip PHP Anda disini ?>
```

```
<?php Skrip PHP Anda disini ?>
```

```
<% Skrip PHP Anda disini ?>
```

```
<SCRIPT language="php"> Skrip PHP Anda disini  
</SCRIPT>
```

Jadi, semua perintah yang diletakkan pada daerah skrip akan dianggap sebagai perintah *PHP*. Sehingga jika terjadi kesalahan ataupun kata-kata yang tidak sesuai dengan program akan dianggap salah dan akan mengakibatkan program yang kita buat menjadi *error* (salah).

Adapun kriteria yang harus diperhatikan dalam penulisan skrip adalah sebagai berikut:

- Setiap halaman yang mengandung skrip *PHP* harus disimpan dengan ekstensi *PHP* sesuai dengan program *PHP* yang mendukungnya, misalnya *nama_file.php*
- Setiap skrip *PHP* harus didahului dengan pembuka *PHP* (`<?`) dan diakhiri dengan penutup (`?>`)
- Setiap baris skrip isi harus didahului pernyataan cetak yang dibedakan menjadi dua yaitu *print* dan *echo*, adapun kriteria penulisannya adalah sebagai berikut :

```
print (" isi data ");  
echo " isi data ";
```

- Setiap akhir baris perintah harus diakhiri dengan titik koma (;).
- Setiap bentuk variabel harus diawali tanda *dollar* (\$).
- Penulisan *comment* atau keterangan didahului dengan tanda pembuka (`/*`) dan diakhiri dengan tanda (`*/`), biasanya kita menggunakan ini untuk memberikan keterangan program yang berbentuk kalimat.

2.3.3 MySQL

2.3.3.1 Pengenalan MySQL

Nugroho, (2013:1), “MySQL adalah *Relational Database Management System (RDBMS)* yaitu database relasi yang memiliki perintah standar adalah *SQL (Structured Query Language)*”.

Nugroho (2009:91), “MySQL termasuk jenis *RDBMS (Relation Database Management System)*. Sehingga istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL. Pada MySQL sebuah *database* mengandung satu atau beberapa tabel, tabel terdiri dari sejumlah baris dan kolom”.

Dalam konteks bahasa *SQL*, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur 2 dimensi yang terdiri atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column*). Baris pada tabel sering disebut sebagai *instance* dari data sedangkan kolom sering disebut sebagai *attributes* atau *field*.



Sebagai sebuah program penghasil *database*, *MySQL* tidak mungkin berjalan sendiri tanpa adanya sebuah aplikasi pengguna (*interface*) yang berguna sebagai program aplikasi pengakses *database* yang dihasilkan. *MySQL* dapat didukung oleh hampir semua program aplikasi baik yang *Open Source* seperti *PHP* maupun yang tidak *Open Source* yang ada pada *platform Windows*, seperti *Visual Basic*, *Delphi* dan lainnya.

Program-program aplikasi yang mendukung *MySQL*:

- a. *PHP (Page Hipertext Preprocessor)*
- b. *Borland Delphi, Borland C++ Builder*
- c. *Visual Basic 5.0/6.0 dan .Net*
- d. *Visual FoxPro*
- e. *Cold Fusion*

2.3.4 Pengenalan XAMPP

Nugroho (2009:74), *XAMPP* adalah paket *PHP* yang berbasis *Open Source* yang dikembangkan oleh suatu komunitas *Open Source*.

Dengan menggunakan *XAMPP* anda tidak perlu lagi bingung untuk melakukan penginstalan program-program yang lain, karena semua kebutuhan telah disediakan oleh *XAMPP*. Adapun beberapa paket yang telah disediakan sebagai berikut:

- a. *Apache HTTPD 2.0.54*
- b. *MySQL 4.1.12*
- c. *PHP 5.0.3*
- d. *FilZilla FTP Server 0.9 Beta*
- e. *phpMyAdmin 2.6.1-pl3*

2.4 Teori Khusus

Teori ini akan membahas mengenai desain sistem yang merupakan suatu fase dimana diperlukan suatu keahlian perancangan untuk elemen-elemen komputer yang akan menggunakan sistem baru ada beberapa alat bantu yang digunakan dalam desain sistem yaitu :

2.4.1 Basis Data (*Database*)

Husein (2013:411), “*Database* adalah istilah yang menyatakan kumpulan data yang disimpan dalam bentuk yang mudah diakses oleh pemakai”.

Pahlevi (2013:1), “Basis Data adalah sekumpulan data yang saling berhubungan secara logis beserta deskripsinya, yang digunakan secara bersama-sama dan dirancang untuk memenuhi kebutuhan Informasi di suatu tempat”.

2.4.2 Kamus Data (*Data Dictionary*)

Ladjamudin (2013:70), “Kamus Data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”.

Shalahuddin (2013:73), “Kamus Data (*data dictionary*) adalah kumpulan daftar elemen data yang mengalir pada sistem sehingga masukan dan keluaran dapat dipahami secara umum (memiliki standar cara penulisan)”.

2.4.3 *Even List*

Kristanto (2008:70-71), “*Event list* atau daftar kejadian digambarkan dalam bentuk kalimat sederhana dan berfungsi memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggapan atau respon dari *system*”.

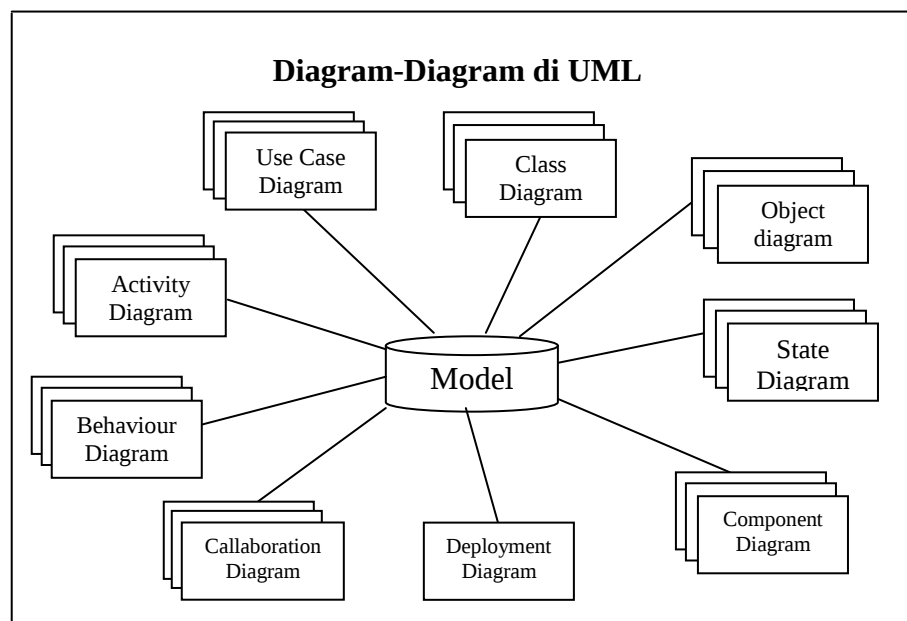
2.4.4 Diagram pada UML (*Unified Modelling Language*)

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu.

Berikut adalah diagram-diagram yang didefinisikan dalam UML :

1. *Use Case diagram*
2. *Class diagram*
3. *Behaviour diagram*
 - a. *Statechart diagram*
 - b. *Activity diagram*

4. *Interaction diagram*
 - a. *Sequence diagram*
 - b. *Callaboration diagram*
5. *Component diagram*
6. *Deployment diagram*



Gambar 2.4 Diagram-Diagram Di UML

2.4.4.1 Use Case Diagram

Widodo (2012:21), "Use Case adalah fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas."

Diagram *use case* memperlihatkan pada hubungan-hubungan yang terjadi antara aktor-aktor dengan *use case –use case* dalam sistem.

Ada beberapa simbol yang digunakan dalam *use case* antara lain :

Tabel 2.1 Simbol-Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terstruktur bagi suatu <i>actor</i>
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemenlain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah elemen-elemennya (<i>sinergi</i>)
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

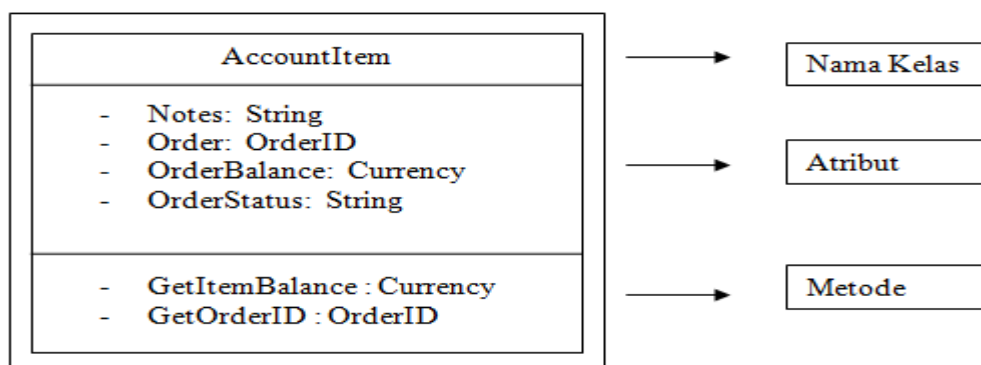
Sumber : Prabowo Pudjo Widodo. 2012. “Menggunakan UML”. Bandung: Informatika.

2.4.4.2 Class Diagram

Diagram Kelas adalah inti dari proses pemodelan objek. Widodo (2012:39), "Class diagram adalah kumpulan kelas-kelas objek. Oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas. Kelas dapat diartikan sebagai satu set objek yang memiliki atribut dan perilaku yang sama."

Class memiliki tiga area pokok :

1. Nama kelas (dan *stereotype*),
2. *Atribut*,
3. *Metode/Operas*



Gambar 2.5 Class Diagram

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

Berikut adalah hubungan antar *class* dalam *class diagram* :

1. *Asosiasi*, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui *eksistensi class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. *Agregasi*, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3. *Pewarisan*, yaitu hubungan *hierarkis* antara *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan

menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah *generalisasi*.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang dijelaskan kemudian.

2.4.4.3 Activity Diagram

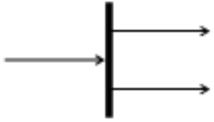
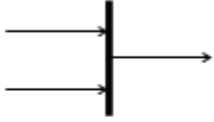
Widodo (2012:144), "Activity diagram adalah kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah lagi. Ketika melakukan diagram aktivitas untuk memodelkan perilaku suatu *classifier*, *classifier* dikatakan konteks dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku."

Activity diagram mungkin berdiri sendiri untuk memvisualisasikan, menspesifikasikan, mengkonstruksikan, serta mendokumentasikan sifat dinamis dari sekumpulan objek, atau ia dapat digunakan untuk memodelkan aliran kendali dari suatu operasi. Jika *Interaction Diagram* menekankan aliran dari kendali (*control*) dari suatu objek ke objek lainnya, maka *Activity Diagram* menekankan aliran *control* dari suatu aktivitas ke aktivitas lainnya.

Activity Diagram tidak hanya penting untuk pemodelan aspek dinamis dari sistem, tetapi juga penting untuk mengkonstruksikan sistem-sistem yang dapat dieksekusi, baik untuk *forward engineering* atau *reverse engineering*.

Aktivitas digambarkan dengan persegi panjang tumpul, Namanya ditulis dikiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.

Tabel 2.2 Simbol-Simbol Activity Diagram

No	Simbol	Keterangan
1		Start Point
2		End Point
3		Activities
4		Fork (Percabangan
5		Join (penggabungan)
6		Decision
7	Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan actor (mengelompokkan activity dalam sebuah urutan yang sama)

Sumber : Prabowo Pudjo Widodo. 2012. "*Menggunakan UML*". Bandung: Informatika.

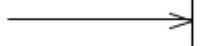
2.4.4.4 Sequence Diagram

Widodo (2012:220), "Sequence Diagram adalah *interaction diagram* yang memperlihatkan *event-event* yang beurutur sepanjang berjalannya waktu, kita membaca diagram ini dari atas ke bawah."

Sequence diagram biasa digunakan untuk mengghambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan *use case diagram*. Memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case.

Tabel 2.3 Simbol-Simbol *Sequence Diagram*

No	Simbol	Keterangan
1	 <i>An Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
2	 <i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
3	 <i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari form
4	 <i>Control Class</i>	Menggambarkan penghubung antarax boundary dengan tabel
5	 <i>A focus of control & alife line</i>	Menggambarkan tempat mulai dan berakhirnya sebuah message (pesan)
6		Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : Prabowo Pudjo Widodo. 2012. "*Menggunakan UML*". Bandung: Informatika.