



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Asropudin (2013:19), Komputer alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing computer.

Hartono (2013:27), Komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang untuk secara otomatis melakukan serangkaian urutan penghitungan (*arithmetic*) atau proses-proses yang diurutkan secara logis. Urutan-urutan tersebut dapat diubah seketika oleh computer, sehingga computer dapat menyelesaikan lebih dari satu tugas.

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.

Ladjamudin (2013 : 20), Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.

2.1.3 Pengertian Sistem

Hartono (2013:10), Sistem yakni suatu benda atau entitas (yaitu himpunan dari berbagai bagian atau komponen), dan sekaligus juga suatu proses atau metode cara untuk mencapai tujuan (yaitu saling berhubungan secara terorganisasi berdasar fungsi-fungsinya).

Rosa dan shalahudin (2013:291), Sistem Kumpulan komponen yang saling terakait dan mempunyai satu tujuan yang ingin dicapai.

2.1.4 Pengertian Informasi

Hartono (2013:15), Informasi pada dasarnya adalah sehimpunan data yang telah diolah menjadi Sesuatu yang memiliki arti dan kegunaan lebih luas.

Hartono (2013:15), Davis (1994) 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.

2.1.5 Pengertian Puskesmas

Berdasarkan peraturan Menteri Kesehatan Republik Indonesia Nomor 75 tahun 2014 Tentang Pusat Kesehatan Masyarakat dalam BAB 1 Pasal 1 Butir 2, Pusat Kesehatan Masyarakat yang selanjutnya disebut Puskesmas adalah fasilitas pelayanan kesehatan yang menyelenggarakan upaya kesehatan masyarakat dan upaya kesehatan perseorangan tingkat pertama, dengan lebih mengutamakan upaya promotif dan preventif, untuk mencapai derajat kesehatan masyarakat yang setinggi-tingginya di wilayah kerjanya.

2.1.6 Pengertian Sistem Informasi

Hartono (2013:20), Sistem Informasi adalah sebuah sistem yaitu rangkaian terorganisasi dari sejumlah bagian/komponen yang secara bersama-sama berfungsi atau bergerak menghasilkan informasi.

2.1.7 Sistem Informasi Integrasi Pelayanan Konsumen Pada Puskesmas Kertapati Kota Palembang

Merupakan sebuah Sistem Informasi Pelayanan Konsumen pada Puskesmas Secara Online dengan menggunakan Metode RUP Pada Puskesmas Kertapati Kota Palembang, sehingga dapat memberikan kemudahan dalam melakukan pelayanan konsumen di puskesmas kota Palembang menggunakan program *PHP* dan *MySQL*



2.2 Teori Khusus

2.2.1 Pemrograman Berorientasi Objek

2.2.1.1 Pengertian Pemrograman Berorientasi Objek

Rosa dan Shalauddin (2013:100), Meteologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang berlaku terhadapnya.

Keuntungan menggunakan meteologi berorientasi objek adalah sebagai berikut:

1. Meningkatkan produktivitas

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut(reusable).

2. Kecepatan pengembangan

Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurang kesalahan pada saat pengkodean.

3. Kemudahan pemeliharaan

Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

4. Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.



2.2.1.2 Meteologi *Rational Unified Process* (RUP)

Rosa dan Shalauddin (2013:125), *Rational Unified Process* (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*).

Rational Unified Process (RUP) menggunakan konsep *object oriented*, dengan aktivitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML).

2.2.1.3 Proses Pengulangan pada Meteologi *Rational Unified Process* (RUP)

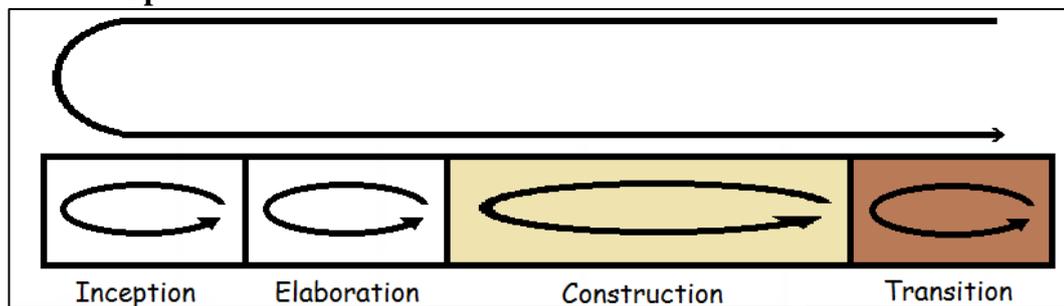
Proses pengulangan/interaktif pada RUP secara global dapat dilihat pada gambar berikut:



Gambar 2.1 Proses Interatif *Rational Unified Process* (RUP)



2.2.1.4 Penerapan Tahap Meteologi Pengembangan Perangkat Lunak dengan Rup



Gambar 2.2 Alur hidup RUP

(Sumber : Rosa dan Shalauddin, 2013:128)

Rosa dan Shalauddin (2013:128), Dalam Rational Unified Process terhadap empat tahap pengembangan perangkat lunak yaitu:

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

3. *Construction* (kontruksi)

Tahapan ini fokus pada pengembangan komponen dan fitur-fitur. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas kemampuan operasional awal.



4. *Transition* (transisi)

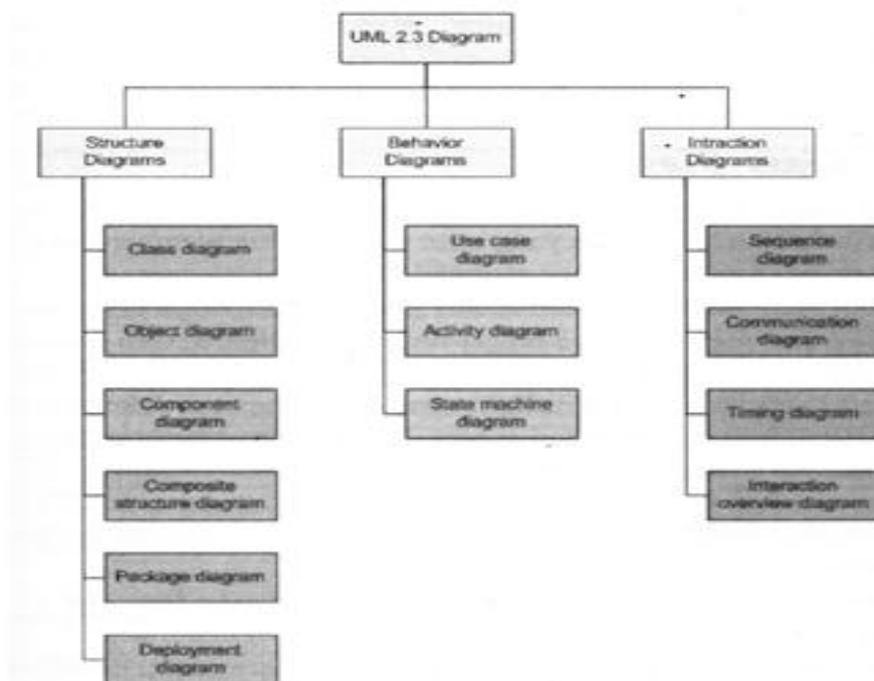
Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas / tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.2.2 *Unified Modeling Language* (UML)

Rosa dan Shalauddin (2013:137), UML, merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan sistem dengan menggunakan diagram dan teks-teks pendukung.

2.2.2.1 Macam-macam Diagram UML

Rosa dan Shalauddin (2013:140), Pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.3 Macam-macam Diagram UML

(Sumber : Rosa dan Shalauddin, 2013:140)



Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. Structure diagram

yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan .

b. Behaviour diagram

yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada suatu sistem.

c. Interaction diagram

yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun antar sistem pada suatu sistem.

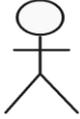
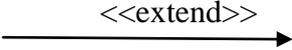
2.2.3 Diagram *Use Case* (*Use Case Diagram*)

Rosa dan Shalauddin (2013:155) *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

Tabel 2.1 Simbol-Simbol Diagram *Use Case* (*Use Case Diagram*)

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit–unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>

Lanjutan Tabel 2.1 Simbol-Simbol Diagram *Use Case* (*Use Case Diagram*)

Simbol	Deskripsi
Aktor/ <i>actor</i>  Nama Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/ <i>association</i> 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
Ekstensi/ <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.

Lanjutan Tabel 2.1 Simbol-Simbol Diagram *Use Case* (*Use Case Diagram*)

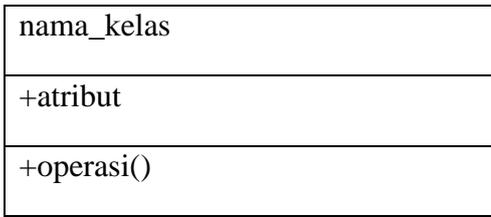
Simbol	Deskripsi
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum–khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>include/uses</i> <<include>>  <<uses>> 	Relasi <i>use case</i> tambahan ke sebuah <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 :Rosa Dan Shalauddin, 2013:155)

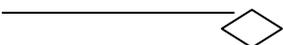
2.2.4 Diagram Kelas (*Class Diagram*)

Rosa Dan Shalauddin (2013:141), Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.2 Simbol-Simbol Diagram Kelas (*Class Diagram*)

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem

Lanjutan Tabel 2.2 Simbol-Simbol Diagram Kelas (*Class Diagram*)

Simbol	Deskripsi
Antarmuka/interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi/association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan/dependency 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi/aggregation 	Relasi antarkelas dengan makna semuabagian (whole-part).

(Sumber :Rosa Dan Shalauddin, 2013:147)

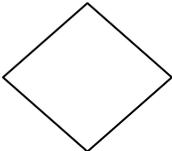
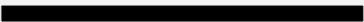
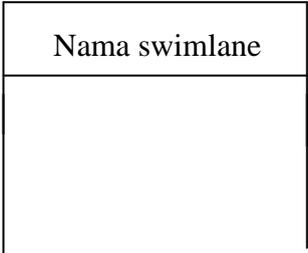
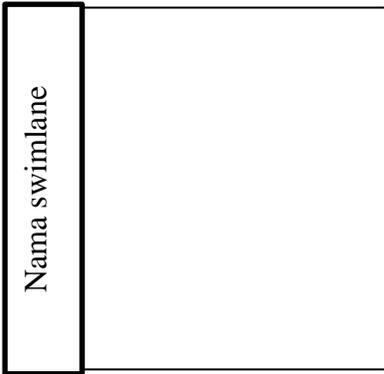
2.2.5 Diagram Aktivitas (*Activity Diagram*)

Rosa Dan Shalauddin, 2013:161 Diagram aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Tabel 2.3 Simbol-Simbol Diagram Aktivitas (*Activity Diagram*)

Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem, sebuah sistem diagram aktivitas memiliki sebuah status awal.

Lanjutan Tabel 2.3 Simbol-Simbol Diagram Aktivitas (*Activity Diagram*)

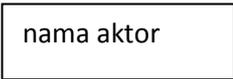
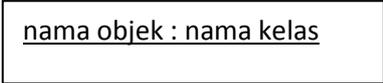
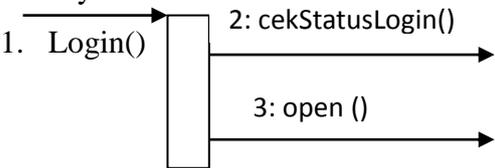
Simbol	Deskripsi
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber :Rosa Dan Shalauddin, 2013:162)

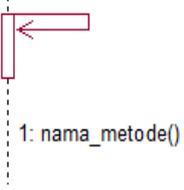
2.2.6 Diagram sekuen (*Sequence Diagram*)

Rosa Dan Shalauddin, (2013:165) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel 2.4 Simbol-Simbol Diagram sekuen (*Sequence Diagram*)

Simbol	Deskripsi
Aktor  nama aktor Atau  nama aktor Tanpa waktu aktif	orang , proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasa nya dinyatakan menggunakan kata benda di awal frase nama aktor.
Garis hidup / lifeline	menyatakan kehidupan suatu objek
Objek  nama objek : nama kelas	menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya 

Lanjutan Tabel 2.4 Simbol-Symbol Diagram sekuen (*Sequence Diagram*)

Simbol	Deskripsi
	<p>maka cekStatusLogin () dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif</p>
Pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe call 1: nama_metode 	<p>menyatakan suatu objek memanggil operasi/ metode yan ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah megarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
Pesan tipe send 1 : masukkan 	menyatakan bahwa suatu objek megirimkan data/ masukkan/ informasi ke objek lainnya, arah panah megarah pada objek yang dikirim
pesan tipe return 1: keluaran 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

Lanjutan Tabel 2.4 Simbol-Simbol Diagram sekuen (*Sequence Diagram*)

Simbol	Deskripsi
<p>pesan tipe return</p> <p>1: keluaran</p> 	<p>menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> 	<p>menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>

(Sumber :Rosa Dan Shalauddin, 2013:167)

2.3 Teori Program

2.3.1 Pengenalan PHP

Mulhim (2013:2), PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang berjalan disisi server, untuk itulah dibutuhkan sebuah server yang dapat menterjemahkan atau menjalankan script PHP kedalam Web Browser.

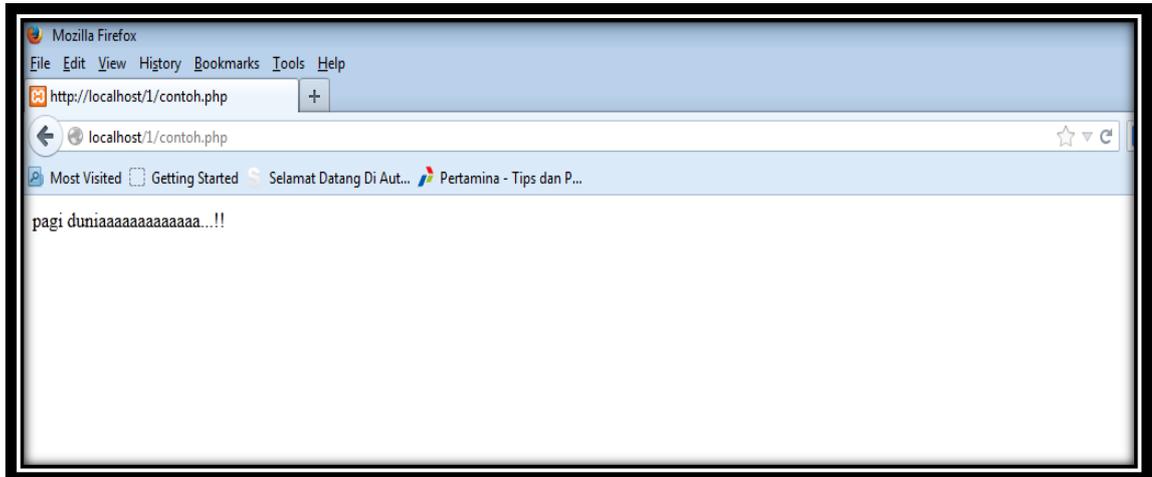
2.3.1.1 Dasar Perintah PHP

Nugroho (2013:155), Untuk menuliskan kode program PHP, anda harus mengetikkan kode pengenal kode PHP, yaitu dengan cara memulai menggunakan perintah `<?php` dan diakhiri dengan perintah `?>`.

Contoh :

```
<?php
echo "pagi duniaaaaaaaaaaaaa...!!";
?>
```

Bila dijalankan browser, kode tersebut membentuk tampilan sebagai berikut :



Gambar 2.4 Tampilan PHP Sederhana

2.3.2 Pengertian *MySql*

Nugroho (2013:26), *MySQL* itu adalah software atau program **Database Server**, dia itu software atau programnya. Sedangkan *SQL* adalah bahasa pemrogramannya, dia itu bahasa permintaan(Query) dalam database server, termasuk dalam *MySQL* itu sendiri. *SQL* juga dipakai dalam software database server lain, seperti *SQL Server, Oracle, PostgreSQL* dan lainnya.

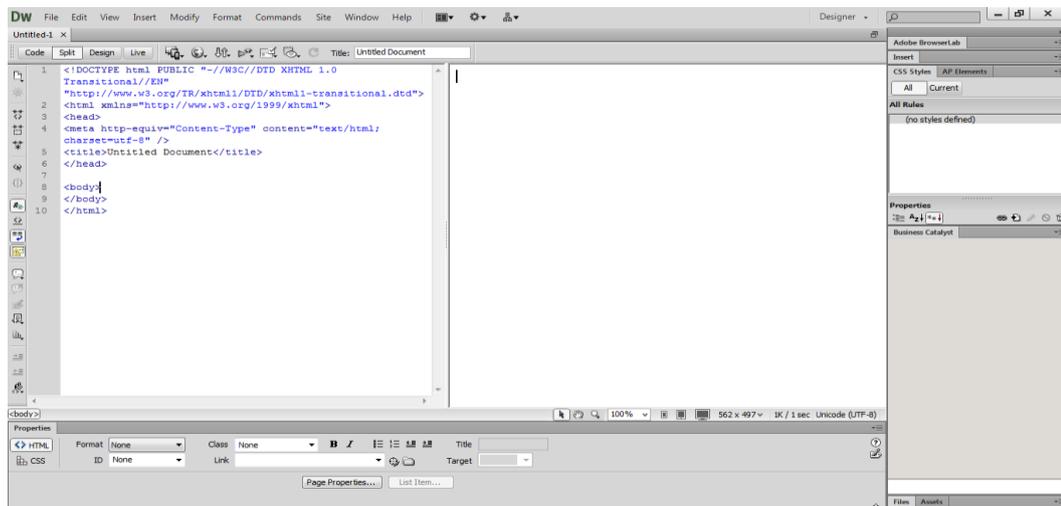
2.3.3 Pengenalan *Dreamweaver CS6*

Elcom (2013:2), *Dreamweaver CS6* adalah sebuah editor profesional untuk mendesain *web* secara visual dan mengelola situs atau halaman *web*. Saat ini terdapat banyak *software adobe* yang digunakan untuk mendesain suatu *web*. Versi terbaru dari adobe dreamweaver CS6 memiliki beberapa kemampuan. Versi ini bukan hanya software untuk desain *web*, tetapi juga untuk menyunting kode serta pembuatan aplikasi web antara lain *JSP, PHP, ASP, XML*, dan *ColdFusion*.

Dreamweaver merupakan software utama yang digunakan oleh desainer *web* dan programmer *web* untuk mengembangkan suatu situs *web*. Dreamweaver memiliki ruang kerja, fasilitas, dan kemampuan yang mampu meningkatkan produktivitas dan efektivitas, baik dalam desain maupun pembuatan situs *web*.



2.3.4 Tampilan Ruang Kerja *Adobe Dreamweaver CS6*



Gambar 2.5 Tampilan Ruang Kerja *Dreamweaver CS6*

Adapun keterangan komponen-komponen yang terdapat di dalam ruang kerja *dreamweaver CS6* :

1. **Insert Bar**, berisi tombol-tombol untuk menyisipkan berbagai macam objek seperti image, tabel, dan layer ke dalam dokumen.
2. **Document Toolbar**, berisi tombol-tombol dan menu pop-up yang menyediakan tampilan berbeda dari jendela dokumen.
3. **Coding Window**, berisi kode-kode HTML dan tempat untuk menuliskan kode-kode pemrograman, misal PHP atau ASP.
4. **Panel Group**, berisi kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan di bawah satu judul.
5. **Property Inspector**, digunakan untuk melihat dan mengubah berbagai property objek atau teks.
6. **Jendela Dokumen**, digunakan untuk menampilkan dokumen saat di mana anda sekarang bekerja.
7. **Ruler**, mempermudah ukuran dalam mendesain halaman *web*.
8. **Site Panel**, digunakan untuk mengatur file-file dan folder-folder yang membentuk situs *web* anda.