



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Aplikasi

Asropudin (2013:6), “*application* atau aplikasi merupakan *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel”.

Sutarman (2012:285), “program aplikasi adalah program-program yang dibuat oleh suatu perusahaan komputer untuk para pemakai yang beroperasi dalam bidang-bidang umum, seperti toko, penerbitan, komunikasi, penerbangan, perdagangan, dan sebagainya”.

Jadi aplikasi atau program aplikasi adalah *software* atau program-program yang dibuat untuk membantu para pemakai dalam mengerjakan tugas-tugas tertentu.

2.1.2. Pengertian Data

Asropudin (2013:22), “data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi”.

Sutarman (2012:3), “data adalah fakta dari sesuatu pernyataan yang berasal dari kenyataan, dimana pernyataan tersebut merupakan hasil pengukuran atau pengamatan. Data dapat berupa angka-angka, huruf-huruf, simbol-simbol khusus, atau gabungan darinya”.

Secara keseluruhan, data merupakan kumpulan angka-angka, huruf-huruf, simbol-simbol khusus atau gabungan darinya yang belum memiliki arti sehingga dapat diolah untuk menghasilkan suatu informasi.

2.1.3. Pengertian Pengolahan Data

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



Sutarman (2012:4) mengatakan, “pengolahan data (data processing) adalah proses perhitungan/transformasi data *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan”.

Jadi, dapat disimpulkan bahwa pengolahan data adalah proses perhitungan/transformasi data menjadi informasi yang memiliki kegunaan.

2.1.4. Pengertian Pembayaran

Tim Pustaka Phoenix (2009:116) menyatakan, “pembayaran adalah perbuatan membayarkan atau membayar”.

2.1.5. Pengertian Biaya

Alwi (2007:146) menyatakan, “biaya adalah uang yang dikeluarkan untuk mengadakan (mendirikan, melakukan, dsb) sesuatu”.

2.1.6. Pengertian Perawatan

Alwi (2007:935) menyatakan, “perawatan adalah proses, cara, perbuatan merawat”, sedangkan “rawat jalan merupakan perawatan pasien dengan cara berobat jalan” dan “rawat inap merupakan perawatan pasien dengan menginap (di rumah sakit)”.

2.1.7. Pengertian Pasien

Alwi (2007:834) menyatakan, “pasien adalah orang sakit (yg dirawat dokter) atau penderita (sakit)”.

2.1.8. Pengertian Rumah Sakit

Alwi (2007:967) menyatakan, “rumah sakit adalah gedung tempat merawat orang sakit atau gedung tempat menyediakan dan memberikan pelayanan kesehatan yang meliputi berbagai masalah kesehatan”.

2.1.9. Pengertian Web

Asropudin (2013:109), “*website* adalah sebuah kumpulan halaman (*webpages*) yang diawali dengan halaman muka (*homepage*) yang berisikan informasi, iklan serta program interaksi”.



Kadir (2008:356), “aplikasi *web* atau aplikasi berbasis *web* (*web-based-application*) adalah aplikasi yang dijalankan melalui *browser*”.

Jadi, *website* atau aplikasi *WEB* adalah sebuah halaman berisikan informasi, iklan serta program interaksi yang dijalankan melalui *browser*.

2.1.10. Pengertian Internet

Asropudin (2013:48), “internet merupakan istilah umum yang dipakai untuk menunjuk *network* tingkat dunia yang terdiri dari komputer dan layanan servis atau sekitar 30 sampai 50 juta pemakai komputer dan puluhan sistem informasi termasuk *e-mail*, *Gopher*, *FTP* dan *world wide web*”.

Sutarman (2012:283), “internet adalah kumpulan dari berbagai macam jenis komputer yang saling terhubung dengan menggunakan media telekomunikasi (telepon, wireless, satelit, dan sebagainya) dengan jangkauan seluruh dunia/global”.

Jadi, internet adalah kumpulan berbagai komputer dan layanan servis yang saling terhubung menggunakan media telekomunikasi dengan jangkauan seluruh dunia/global.

2.1.11. Pengertian Judul Secara Keseluruhan

Dari uraian diatas dapat disimpulkan bahwa Aplikasi Pengolahan Data Pembayaran Biaya Perawatan Pasien pada Rumah Sakit Khusus Mata Provinsi Sumatera Selatan Berbasis *WEB* merupakan suatu aplikasi yang berfungsi untuk mengolah data pembayaran biaya perawatan pasien, baik pasien rawat jalan maupun pasien rawat inap yang berbasis *WEB* sehingga mudah diakses oleh user dengan menggunakan internet dan mempermudah dalam pembuatan laporan mengenai pembayaran pasien rawat inap dan rawat jalan pada Rumah Sakit Khusus Mata Provinsi Sumatera Selatan.

2.1.12. Metodologi Pengembangan Sistem

Sukamto dan Shalahuddin (2013:26) SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk



mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik. SDLC memiliki beberapa model dalam penerapan tahapan prosesnya, salah satunya adalah model *waterfall*.

Sukanto dan Shalahuddin (2013:26) model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

2.1.12.1. Analisis

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2.1.12.2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan 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. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

2.1.12.3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

2.1.12.4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.



2.1.12.5. Pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak 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. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.13. Studi Kelayakan

Fatta (2007:75), untuk memastikan usulan suatu proposal pendahuluan proyek bisa diteruskan menjadi proyek yang menguntungkan maka proposal proyek harus dievaluasi kelayakannya dari berbagai segi kelayakan, diantaranya:

1. Kelayakan teknis

Kelayakan teknis menyoroti kebutuhan sistem yang telah disusun dari aspek teknologi yang akan digunakan. Jika teknologi yang dikehendaki untuk pengembangan sistem merupakan teknologi yang mudah didapat, murah, dan tingkat pemakaiannya mudah, maka secara teknis usulan kebutuhan sistem bisa dinyatakan layak.

2. Kelayakan operasional

Untuk disebut layak secara operasional, usulan kebutuhan sistem harus benar-benar bisa menyelesaikan masalah yang ada di sisi pemesan sistem informasi. Disamping itu, informasi yang dihasilkan oleh sistem harus merupakan informasi yang benar-benar dibutuhkan oleh pengguna tepat pada saat pengguna menginginkannya.

3. Kelayakan ekonomi

Untuk menganalisis kelayakan ekonomi digunakan kalkulasi yang dinamakan *Cost Benefit Analysis* atau Analisis Biaya dan Manfaat. Adapun tujuan dari analisis biaya dan manfaat ini adalah untuk memberikan gambaran kepada pengguna apakah manfaat yang diperoleh dari sistem baru lebih besar dibandingkan dengan biaya yang dikeluarkan.



2.1.14. Analisis PIECES

Fatta (2007:51) untuk mengidentifikasi masalah, harus dilakukan analisis terhadap kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan. Panduan ini dikenal dengan analisis PIECES (*performance, information, economy, control, efficiency, dan services*).

Tabel 2.1. Analisis PIECES

No.	Analisis PIECES	Penjelasan
1.	Performance (Kinerja)	Kinerja diukur dengan jumlah produksi dan waktu tanggap. Jumlah produksi adalah jumlah pekerjaan yang bisa diselesaikan selama jangka waktu tertentu. Waktu tanggap adalah keterlambatan rata-rata antara suatu transaksi dengan tanggapan yang diberikan kepada transaksi tersebut.
2.	Information (Informasi)	Informasi merupakan fokus dari suatu batasan atau kebijakan. Analisis informasi memeriksa <i>output</i> sistem, analisis data, meneliti data yang tersimpan dalam sebuah sistem.
3.	Economy (Ekonomi)	Hal-hal yang perlu diperhatikan dalam analisis ekonomi adalah biaya dan keuntungan.
4.	Control (Keamanan)	Hal-hal yang perlu diperhatikan dalam analisis keamanan adalah keamanan yang lemah dan keamanan yang berlebihan.
5.	Efficiency (Efisiensi)	Efisiensi menyangkut bagaimana menghasilkan <i>output</i> sebanyak-banyaknya dengan input yang sekecil mungkin.
6.	Services (Layanan)	Terdapat beberapa kriteria penilaian dimana kualitas suatu sistem bisa dikatakan buruk, yaitu: sistem menghasilkan produk yang tidak akurat, sistem menghasilkan produk yang tidak konsisten, sistem menghasilkan produk yang tidak dipercaya, sistem tidak mudah dipelajari, sistem tidak mudah digunakan, sistem canggung untuk digunakan, dan sistem tidak fleksibel.

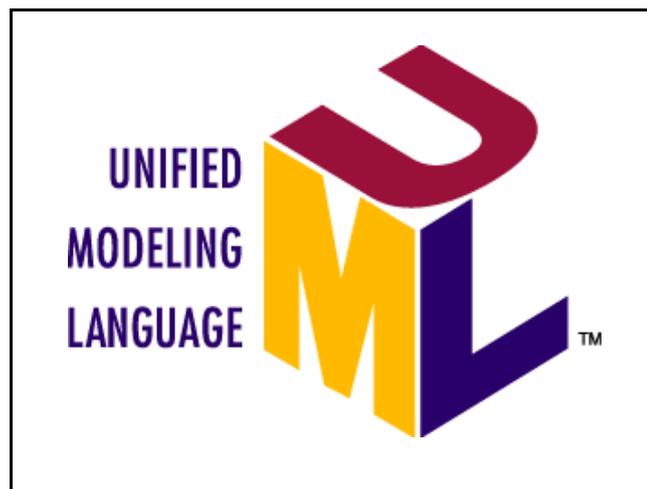


2.2. Teori Khusus

2.2.1. UML (*Unified Modeling Language*)

Sukamto dan Shalahuddin (2013:133), “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.



Gambar 2.1. Tampilan Logo *UML*

Widodo dan Herlawati (2011:6-7), *UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

**Tabel 2.2.** Tipe Diagram *UML*

No.	Diagram	Tujuan
1	<i>Class</i>	Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi
2	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen
3	<i>Use case</i>	Diagram ini memperlihatkan himpunan <i>use case</i> dan aktor-aktor (suatu jenis khusus dari kelas)
4	<i>Sequence</i>	Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu
5	<i>Communication</i>	Sebagai pengganti diagram kolaborasi <i>UML</i> 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan
6	<i>Statechart</i>	Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (<i>state</i>), transisi, kejadian serta aktivitas
7	<i>Activity</i>	Tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem
8	<i>Component</i>	Memperlihatkan organisasi serta kebergantungan sistem / perangkat lunak pada komponen-komponen yang telah ada sebelumnya
9	<i>Deployment</i>	Memperlihatkan konfigurasi saat aplikasi dijalankan (<i>run-time</i>)

Sumber: Widodo dan Herlawati (2011:10-12)

2.2.2. Jenis-Jenis Diagram *UML*

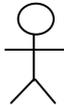
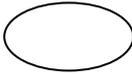
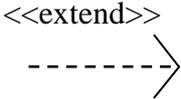
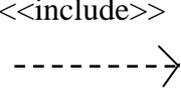
2.2.2.1. *Use case* Diagram

Sukamto dan Shalahuddin (2013:155), “*use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.3. Simbol-simbol *Use case* Diagram

No.	Simbol	Nama	Deskripsi
1		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: Sukamto dan Shalahuddin (2013:156-158)

Komponen pembentuk diagram *use case* adalah:

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use case*, aktivitas / sarana yang disiapkan oleh bisnis / sistem.
3. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

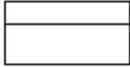
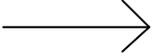
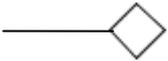
2.2.2.2. Class Diagram

Sukamto dan Shalahuddin (2013:141), “*class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi

perancangan dan perangkat lunak sinkron. Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut:

Tabel 2.4. Simbol-simbol *Class* Diagram

No	Gambar	Nama	Deskripsi
1		<i>Class</i>	Kelas pada stuktur sistem.
2		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber: Sukamto dan Shalahuddin (2013:146-147)

2.2.2.3. Activity Diagram

Sukamto dan Shalahuddin (2013:161) “*activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Adapun simbol-simbol yang digunakan dalam *activity* diagram adalah sebagai berikut:

Tabel 2.5. Simbol-simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
3		Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir	Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

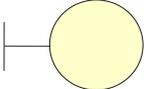
Sumber: Sukamto dan Shalahuddin (2013:162-163)

2.2.2.4. *Sequence Diagram*

Sukamto dan Shalahuddin (2013:165) , “diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

Sequence diagram menunjukkan urutan *event* kejadian dalam suatu waktu. Komponen *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. Simbol-simbol yang digunakan dalam *sequence* diagram adalah:

Tabel 2.6. Simbol-simbol *Sequence Diagram*

No	Simbol	Nama	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 <i>form</i>
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of control</i>	Menggambarkan tempat mulai dan berakhirnya sebuah <i>message</i> (pesan)
6		<i>A line of life</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

2.3. Teori Program

2.3.1. Basis Data (*Database*)

Sukamto dan Shalahuddin (2013:43), “basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Sutarman (2012:15), “*database* merupakan sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan diantaranya”.

Kadir (2008:3), “*database* berarti koleksi data yang saling terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusunan data yang terstruktur yang disimpan dalam media pengingat (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat”.



Jadi, *database* atau basis data merupakan kumpulan data-data yang saling berhubungan dan terorganisasi yang disimpan dalam media pengingat (*hard disk*) agar dapat diakses dengan mudah dan cepat.

2.3.2. PHP

2.3.2.1. Pengertian PHP

Sidik (2012:5), “*PHP* adalah kependekan dari *PHP:HyperText Preprocessor* (rekursif, mengikut gaya penamaan di *nix), merupakan bahasa utama *script server-side* yang disisipkan pada *HTML* yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi *desktop*”.

Komputer (2012:76) menjelaskan bahwa “*PHP* merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. *PHP* dapat dikatakan sebagai sebuah *server-side embedded script language*, artinya sintak-sintak dan perintah program yang Anda tulis akan sepenuhnya dijalankan oleh server tetapi dapat disertakan pada halaman *HTML* biasa”.

Jadi, *PHP (HyperText Preprocessor)* adalah bahasa pemrograman berbasis *WEB* yang dijalankan di server dan bisa digunakan untuk membuat aplikasi *desktop*.



Gambar 2.2 Tampilan Logo PHP

2.3.2.2. Script PHP

PHP adalah program yang fleksibel, artinya *script-script PHP* dapat Anda tuliskan pada sela-sela tag *HTML*. Karena *PHP* memiliki sifat yang dapat berkonteraksi dengan program lain, maka kita tidak usah bingung dalam

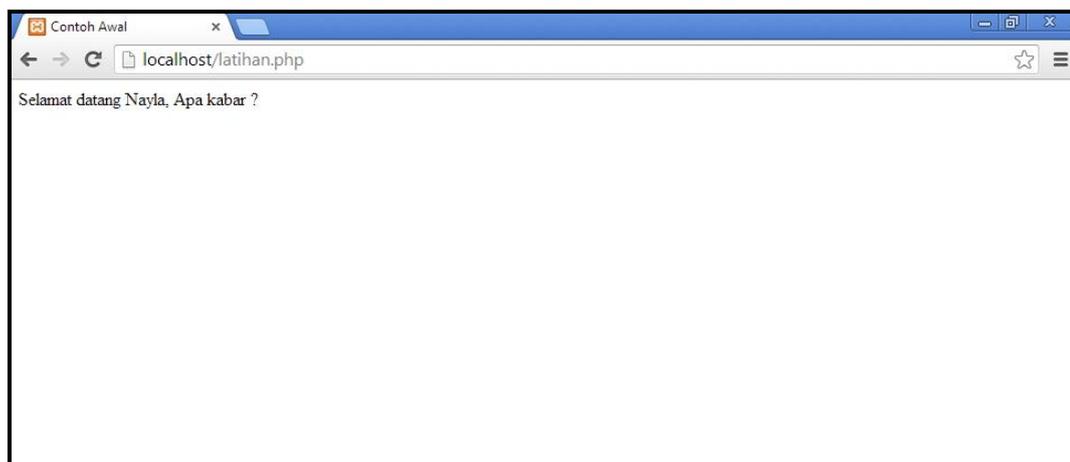


menuliskan skripnya. Jadi, kita dapat membuat sebuah program *PHP* yang di dalamnya berupa tag-tag *HTML* murni yang kemudian disisipi *script PHP*. Akan tetapi, *PHP* juga dapat berdiri sendiri sehingga tidak memerlukan *HTML* murni untuk meletakkan *script-scriptnya*.

Contoh *script*

```
<HTML>
<HEAD>
<TITLE> Contoh Awal </TITLE>
</HEAD>
<?php
    Print ("Selamat datang Nayla, Apa kabar ? ");
?>
<BODY>
</BODY>
</HTML>
```

Bila dijalankan melalui browser, maka kode tersebut akan membentuk tampilan seperti berikut:



Gambar 2.3. Contoh *Script PHP*

2.3.2.3. Teknik Penulisan *PHP*

Ada beberapa cara menuliskan *script PHP*:

- a. `<? ... Script PHP disini... ?>` atau



- b. `<?php ... Script PHP disini... ?>` atau
- c. `<% ... Script PHP disini... %>` atau
- d. `<SCRIPT language="php"> ... Script PHP disini... </SCRIPT>`

Jadi, semua kata dan *script* yang diletakkan pada daerah *script* akan dianggap sebagai perintah *PHP* sehingga jika terjadi kesalahan atau kata-katanya tidak sesuai dengan program akan dianggap salah dan akan mengakibatkan program yang kita buat menjadi *error*.

Adapun kriteria yang harus diperhatikan dalam penulisan *script PHP* adalah sebagai berikut:

- a. Setiap halaman yang mengandung *script PHP* harus disimpan dengan EXTENSI *PHP* sesuai dengan program *PHP* yang mendukungnya (nama_file.php, atau nama_file.php3, nama_file.php4).
- b. Setiap *script PHP* harus didahului dengan pembuka *PHP* (`<?php` dan lain-lain) dan kemudian diakhiri dengan penutup (`?>`).
- c. Setiap baris *script* harus didahului pernyataan cetak atau dalam program pascal kita sebut *write* atau *REM*, sedangkan dalam *PHP* pernyataan cetak dibedakan menjadi dua yaitu *Print* dan *Echo*. Adapun kriteria penulisannya sebagai berikut:

Print (“ Isi Perintah “)

Printf (“ Isi Perintah “)

Echo “ Isi Perintah “

Echo Perintah

- d. Setiap akhir baris perintah harus diakhiri dengan titik koma (;) misalnya kita akan menuliskan “Hello, Nama Saya Nayla”, maka *script* yang harus anda tuliskan adalah sebagai berikut:

<?php

Echo “Hello, Nama Saya Nayla”;

?>

- e. Semua variabel harus diberi tanda string dolar (\$) pada penulisan awalnya. Sebagai contoh, kita akan menuliskan nama sebagai variabel yang isinya Nayla, maka kita harus menuliskan sebagai berikut:



\$nama = Nayla;

- f. Penulisan *comment* atau keterangan didahului dengan pembuka `/*` dan diakhiri dengan `*/`. Biasanya kita menggunakan ini untuk memberikan *comment* yang berbentuk kalimat. Sebagai contoh, kita akan memberi keterangan penulisan nama diatas.

Contoh *Script*:

```
<?php
```

```
/* Untuk menuliskan komentar, biasanya komentar yang berbentuk kalimat */
```

```
Echo "Hello, Nama Saya Nayla";
```

```
?>
```

- g. Selain menggunakan tanda `/*`, penulisan *comment* juga dapat menggunakan tanda *slash* ganda (`//`). Tanda ini hanya digunakan untuk menuliskan pesan yang hanya berisi satu baris saja.

Contoh *Script*:

```
<?php
```

```
// Untuk menuliskan komentar satu baris
```

```
Echo "Hello, Nama Saya Nayla";
```

```
?>
```

- h. Semua *script HTML* yang akan digabungkan dalam *script PHP* harus dihilangkan tanda petiknya (`" "`). Sebagai contoh, jika kita ingin memberi warna *background* pada halaman kita, maka perintah yang kita gunakan dalam *HTML* adalah `<BODY bgcolor="#000000">`. Akan tetapi, jika kita letakkan dalam *PHP*, maka *script* tersebut harus diubah menjadi `<BODY bgcolor=#000000>`. Untuk lebih jelasnya, lihatlah contoh berikut.

Script pada HTML:

.....

```
<BODY bgcolor="#000000">
```

```
<?php
Print ("Nama Saya Nayla") ;
?>
.....
```

Script pada PHP:

```
<?
Echo (" <BODY bgcolor=#3366FF? ");
Print ("Halo Nayla, apa kabar");
?>
```

2.3.3. MySQL

Kadir (2008:2), "MySQL (baca: mai-se-kyu-el) merupakan *software* yang tergolong *DBMS (Database Management System)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara *men-download* (mengunduh) di internet secara gratis".

Kelebihan dari *MySQL* adalah dapat digunakan untuk aplikasi *multi-user* (banyak pengguna) dan menggunakan bahasa *query* (permintaan) standar *SQL (Structured Query Language)*.

Kekurangan dari *MySQL* adalah untuk koneksi ke bahasa pemrograman *visual* seperti *VB, Delphi, dan Foxpro, MySQL* kurang *support*, karena koneksi ini menyebabkan *field* yang dibaca harus sesuai dengan koneksi dari program visual tersebut, dan ini yang menyebabkan *MySQL* jarang dipakai dalam program visual serta data yang ditangani belum begitu besar.



2.3.4. Fungsi-fungsi MySQL

Untuk dapat menggunakan semua perintah *SQL* pada *PHP*, kita memerlukan fungsi bawaan *PHP* (*reserved word*) dengan beragam kemampuan, mulai dari menampilkan informasi, memanipulasi informasi, keperluan kondisional, keperluan perulangan, sampai pengaksesan berbagai *database*, termasuk *MySQL*.

Berikut adalah daftar beberapa fungsi *PHP* yang secara khusus digunakan untuk mengakses dan mengelola *database MySQL*.

Tabel 2.7. Fungsi-fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
1	<i>MySQL_connect()</i>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu <i>host</i> .	<i>mysql_connect</i> (<i>host</i> , <i>nama_pemakai</i> , <i>password</i>)
2	<i>MySQL_close()</i>	Menutup hubungan ke <i>database MySQL</i> .	<i>mysql_close</i> (<i>pengenal_hubungan</i>)
3	<i>MySQL_select_db()</i>	Memilih <i>database</i>	<i>mysql_select_db</i> (<i>database</i> , <i>pengenal_hubungan</i>)
4	<i>MySQL_query()</i>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel.	<i>mysql_query</i> (<i>permintaan</i> , <i>pengenal_hubungan</i>)
5	<i>MySQL_db_query()</i>	Menjalankan suatu permintaan terhadap suatu <i>database</i> .	<i>mysql_db_query</i> (<i>database</i> , <i>permintaan</i> , <i>pengenal_hubungan</i>)
6	<i>MySQL_num_rows()</i>	Memperoleh jumlah baris dari suatu hasil permintaan (<i>query</i>) yang menggunakan <i>SELECT</i> .	<i>mysql_num_rows</i> (<i>pengenal_hasil</i>)
7	<i>MySQL_affected_rows()</i>	Memperoleh jumlah baris yang dikenai operasi <i>INSERT</i> , <i>DELETE</i> , dan <i>UPDATE</i> .	<i>mysql_affected_rows</i> (<i>[pengenal_hubungan]</i>)
8	<i>MySQL_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan.	<i>mysql_num_fields</i> (<i>pengenal_hasil</i>)



Lanjutan Tabel 2.7. Fungsi-fungsi MySQL

9	<i>MySQL_fetch_row()</i>	Menghasilkan suatu array yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil.	<i>mysql_fetch_row</i> (<i>pengenal_hasil</i>)
10	<i>MySQL_fetch_array()</i>	Mempunyai kegunaan serupa dengan <i>mysql_fetch_row()</i> . Hanya saja, setiap kolom akan disimpan dua kali pada array hasil.	<i>mysql_fetch_array</i> (<i>pengenal_hasil</i>)
11	<i>MySQL_fetch_field()</i>	Memperoleh informasi suatu kolom.	<i>mysql_fetch_field</i> (<i>pengenal_hasil</i> [, <i>nomor_kolom</i>])
12	<i>MySQL_data_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke baris tertentu.	<i>mysql_data_seek</i> (<i>pengenal_hasil</i> , <i>nomor_baris</i>)
13	<i>MySQL_field_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke kolom tertentu.	<i>mysql_data_seek</i> (<i>pengenal_hasil</i> , <i>nomor_kolom</i>)
14	<i>MySQL_create_db()</i>	Menciptakan <i>database MySQL</i> .	<i>mysql_create_db</i> (<i>database</i> [, <i>pengenal_hubungan</i>])
15	<i>MySQL_drop_db()</i>	Menghapus <i>database MySQL</i> .	<i>mysql_drop_db</i> (<i>database</i> [, <i>pengenal_hubungan</i>])
16	<i>MySQL_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i> .	<i>mysql_list_dbs</i> ([<i>pengenal_hubungan</i>])
17	<i>MySQL_list_tables()</i>	Memperoleh daftar nama tabel dalam suatu <i>database</i> .	<i>mysql_list_tables</i> (<i>database</i> [, <i>pengenal_hubungan</i>])
18	<i>MySQL_list_fields()</i>	Memperoleh daftar nama kolom dalam suatu <i>database</i> .	<i>mysql_list_fields</i> (<i>database</i> [, <i>pengenal_hubungan</i>])

Sumber: Kadir (2008:360-379)



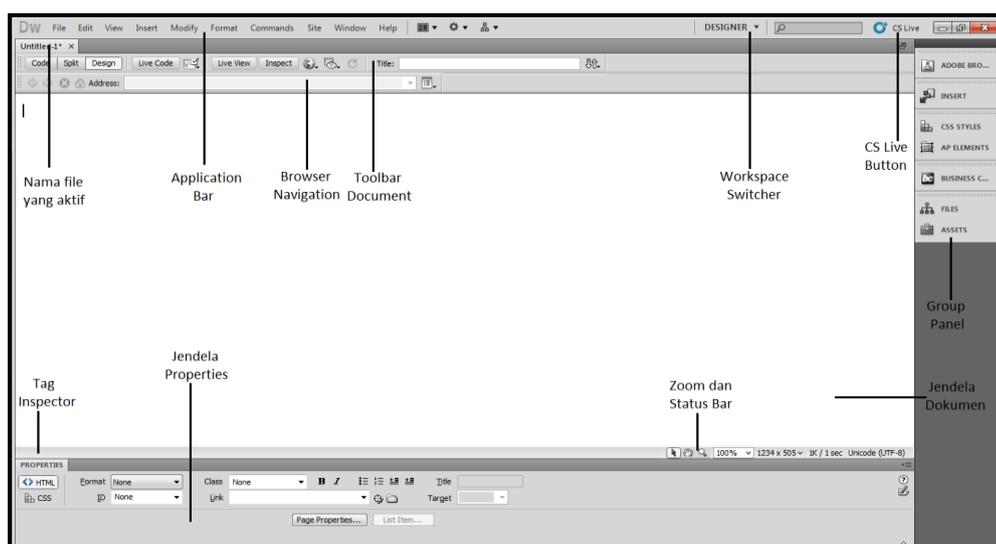
2.3.5. Adobe Dreamweaver CS5

Madcoms (2011:2), “*dreamweaver* adalah sebuah *HTML editor* profesional untuk mendesain *Web* secara *visual* dan mengelola situs atau halaman *Web*”. Pada *dreamweaver CS5*, terdapat beberapa kemampuan bukan hanya sebagai *software* untuk desain *Web* saja tetapi juga untuk menyunting kode serta pembuatan aplikasi *Web* dengan menggunakan berbagai bahasa pemrograman *Web*.



Gambar 2.4. Logo *Dreamweaver CS5*

Tampilan utama ruang kerja *Adobe Dreamweaver CS5* adalah sebagai berikut:



Gambar 2.5. Tampilan Ruang Kerja *Adobe Dreamweaver CS5*

1. *Application Bar*

Berada di bagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *workspace (workspace switcher)*, *CS Live*, menu dan aplikasi lainnya.



Gambar 2.6. Tampilan *Application Bar*

2. *Toolbar Document*

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *Design* atau tampilan *Code*. Juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.



Gambar 2.7. Tampilan *Toolbar Document*

3. Jendela Dokumen

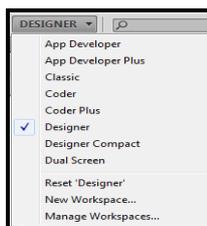
Adalah lembar kerja tempat membuat dan mengedit desain halaman *web*.



Gambar 2.8. Jendela Dokumen

4. *Workspace Switcher*

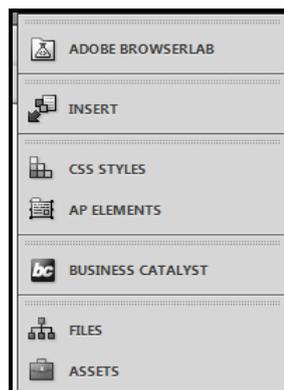
Digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Adobe Dreamweaver CS5*.



Gambar 2.9. Tampilan *Workspace Switcher*

5. *Panel Groups*

Merupakan kumpulan-kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan.



Gambar 2.10. Tampilan *Panel Groups*

6. *Tag Selector*

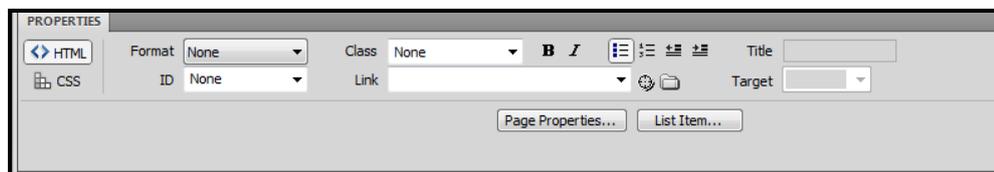
Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut.



Gambar 2.11. *Tag Selector*

7. *Panel Property*

Digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*.

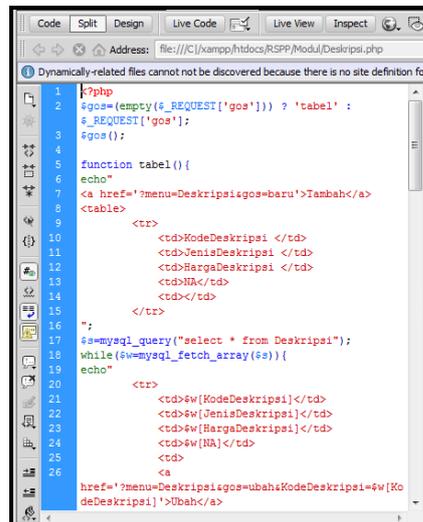


Gambar 2.12. Tampilan Jendela *Properties*



8. *Toolbar Coding*

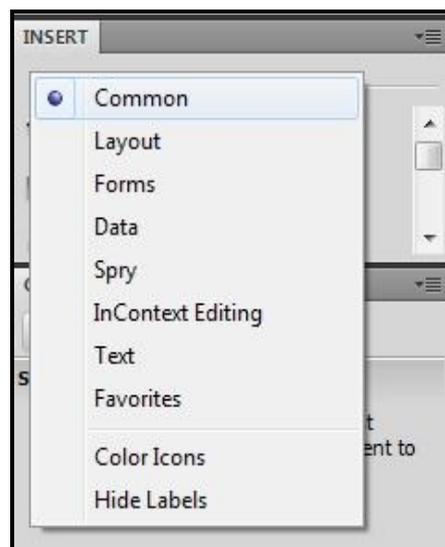
Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Jendela ini hanya akan tampil pada jendela *Code* dan *Split*.



Gambar 2.13. Tampilan *Toolbar Coding*

9. *Panel Insert*

Berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media ke dalam jendela dokumen.



Gambar 2.14. Tampilan *Panel Insert*



10. Panel Groups

Kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan dibawah satu judul, seperti: *Insert*, *Files*, *CSS Styles*, dan sebagainya.



Gambar 2.15. Tampilan *Panel Groups*

11. Panel Files

Digunakan untuk mengatur *file-file* dan *folder-folder* yang membentuk situs *Web* Anda.



Gambar 2.16. Tampilan *Panel Files*