



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Hartono (2013:27), “Komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang secara otomatis melakukan serangkaian urutan perhitungan (*arithmetic*) atau proses-proses yang diurutkan secara logis”.

Sutarman (2012:2), “Komputer adalah sistem untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan agar secara otomatis menerima dan menyimpan data *input*, memprosesnya, dan menghasilkan *output* dibawah pengawasan suatu langkah-langkah instruksi program yang tersimpan pada memori (*stored program*)”.

2.1.2. Pengertian Teknologi

Phoenix (2009:1422), “Teknologi adalah metode ilmiah untuk mencapai tujuan praktis; ilmu pengetahuan terapan; keseluruhan sarana untuk menyediakan barang-barang yang diperlukan bagi kelangsungan dan kenyamanan hidup manusia”.

Nurianti (2013), “Teknologi adalah keseluruhan sarana untuk menyediakan barang-barang yang diperlukan bagi kelangsungan dan kenyamanan hidup manusia”.

2.1.3. Pengertian Internet

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

Budinugroho (2013), “*Internet (Interconnection-Networking)* adalah jaringan besar yang saling berhubungan dari jaringan-jaringan komputer yang



menghubungkan orang-orang dan komputer-komputer diseluruh dunia, melalui telepon, satelit dan sistem-sistem komunikasi yang lain”.

2.1.4. Pengertian Web

Raharjo (2011:2), “*Web (world wide web/www)* adalah suatu layanan di dalam jaringan internet yang berupa informasi”.

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

2.1.5. Pengertian Pengolahan data

Kristanto (2008:8), “Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan”. Operasi yang dilakukan dalam pengolahan data antara lain: *input* data, *transformasi* data dan *output* data.

2.1.6. Pengertian Basis Data (*Database*)

Kristanto (2008:73), “Basis data (*database*) adalah kumpulan data yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi”.

Jogiyanto (2009:46), “Basis data (*database*) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya”.

2.1.7. Pengertian Sekolah

Phoenix (2009:1244), “Sekolah adalah bangunan atau lembaga untuk belajar dan mengajar serta tempat menerima dan memberi pelajaran”.

Lentera (2011), “Sekolah adalah sebuah lembaga yang dirancang untuk pengajaran siswa/murid di bawah pengawasan guru”.



2.1.8. Metode Pengembangan Sistem

Sukamto dan Shalahudin (2013:28), menjelaskan tentang metode pengembangan sistem model waterfall, sebagai berikut:

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 dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasi kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. 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. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

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

4. Pengujian

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

5. Pendukung (*support*)

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.



2.1.9. Metode Pengujian Perangkat Lunak

2.1.9.1. Pengertian Metode Pengujian

Shihab (2011), “Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan”.

Pengujian perangkat lunak perlu dilakukan untuk mengevaluasi baik secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum, dan untuk menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya.

2.1.9.2. Metode Pengujian

Perangkat lunak dapat diuji dengan dua cara, yaitu:

1. Pengujian dengan menggunakan data uji untuk menguji semua elemen program (data internal, loop, keputusan dan jalur). Data uji dibangkitkan dengan mengetahui struktur internal (kode sumber) dari perangkat lunak.
2. Pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional perangkat lunak bekerja dengan baik. Data uji dibangkitkan dari spesifikasi perangkat lunak.

2.1.9.3. Metode *Black Box Testing*

Iskandaria (2012), “Pengujian *blackbox (blackbox testing)* adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada *input* dan *output* aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Tahap pengujian merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak.

Shihab (2011), “*Black Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program”.



Shihab (2011), mengemukakan ciri-ciri *black box testing*, yaitu:

1. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih dari pada itu, ia merupakan pendekatan pelengkap dalam mencakup error dengan kelas yang berbeda dari metode *white box testing*.
3. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

2.2. Teori Judul

2.2.1. Pengertian Sistem

Al Fatta (2007:3), “Sistem adalah suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain”.

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”.

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

Dapat disimpulkan bahwa definisi dari sistem adalah kumpulan dari elemen-elemen yang terorganisasi, terelasi dan bekerja sama untuk menyelesaikan tujuan tertentu.

2.2.2. Pengertian Informasi

Al Fatta (2007:9), “Informasi adalah suatu data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang”.

Kristanto (2008:6), “Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima”.



Hartono (2013:9), “Informasi adalah sehimpunan data yang telah diolah menjadi sesuatu yang memiliki arti dan kegunaan lebih luas”.

Dapat disimpulkan bahwa definisi dari informasi adalah hasil dari pengolahan data yang bermanfaat bagi penerimanya.

2.2.3. Pengertian Sistem Informasi

Al Fatta (2007:14), “Sistem informasi merupakan sistem dengan komponen-komponen yang bekerja untuk mengolah data menjadi informasi”.

Kristanto (2008:11), “Sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut”.

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

Dapat disimpulkan bahwa definisi dari sistem informasi adalah sistem dengan elemen-elemen yang saling bekerja sama untuk menghasilkan informasi yang bermanfaat bagi penerimanya.

2.2.4. E-Learning

2.2.4.1. Pengertian E-Learning

Wena (2013:202), “*E-education* atau *e-learning* adalah kegiatan pendidikan atau pembelajaran melalui media elektronik, khususnya melalui jaringan internet”.

2.2.4.2. Fungsi E-Learning

Wena (2013:212), ada tiga fungsi pembelajaran elektronik atau *e-learning* terhadap kegiatan pembelajaran di dalam kelas, yaitu sebagai berikut:

1. Sebagai suplemen/tambahan pembelajaran yang sifatnya pilihan atau optional.

Apabila peserta didik mempunyai kebebasan memilih, apakah siswa akan memanfaatkan materi pembelajaran elektronik atau menggunakan pembelajaran model konvensional. Jadi, dalam hal ini tidak ada kewajiban/keharusan bagi siswa untuk mengakses materi pembelajaran elektronik. Sekalipun sifatnya optional, peserta didik yang



memanfaatkannya tentu akan memiliki tambahan pengetahuan atau wawasan.

2. Sebagai pelengkap/komplemen pembelajaran.
Apabila materi pembelajaran elektronik diprogramkan untuk melengkapi materi pembelajaran yang diterima siswa di dalam kelas konvensional.
3. Sebagai pengganti/substitusi pembelajaran.
Jika pembelajaran elektronik sepenuhnya digunakan dalam proses pembelajaran. Dalam kondisi ini, siswa hanya belajar lewat pembelajaran elektronik saja, tanpa menggunakan model pembelajaran lainnya.

2.2.4.3. Manfaat *E-Learning*

Wena (2013:213), pembelajaran elektronik (*e-learning*) bermanfaat bagi berbagai pihak terkait, diantaranya:

1. Bagi siswa
Dengan kegiatan pembelajaran melalui *e-learning* dimungkinkan berkembangnya fleksibilitas belajar siswa yang optimal, di mana siswa dapat mengakses bahan-bahan belajar setiap saat dan berulang-ulang.
2. Bagi guru
Dengan adanya kegiatan pembelajaran *e-learning* ada beberapa manfaat yang diperoleh guru, yaitu:
 - a. Lebih mudah melakukan pemutakhiran bahan-bahan belajar yang menjadi tanggung jawabnya sesuai dengan tuntutan perkembangan keilmuan yang terjadi.
 - b. Mengembangkan diri atau melakukan penelitian guna meningkatkan wawasannya karena waktu luang yang dimiliki relatif lebih banyak.
 - c. Mengontrol kebiasaan belajar siswa.
 - d. Mengecek apakah siswa mengerjakan soal-soal latihan setelah memberikan topik tertentu.

Manfaat pembelajaran elektronik secara umum terdiri atas 4 hal, yaitu:

- a. Meningkatkan kadar interaksi pembelajaran antara siswa dengan guru (*enchance interactivity*).
- b. Memungkinkan terjadinya interaksi pembelajaran dari mana dan kapan saja (*time and place flexibility*).
- c. Menjangkau peserta didik dalam jangkauan yang luas (*potential to reach a global audience*).
- d. Mempermudah penyempurnaan dan penyampaian materi pembelajaran (*easy updating of contents as well as aechivable capabilities*).



3. Bagi sekolah

Dengan adanya model pembelajaran *e-learning*, maka sekolah:

- a. Akan tersedia bahan ajar yang telah divalidasi sesuai dengan bidangnya, sehingga guru dapat menggunakan dengan mudah serta efektivitas dan efisiensi pembelajaran secara keseluruhan akan meningkat.
- b. Pengembangan isi pembelajaran akan sesuai dengan pokok-pokok bahasan.
- c. Sebagai pedoman praktis implementasi pembelajaran sesuai dengan kondisi dan karakteristik pembelajaran.
- d. Mendorong menumbuhkan sikap kerja sama antara guru dengan guru, guru dengan siswa, siswa dengan sesama siswa dalam memecahkan masalah pembelajaran.

2.2.4.4. Kelemahan *E-Learning*

Kelemahan utama pembelajaran *e-learning* adalah sebagai berikut:

1. Frekuensi kontak secara langsung antar sesama siswa maupun antar siswa dengan narasumber sangat minim.
2. Peluang siswa untuk bersosialisasi dengan siswa lain sangat terbatas.

Salah satu yang harus ditekankan dan dipahami adalah bahwa *e-learning* tidak dapat sepenuhnya menggantikan kegiatan pembelajaran di kelas.

2.2.5. Pengertian SMA

Lentera (2011), “SMA (Sekolah Menengah Atas) merupakan jenjang pendidikan menengah setelah menamatkan Sekolah Menengah Pertama (SMP) atau yang sederajat”.

2.2.6. Pengertian Sistem Informasi *E-Learning* pada SMA Negeri 4 Palembang

Pengertian Sistem Informasi *E-Learning* pada SMA Negeri 4 Palembang adalah suatu sistem pembelajaran melalui media elektronik yang terhubung dengan jaringan internet yang dibuat untuk mempermudah guru dan siswa dalam memberikan, mencari serta menerima informasi yang lebih berguna dan lebih berarti.



2.3. Teori Khusus

2.3.1. Data Flow Diagram (DFD)

Kristanto (2008:55), “*Data flow diagram/DFD* adalah suatu model logika atau proses yang dibuat untuk menggambarkan dari mana asal data dan ke mana tujuan data yang keluar dari sistem, di mana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

Al-Fatta (2007:105), ”*Data flow diagram/DFD* merupakan diagram yang digunakan untuk menggambarkan proses-proses yang terjadi pada sistem yang akan dikembangkan”. Dengan model ini, data-data yang terlibat pada masing-masing proses dapat diidentifikasi.

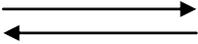
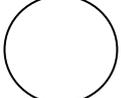
Sukanto dan Shalahudin (2013:70), “*Data flow diagram (DFD)* adalah representasi grafik yang menggambarkan aliran informasi dan transformasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

Dapat disimpulkan bahwa *data flow diagram (DFD)* adalah diagram yang menjelaskan diagram konteks secara lebih terperinci.

Ada beberapa simbol DFD yang dipakai untuk menggambarkan data beserta proses transformasi data, antara lain:

1. Teknik Yourdan Dan De Marco

Tabel 2.1. Simbol- Simbol *Data Flow Diagram* Yourdan & De Marco

Simbol	Keterangan
	<i>External Entity</i> , yaitu merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem.
	<i>Data Flow</i> , digunakan untuk menggambarkan aliran data dari satu proses ke proses lainnya.
	<i>Process</i> , digunakan untuk mentransformasikan data secara umum



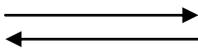
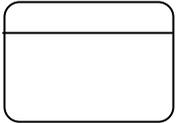
Lanjutan **Tabel 2.1.** Simbol- Simbol *Data Flow Diagram* Yourdan & De Marco

Simbol	Keterangan
	<i>Data Store</i> , digunakan untuk menyimpan data seperti: suatu <i>file</i> suatu arsip, suatu kotak, suatu tabel dan suatu agenda.

(Sumber: Al Fatta , 2007:107)

2. Teknik Gane dan Sarson

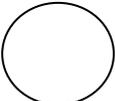
Tabel 2.2. Simbol- Simbol *Data Flow Diagram* Gane & Sarson

Simbol	Keterangan
	<i>External Entity</i> , yaitu merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem.
	<i>Data Flow</i> , digunakan untuk menggambarkan aliran data dari satu proses ke proses lainnya.
	<i>Process</i> , digunakan untuk mentransformasikan data secara umum.
	<i>Data Store</i> , digunakan untuk menyimpan data seperti: suatu <i>file</i> , suatu arsip, suatu kotak, suatu tabel dan suatu agenda.

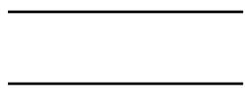
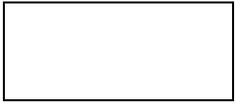
(Sumber: Al Fatta , 2007:107)

Ada beberapa simbol DFD yang dipakai untuk menggambarkan data beserta proses transformasi data, antara lain:

Tabel 2.3. Simbol- Simbol *Data Flow Diagram*

Notasi	Keterangan
	Proses atau fungsi atau prosedur. Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.

Lanjutan Tabel 2.3. Simbol- Simbol *Data Flow Diagram*

Notasi	Keterangan
	<i>Field</i> atau basis data atau penyimpanan (<i>storage</i>). Nama yang diberikan pada sebuah penyimpanan biasanya berupa kata benda.
	<i>Entitas</i> atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang. Nama yang diberikan biasanya berupa kata benda.
	Aliran data; merupakan data yang dikirim antarproses, dari penyimpanan ke proses atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>). Nama yang diberikan biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data_siswa” atau tanpa kata data misalnya “siswa”.

(Sumber: Sukamto dan Shalahudin , 2013:71)

2.3.2. Kamus Data (*Data Dictionary*)

Jogiyanto (2005:725), “Kamus data atau *data dictionary* atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”.

Kristanto (2008:66), “Kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau file di dalam sistem”.

Sukamto dan Shalahudin (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)”.

Adapun simbol-simbol dalam kamus data adalah sebagai berikut:

Tabel 2.4. Simbol-simbol Kamus Data (*Data Dictionary*)

Simbol	Keterangan
=	Disusun, terdiri atas, mendefinisikan, diuraikan menjadi, artinya
+	Dan
()	Opsional/Pilihan (boleh Ya atau Tidak)

Lanjutan **Tabel 2.4.** Simbol-simbol Kamus Data (*Data Dictionary*)

Simbol	Keterangan
[]	Salah satu alternative
*	Komentar, catatan, keterangan
@	Identifikasi atribut kunci, petunjuk (<i>key field</i>)
	Pemisah alternatif simbol []

(Sumber: Kristanto, 2008:66)

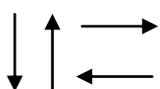
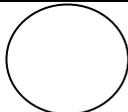
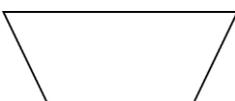
2.3.3. Pengertian *Flowchart*

Community eWolf (2012:16), “*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung”. Jadi setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya.

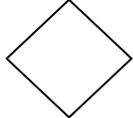
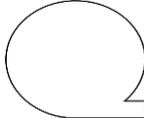
Flowchart digunakan untuk mempermudah penyusunan program. Dengan menggunakan *flowchart*, logika pemrograman lebih dapat dipahami dan dianalisis, sehingga dapat menentukan kode-kode pemrograman yang sesuai dengan pekerjaannya.

Berikut adalah beberapa simbol standar yang sering digunakan dalam pemrograman komputer, yaitu:

Tabel 2.5. Simbol-simbol *Flowchart*

Simbol	Nama Simbol	Keterangan
	Garis Alir (<i>Flow Line</i>)	Simbol untuk menghubungkan antar proses atau antar simbol
	<i>Connector</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang sama
	<i>Offline Connector</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
	<i>Process</i>	Simbol pemrosesan yang terjadi pada sebuah alur kerja
	<i>Manual</i>	Simbol yang mendefinisikan proses yang dilakukan secara manual

Lanjutan Tabel 2.5. Simbol-simbol *Flowchart*

Simbol	Nama Simbol	Keterangan
	<i>Decision</i>	Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
	<i>Terminal</i>	Simbol <i>start</i> atau <i>end</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i>
	<i>Input/Output</i>	Simbol <i>input/output</i> yang mendefinisikan masukan dan keluaran proses
	<i>Magnetic Tape</i>	Simbol masukan atau keluaran dari atau ke sebuah pita magnetik
	<i>Document</i>	Simbol masukan atau keluaran dari atau ke sebuah dokumen
	<i>Display</i>	Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll
	<i>Data Storage</i>	Simbol database atau basis data
	<i>Sub Program</i>	Simbol yang menyatakan bagian dari program (sub program)

(Sumber: Community eWolf, 2012:17)

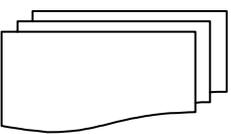
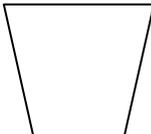
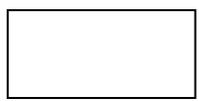
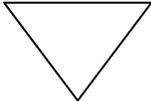
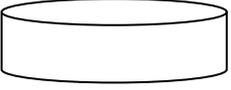
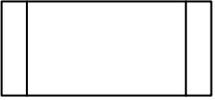
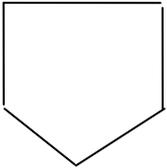
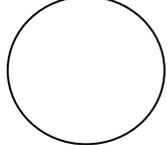
2.3.4. Pengertian *Blockchart*

Kristanto (2008:68), “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

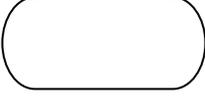
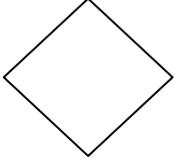


Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

Tabel 2.6. Simbol-simbol dalam *Blockchart*

Simbol	Keterangan
	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
	Multi dokumen
	Proses manual.
	Proses yang dilakukan oleh komputer.
	Menandakan dokumen yang diarsifkan(arsif manual)
	Data penyimpanan(<i>data storage</i>)
	Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama

Lanjutan **Tabel 2.6.** Simbol-simbol dalam *Blockchart*

Simbol	Keterangan
	Terminasi yang menandakan awal dan akhir dari suatu aliran
	Pengambilan keputusan (<i>decision</i>)
	Layar peraga (<i>monitor</i>)
	Pemasukan data secara manual

(Sumber: Kristanto, 2008:68)

2.3.5. Entity Relational Diagram (ERD)

Al Fatta (2007:121), *entity relationship diagram/ERD* adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan antar data.

1. Entitas

Entitas bisa berupa orang, kejadian, benda di mana data akan dikumpulkan.

2. Relationship

- a. Hubungan antar 2 entitas.
- b. Entitas pertama dalam *relationship* disebut entitas induk, entitas kedua disebut sebagai entitas anak.
- c. *Relationship* harus memiliki nama yang berupa kata kerja.
- d. *Relationship* berjalan 2 arah.

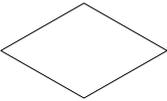
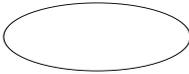
3. Atribut

- a. Informasi yang diambil tentang sebuah entitas.
- b. Hanya yang digunakan oleh organisasi yang dimasukkan dalam model.
- c. Nama atribut harus merupakan kata benda.



Simbol-simbol yang digunakan dalam ERD, yaitu:

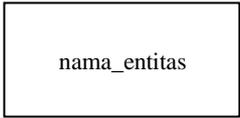
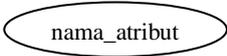
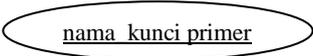
Tabel 2.7. Simbol-simbol *Entity Relationship Diagram (ERD)*

Nama Simbol	Simbol	Keterangan
<i>Entitas</i>		Orang,tempat, benda memiliki nama tunggal.
<i>Relationship</i>		Menunjukkan hubungan antar 2 entitas. Dideskripsikan dengan kata kerja.
<i>Atribut</i>		Properti dari entitas harus digunakan oleh minimal 1 proses bisnis dipecah dalam detail.
<i>Link</i>		Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

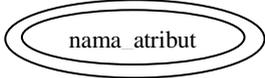
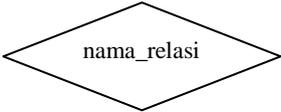
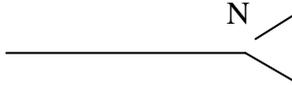
(Sumber: Al Fatta , 2007:124)

Simbol-simbol yang digunakan dalam ERD, yaitu:

Tabel 2.8. Simbol- Simbol *Entity Relationship Diagram (ERD)*

Simbol	Deskripsi
Entitas / <i>Entity</i> 	Entitas merupakan data inti yang akan tersimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut Kunci Primer 	<i>Field/</i> kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

Lanjutan Tabel 2.8. Simbol- Simbol *Entity Relationship Diagram* (ERD)

Simbol	Deskripsi
Atribut Multinilai / <i>Multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antarentitas; biasanya diawali dengan kata kerja.
Asosiasi / <i>Association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya punya <i>multiplicity</i> kemungkinan jumlah pemakaian.

(Sumber: Sukamto dan Shalahudin , 2013:50)

Ada beberapa pedoman yang perlu diperhatikan untuk membuat ERD, diantaranya:

- Entitas harus memiliki banyak kejadian atau realitas.
- Hindari penggunaan atribut yang tidak perlu.
- Berilah label yang jelas untuk semua komponen.
- Pasangkan kardinalitas dan modalitas yang jelas dan benar.
- Pecah atribut menjadi level serendah mungkin yang diperlukan.
- Level harus menrefleksi istilah-istilah bisnis yang umum.
- Asumsi harus disebutkan dengan jelas.

2.4. Teori Program

2.4.1. Sekilas Tentang PHP

2.4.1.1. Pengertian PHP

Kadir (2008:2), “*PHP* merupakan singkatan dari *PHP Hypertext Preprocessor*) yang merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*”. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. *PHP* dirancang untuk membentuk aplikasi *web* dinamis.



Madcoms (2010:341), “Bahasa pemrograman *PHP* adalah bahasa pemrograman yang bekerja dalam sebuah *web server*”. *Script-script PHP* dibuat harus tersimpan dalam sebuah *server* dan dieksekusi atau diproses dalam *server* tersebut.

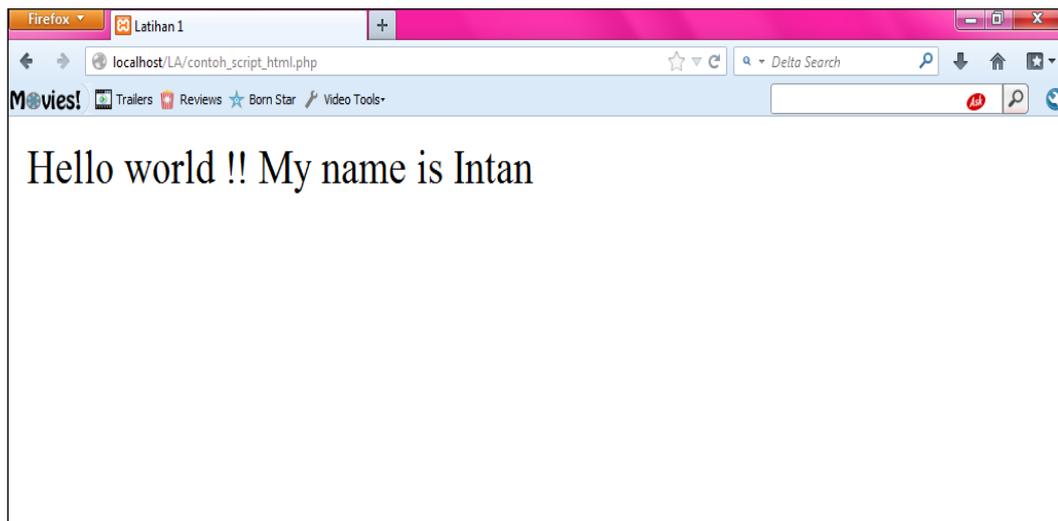
Berdasarkan definisi mengenai pengertian *PHP* di atas, maka dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pengolah data pada sebuah *server*.

2.4.1.2. *Script PHP*

Script PHP berkedudukan sebagai tag dalam bahasa *HTML*. Sebagaimana diketahui, *HTML* (*Hypertext Markup Language*) adalah bahasa standar untuk membuat halaman-halaman *web*. Sebagai contoh, berikut adalah kode *HTML* (disimpan dengan ekstensi *.htm* atau *.html*):

```
<html>
  <head>
    <title>Latihan 1</title>
  </head>
  <body>
    Hello world, my name is Intan
  </body>
</html>
```

Bila dijalankan melalui *browser*, kode diatas akan menampilkan hasil seperti berikut:



Gambar 2.1. Tampilan program HTML pada *browser*



2.4.1.3. Teknik Penulisan Script PHP

Madcoms (2010:350) menyatakan, “*Script PHP* harus selalu diawali dengan `<? Atau <?php` dan diakhiri dengan `?>`. sedangkan untuk menampilkan ke dalam *browser*, digunakan fungsi **echo”data”**”; Semua teks yang diketik setelah tanda buka *script* (`<?`) dan tanda tutup *script* (`?>`) akan dieksekusi sebagai suatu *script PHP*. Contoh penulisan *script PHP* adalah sebagai berikut:

Tabel 2.9. *Script Dasar PHP*

<pre><? Tempat penulisan script PHP ?></pre>
atau
<pre><?php Tempat penulisan script PHP ?></pre>
Dan untuk menampilkan ke <i>browser</i>
<pre><?php Echo “Welcome to PHP”; ?></pre>

Di dalam *script PHP* juga dapat membuat komentar dan komentar tersebut tidak akan dieksekusi sebagai sebuah *script*. Beberapa cara untuk memberi keterangan di dalam *script PHP* adalah sebagai berikut:

- Gunakan *tag /** dan akhiri *tag */* apabila jumlah keterangan lebih dari 1 baris.
- Gunakan *tag //*, *tag* ini digunakan untuk keterangan yang hanya terdiri dari 1 baris saja.
- Gunakan *tag #*, *tag* ini juga digunakan untuk 1 baris komentar saja.

Dalam penulisannya, baris perintah *script PHP* selalu diakhiri dengan menuliskan tanda titik koma (;). Satu baris *script PHP* tidak harus berada dalam satu baris, melainkan dapat menuliskan perintah *script PHP* lebih dari satu baris.



2.4.2. Sekilas Tentang MySQL

2.4.2.1. Pengertian MySQL

Kadir (2008:2), “MySQL (baca mai-se-kyu-el) merupakan *software* yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*”. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dibuat untuk membuat MySQL), selain bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi.

2.4.2.2. Fungsi-Fungsi MySQL

Untuk dapat menggunakan semua perintah *SQL* pada *PHP*, kita memerlukan fungsi *MySQL*. Fungsi ini berguna untuk mengantarkan perintah *SQL* pada *PHP* menuju ke *server* sehingga perintah tersebut dapat dieksekusi oleh semua *server MySQL*.

Fungsi-fungsi *MySQL* yang berawalan *mysql_* yang digunakan untuk mengakses *database server MySQL* adalah sebagai berikut:

Tabel 2.10. Fungsi-fungsi *MySQL* dalam *PHP*

No.	Fungsi MySQL	Penggunaan
1	<i>Mysql_connect()</i>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu host
2	<i>Mysql_close()</i>	Menutup hubungan ke <i>database MySQL</i>
3	<i>Mysql_select_db()</i>	Memilih <i>database</i>
4	<i>Mysql_query()</i>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel
5	<i>Mysql_db_query()</i>	Menjalankan suatu permintaan terhadap suatu <i>database</i>
6	<i>Mysql_num_rows()</i>	Memperoleh jumlah baris dari suatu hasil permintaan yang menggunakan <i>SELECT</i>
7	<i>Mysql_affected_rows()</i>	Memperoleh jumlah baris yang dikenai operasi <i>INSERT, DELETE, UPDATE</i>
8	<i>Mysql_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan

Lanjutan Tabel 2.10. Fungsi-fungsi *MySQL* dalam *PHP*

No.	Fungsi <i>MySQL</i>	Penggunaan
9	<i>Mysql_fetch_row()</i>	Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil
10	<i>Mysql_fetch_array()</i>	Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil yang akan disimpan dua kali pada <i>array</i> hasil
11	<i>Mysql_fetch_field()</i>	Menghasilkan informasi suatu kolom
12	<i>Mysql_data_seek()</i>	Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu
13	<i>Mysql_field_seek()</i>	Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke kolom tertentu
14	<i>Mysql_create_db()</i>	Membuat <i>database MySQL</i>
15	<i>Mysql_drop_db()</i>	Menghapus <i>database MySQL</i>
16	<i>Mysql_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i>
17	<i>Mysql_list_tables()</i>	Memperoleh daftar nama tabel dalam suatu <i>database</i>
18	<i>Mysql_list_fields()</i>	Memperoleh daftar nama kolom dalam suatu <i>database</i>
19	<i>Mysql_fetch_assoc()</i>	Mendapatkan <i>array</i> baris dari suatu <i>recordset</i>
20	<i>Mysql_fetch_lengths()</i>	Mendapatkan panjang baris pada setiap isi <i>field</i>
21	<i>Mysql_fetch_object()</i>	Menghasilkan baris dari <i>recordset</i> sebagai sebuah objek
22	<i>Mysql_field_len()</i>	Mendapatkan informasi panjang maksimum <i>field</i> dalam sebuah <i>recordset</i>
23	<i>Mysql_field_name()</i>	Mendapatkan informasi nama <i>field</i> dalam <i>recordset</i>
24	<i>Mysql_ping()</i>	Memeriksa koneksi <i>server</i> dan akan mencoba untuk melakukan koneksi ulang jika koneksi terputus

(Sumber: Kadir (2008:360) dan Community eWolf (2012:157-168))

2.4.2.3. Keistimewaan *MySQL*

MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas. *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac Os X Server*, *Solaris*, *Amiga*, dan masih banyak lagi.



2. Perangkat lunak sumber terbuka. *MySQL* didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*. *MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.
5. Ragam tipe data. *MySQL* memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. *MySQL* memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol *TCP/IP, Unix socket (UNIX)*, atau *Named Pipes (NT)*.
10. Lokalisasi. *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. *MySQL* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. *MySQL* dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.



13. Struktur tabel. *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya seperti *PostgreSQL* ataupun *Oracle*.

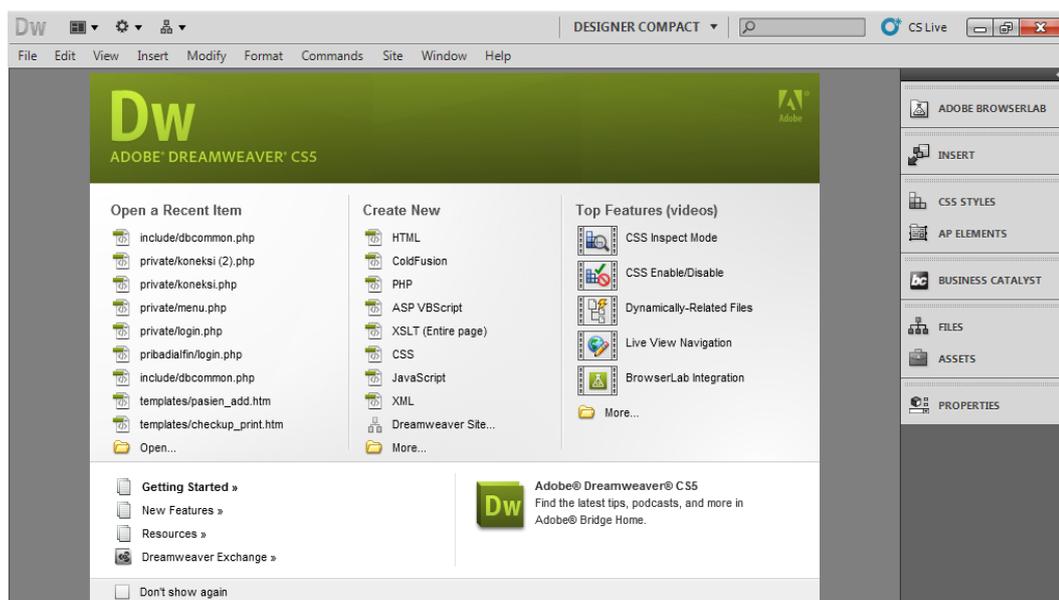
2.4.3. Sekilas Tentang Adobe Dreamweaver CS5

2.4.3.1. Pengertian Adobe Dreamweaver CS5

Madcoms (2010:1), “*Dreamweaver* merupakan *software* aplikasi yang digunakan sebagai *HTML* editor profesional untuk mendesain *web* secara visual. Aplikasi *dreamweaver* dikenal dengan istilah *WYSIWYG* (*What You See Is What You Get*), yang intinya adalah kita tidak harus berurusan dengan *tag-tag HTML* untuk membuat sebuah situs”. *Dreamweaver* juga memberikan keleluasaan untuk menggunakannya sebagai media penulisan bahasa pemrograman *web*.

Adobe terus mengeluarkan beberapa varian *Dreamweaver* yang semakin berkembang, salah satunya yaitu *Dreamweaver CS5*. Ada beberapa fitur baru yang dapat ditemukan pada versi ini, diantaranya: integrasi dengan *Adobe Business Catalyst*, integrasi *Adobe Browser Lab*, Integrasi dengan *CMS*, pengecekan *CSS*, dan lain-lain.

2.4.3.2. Starting Page



Gambar 2.2. Tampilan awal Adobe Dreamweaver CS5



Jendela diatas menampilkan 4 menu pilihan, yaitu:

1. *Open a Recent Item*

Menu ini menampilkan daftar nama dokumen yang pernah dibuka oleh *Adobe Dreamweaver CS5*. Dokumen tersebut dapat dibuka dengan cara memilih nama dokumen yang ada.

2. *Create New*

Menu ini digunakan untuk membuat dokumen *web* yang baru. Dokumen *web* dapat dibuat dengan beberapa tipe yang disediakan oleh *Adobe Dreamweaver CS5*.

3. *Top Features* (video)

Merupakan fitur yang disediakan *Dreamweaver* melalui *internet*.

4. *Help online*

Merupakan menu pilihan untuk menampilkan fasilitas *help* secara *online* melalui *internet*.

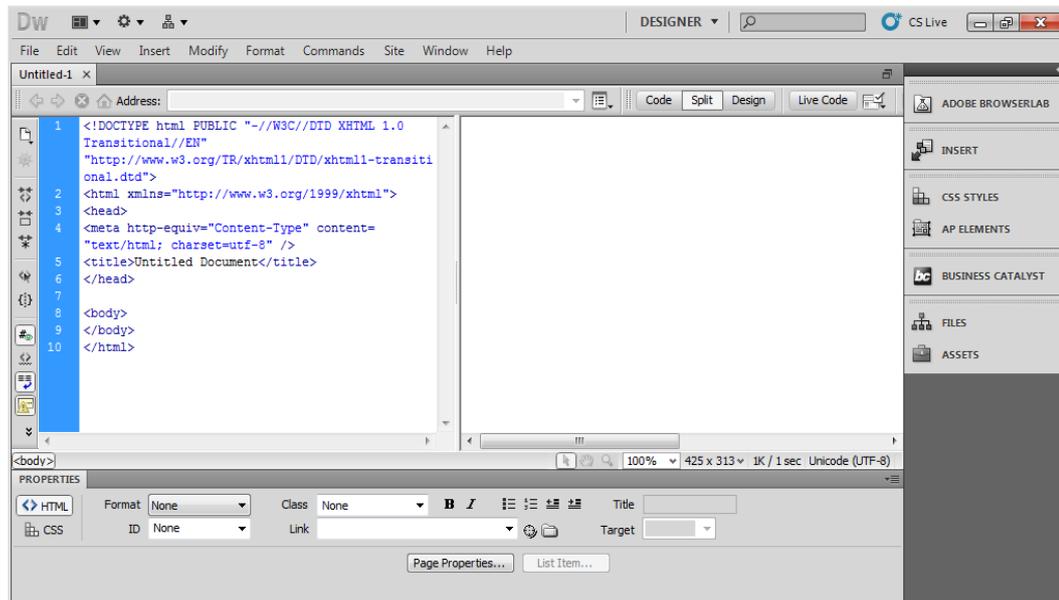
2.4.3.3. Ruang Kerja *Adobe Dreamweaver CS5*

Saat pertama kali menjalankan *Dreamweaver* dengan, akan ditampilkan sebuah kotak dialog *Workspace Setup* yang digunakan untuk memilih tampilan ruang kerja.

1. ***Designer Workspace***, adalah sebuah penggabungan ruang kerja menggunakan MDI (*Multiple Document Interface*), dimana semua jendela *document* dan panel-panel tergabung di dalam satu jendela aplikasi yang benar, dengan sisi panel group di sebelah kanan.

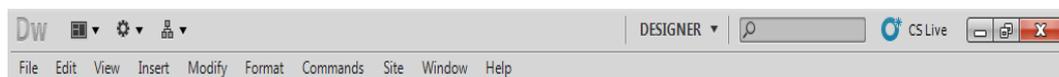
2. ***Coder Workspace***, adalah ruang kerja yang tergabung sama, tapi letak sisi panel group berada di sebelah kiri.

Ruang kerja *Dreamweaver CS5* memiliki komponen-komponen yang memberikan fasilitas dan ruang untuk menuangkan kreasi saat bekerja, seperti *Application Bar*, *Toolbar Document*, *Document Window*, *Workspace Switcher*, *Panel Groups*, *Tag Selector*, *Property Inspector*, *Toolbar Browser Navigation*.



Gambar 2.3. Tampilan Ruang Kerja *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 2.4. *Application Bar* pada *Dreamweaver CS5*

2. **Toolbar Document**, berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *design* atau tampilan kode. Selain itu juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.



Gambar 2.5. *Toolbar Document* pada *Dreamweaver CS5*

3. **Document Window** (Jendela Dokumen), lembar kerja tempat membuat dan mengedit desain halaman *web*.



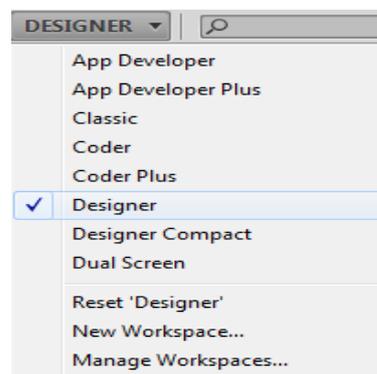
```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Untitled Document</title>
6 </head>
7
8 <body>
9 </body>
10 </html>

```

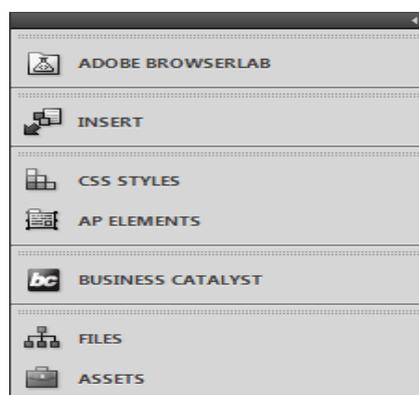
Gambar 2.6. Document window pada Dreamweaver CS5

4. **Workspace Switcher**, digunakan untuk mengubah tampilan ruang kerja (*workspace*) Dreamweaver CS5



Gambar 2.7. Workspace Switcher pada Dreamweaver CS5

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



Gambar 2.8. Panel Groups pada Dreamweaver CS5

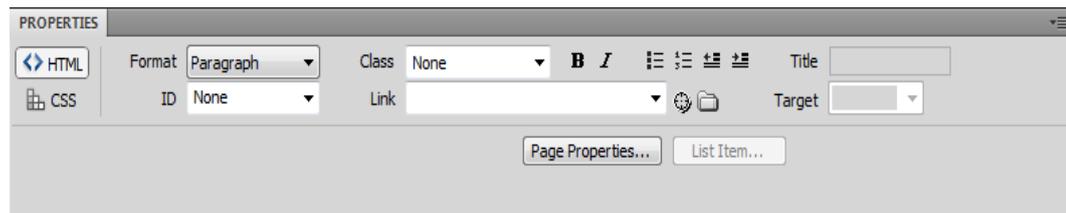


6. **Tag Selector**, terletak di bagian bawah jendela dokumen, satu baris dengan *status bar*. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut. *Tag Selector* juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja desain.



Gambar 2.9. *Tag Selector* pada *Dreamweaver CS5*

7. **Property Inspector**, digunakan untuk melihat dan mengubah berbagai *property* objek atau teks pada jendela desain. *Property* untuk satu objek dengan objek lainnya selalu berbeda-beda. Jendela ini tidak dapat diuraikan pada tampilan jendela kode.



Gambar 2.10. *Property Inspector* pada *Dreamweaver CS5*

8. **Toolbar Browser Navigation**, toolbar ini merupakan *toolbar* baru yang ada di dalam *Dreamweaver CS5* dan letaknya tepat berada di atas jendela dokumen. Toolbar ini berisi tombol-tombol yang digunakan sebagai navigasi di dalam *browser*.



Gambar 2.11. *Toolbar Browser Navigation* pada *Dreamweaver CS5*