BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

2.1.1. Sistem

Sistem berasal dari bahasa latin *(systema)* dan bahasa yunani *(sustema)* adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, dimana suatu model matematika seringkali bisa dibuat. Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak (Husda, 2012 : 111).

Sedangkan pendapat lain menurut (Sutabri, 2012) mengatakan, Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Sedangkan menurut buku *Conceptual, Structure and Development*, "Sistem dapat bersifat abstrak atau fisis. Sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsep-konsep yang saling tergantung (Ali, 2010: 8).

Jadi penulis dapat menyimpulkan secara keseluruhan bahwa sistem adalah kumpulan atau elemen yang saling terhubung dengan gagasan dan informasi yang saling melengkapi.

2.1.1.1. Karakteristik Sistem

Karakterisitik Sistem menurut (Husda, 2012: 112) dapat dibagi menjadi beberapa bagian yaitu:

1. Komponen

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian dari sistem.

2. Batas Sistem

   Batas Sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem *(Environment)*

   Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem *(Interface)*

   Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya.

5. Masukan Sistem

   Masukan Perawatan (*Maintenance Input)* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi.

   Masukan Sinyal *(Signal Input)* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

   Merupakan hasil dari energi yang diolah sistem, meliputi:

   Keluaran yang berguna, contohnya Informasi yang dikeluarkan oleh komputer.

   Keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan, contoh panas yang dikeluarkan komputer.

7. Pengolah Sistem

   Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

8. Tujuan Sistem

   Suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenasi sasaran atau tujuannya.

2.1.2. Informasi

Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi sipenerima dan mempunyai nilai yang nyata atau dapat dirasakan manfaatnya dalam keputusan-keputusan yang akan datang (Ali, 2010: 10).

Secara umum menurut pendapat (Husda, 2012: 117), informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-

kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk lebih berarti bagi yang menerimanya.

Dari pendapat para ahli tentang pengertian informasi, penulis dapat menyimpulkan pengertian informasi adalah data yang telah diolah menjadi sesuatu yang berguna bagi penerima dan memiliki arti yang sangat penting dalam mengambil sebuah keputusan.

Adapun menurut (Husda, 2012), fungsi-fungsi informasi adalah sebagai berikut:

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

2.1.3. Sistem Informasi

Sistem Informasi merupakan suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. Sistem Informasi dapat juga dikatakan sebagai suatu totalitas terpadu terdiri dari prosedur, tenaga pengolah *(brainware)*, perangkat lunak *(software)*, perangkat keras *(hardware)*, pangkalan data *(database)*, perangkat telekomunikasi *(telecommunication)* yang saling ketergantungan dan saling menentukan dalam rangka meneyediakan informasi informasi untuk mendukung proses pengambilan keputusan (Ali, 2010: 13).

Pendapat lain mengatakan, Sistem Informasi didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang,

fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan (Husda, 2012: 119).

Dari pendapat para ahli yang telah disebut, penulis menyimpulkan pengertian sistem informasi adalah suatu kumpulan atau komponen dalam sebuah organisasi yang saling berkaitan dalam menyediakan informasi, sehingga informasi tersebut bisa menjadi dasar dalam pengambilan keputusan yang tepat.

2.1.4. SDLC

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 (Rosa & Shalahuddin, 2011: 23).

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut (Rosa & Shalahuddin, 2011: 23):

1) Inisiasi *(Initation)*

   Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2) Pengembangan Konsep Sistem *(System Concept Development)*

   Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3) Perencanaan *(Planning)*

   Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya *(resources)* yang dibutuhkan untuk memperoleh solusi.

4) Analisis Kebutuhan *(Requirements Analysis)*

   Menganalisis kebutuhan pemakai sistem perangkat lunak *(user)* dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5) Desain *(design)*

   Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem focus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6) Pengembangan *(development)*

   Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau file pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program, peninjauan pengujian.

7) Integrasi dan Pengujian *(Integration and test)*

   Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan

oleh staf penjamin kualitas *(quality assurance)* dan user. Menghasilkan laporan analisa pengujian.

8) Implementasi *(Implementation)*

   Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada user) dan menjalankan resolusi dari permasalahan yang teridentifikai dari fase integrasi dan pengujian.

9) Operasi dan Pemeliharaan *(Operations and Maintenance)*

   Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada user), termasuk implementasi akhir dan masuk pada proses peninjauan.

10) Disposisi *(Disposition)*

    Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas user.

2.1.5. UML

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG *(Object Management Group)*. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification*, UML *Infrastructure*, UML *Superstructure*, dan *Object Constraint Language* (OCL) (Rosa dan Shalahuddin, 2011: 118).

Penjelasan pembagian kategori dan macam-macam diagram (Rosa dan Shalahuddin, 2011: 121):

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakukan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

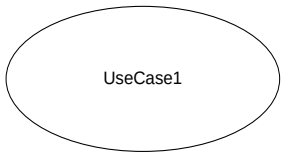
2.1.5.1. Use Case Diagram

Use Case atau diagram use case merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor 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.

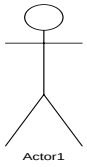
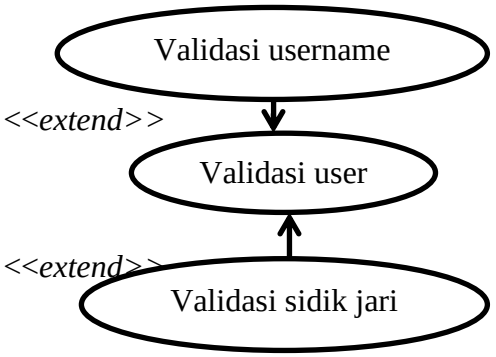
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut *actor* dan *use case* (Rosa dan Shalahuddin, 2011: 130).

1. Aktor merupakan 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 actor adalah gambar orang, tapi actor belum tentu merupakan orang.
2. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau actor.

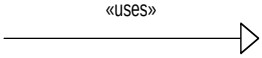
Tabel 2.1. Simbol-simbol *use case*

SIMBOL	DESKRIPSI
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor; biasanya dinyatakan dengan m enggunakan kata kerja di awal frase nama *use case*

Tabel 2.1. Simbol-simbol *use case*

SIMBOL	DESKRIPSI
Aktor/*actor*  Nama *actor*	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi. Biasanya dinyatakan dengan menggunakan kata benda di awal frase nama actor
Asosiasi/*association* 	Komunikasi antar actor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan *actor*
Ekstensi/*extend* <<*extend*>> 	Relasi *use case* tambahkan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan;  Arah panah mengarah pada *use case* yang ditambahkan.

Tabel 2.1. Simbol-simbol *use case*

SIMBOL	DESKRIPSI
Generalisasi/*generalization* «uses» 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Arah panah mengarah pada use case yang menjadi generalisasi (umum).
Menggunakan/*include/user* <<include>> <<uses>> «uses» 	Relasi *use case* tambahkan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. Ada dua sudut pandang yang cukup besar mengenai include di *use case* : Include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan Include berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan.

2.1.5.2. Class Diagram

Menurut (Rosa & Shalahuddin, 2011:130), *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1) Kelas *main*

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2) Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

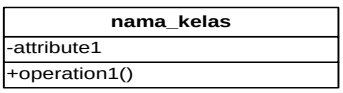
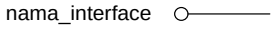
3) Kelas yang diambil dari pendefinisian *use case*

   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

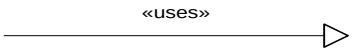
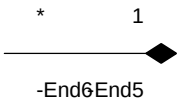
4) Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Tabel 2.2. Simbol-simbol *class* diagram

SIMBOL	DESKRIPSI
Kelas 	Kelas pada struktur sistem
Antarmuka/interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.

Tabel 2.2. Simbol-simbol *class* diagram

SIMBOL	DESKRIPSI
Asosiasi/*association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity.*
Asosiasi berarah / *directed association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
Ketergantungan/*dependency*	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua bagian (*whole-part*)

2.1.5.3. Activity Diagram

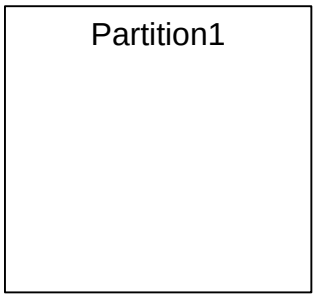
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa & Shalahuddin, 2011: 130):

1. Rancangan prose bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktiitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Tabel 2.3. Simbol-simbol *activity* diagram

SIMBOL	DESKRIPSI
Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

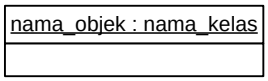
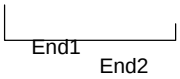
Tabel 2.3. Simbol-simbol *activity* diagram

SIMBOL	DESKRIPSI
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 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.1.5.4. Object Diagram

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.

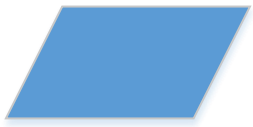
Tabel 2.4. Simbol-simbol diagram objek

SIMBOL	DESKRIPSI
Objek 	Objek dari kelas yang berjalan saat sistem dijalankan
Link 	Relasi antar objek

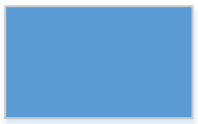
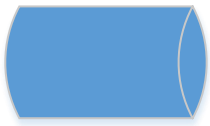
2.1.6. Flowchart

Adapun simbol-simbol *flowchart* adalah sebagai berikut:

Tabel 2.5. Simbol-simbol *flowchart*

Nama	Simbol	Keterangan
Input/output		Simbol ini digunakan untuk mewakili data *input/output*

Tabel 2.5. Simbol-simbol *flowchart*

Nama	Simbol	Keterangan
Proses		Simbol proses ini digunakan untuk mewakili suatu proses
Simbol Dokumen		Menunjukkan dokumen *input* dan *output* baik untuk proses manual, mekanik atau computer.
Terminal		Simbol ini digunakan untuk menunjukkan awal dan akhir dari suatu proses
Kegiatan Manual		Menunjukkan pekerjaan manual
Penyimpanan *Harddisk*		Menunjukkan *input/output* menggunakan *harddisk*

Tabel 2.5. Simbol-simbol *flowchart*

Nama	Simbol	Keterangan
Decision Symbol		Digunakan untuk suatu penyeleksian kondisi didalam program
Sambungan pada halaman lain		Simbol ini digunakan untuk menunjukan sambungan dimana sambungannya pada halaman lain
Sambungan pada satu halaman		Simbol ini digunakan untuk menunjukkan sambungan dari bagian *flowchart* dimana bagian tersebut ada pada halaman yang sama

2.2. Tinjauan Teori Khusus

2.2.1. Akademik

Akademik adalah suatu bidang yang mempelajari tentang kurikulum atau pembelajaran dalam fungsinya untuk meningkatkan pengetahuan dalam segi pendidikan/pembelajaran yang dapat dikelola oleh suatu sekolah atau lembaga pendidikan (Liatmaja, 2013: 59).

akademik adalah keadaan orang-orang bisa menyampaikan dan menerima gagasan, pemikiran, ilmu pengetahuan, dan sekaligus dapat mengujinya secara jujur, terbuka, dan leluasa (Rismawati, 2014: 322).

2.2.2. Sistem Informasi Akademik

Sistem Informasi Akademik (SIA) dihimpun dari berbagai macam data yang dikelola dan diproses se-automatis mungkin dengan alat dan metoda sehingga menghasilkan informasi yang diperlukan bagi terlaksananya kegiatan akademis. Sistem ini dibagi ke dalam beberapa subsistem yakni: 1) Seleksi dan registrasi mahasiswa baru, 2) Kurikulum dan bidang studi, 3) Perkuliahan, 4) tugas, 5) ujian, 6) Pengelolaan dan pengembangan dosen, 7) Kelulusan, wisuda, alumni. Sejumlah sistem informasi lain bisa dikaitkan ke SIA, seperti misalnya: sistem informasi pustaka dan administrasinya (sistem perpustakaan?), sistem informasi kemahasiswaan (ekstra kurikuler) dan sebagainya (Lukum, 2013: 532).

Sistem Informasi Akademik adalah aplikasi yang dirancang dan dibuat untuk mengolah data-data yang berhubungan dengan informasi akademik, meliputi data mahasiswa, karyawan, rekaman prestasi, kurikulum, dan jadwal perkuliahan. Setiap anggota civitas akademika mempunyai tingkat akses yang berbeda-beda terhadap SIA. Tingkat akses yang dimaksud disesuaikan dengan kebutuhan dan tingkat kepentingan dari masing-masing pengguna sistem (Adnyana, 2016: 51).

2.2.3. Website

Web merupakan media informasi berbasis jaringan komputer yang dapat diakses di mana saja dengan biaya relative murah. *Web* merupakan bentuk implementasi dari bahasa pemrograman *web*. Sejarah perkembangan bahasa pemrograman *web* diawali dengan munculnya HTML *(Hypertext Markup Language)*, yang kemudian dikembangkan dengan munculnya CSS *(Cascading Style Sheet)* yang bertujuan untuk memperindah tampilan *website* (Hermawan, 2009: 2).

World Wide Web (WWW), lebih dikenal dengan *web*, merupakan salah satu layanan yang didapat oleh pemakai computer yang terhubung ke internet. *Web* pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext*, pemakai dituntun untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen *web* yang ditampilkan dalam *browser web* (Sidik, 2009: 1).

Web memudahkan pengguna komputer untuk berinteraksi dengan pelaku internet lainnya dan menelusuri (informasi) di internet. Selain itu *web* telah diadopsi oleh perusahaan sebagai sebagian dari strategi teknologi informasinya, karena beberapa alasan (Sidik, 2009: 2):

1. Akses informasi mudah;
2. *Setup* server lebih mudah;
3. Informasi mudah distribusikan; dan
4. Bebas *platform*; informasi dapat disajikan oleh *browser web* pada sistem

operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan.

2.2.4. PHP

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 semua sintaks dan perintah program yang anda tulis akan sepenuhnya dijalankan oleh sever, tetapi dapat disertakan pada halaman HTML biasa (Hermawan, 2009: 3).

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman *Open Source* yang digunakan untuk membuat sebuah aplikasi *web* dan dapat disatukan dengan HTML. PHP berbeda dengan Javascript. PHP merupakan pemrograman *server side* sedangkan Javascript merupakan pemrograman *client side*. Kode program PHP akan dieksekusi oleh server dan hasil eksekusi tersebut akan ditampilakn kepada client, berbeda dengan Javascript, kode program akan dieksekusi oleh browser client (Prakoso, 2015: 1.1).

Beberapa peningkatan fungsionalitas dan penambahan beberapa fitur pada PHP 5 adalah (Hermawan, 2009: 6):

1. Peningkatan kemampuan pemrograman berorientasi objek (OOP).
2. *Exception handling*, sistem standarisasi logika atas pemberitahuan kesalahan pemrograman yang belum didukung oleh PHP 4.
3. Peningkatan terhadap penanganan string lebih lanjut.
4. Peningkatan dukungan terhadap XML dan *Web Service*, yang menggunakan libxm2.

5. Dukungan terhadap SQLite *database* server.

2.2.5. MYSQL

Salah satu *database* untuk server adalah MySQL, jenis database ini sangat popular dan digunakan pada banyak website di internet sebagai bank data. MySQL menggunakan SQL dan bersifat *free* (gratis). Selain itu MySQL dapat berjalan di berbagai *platform*, antara lain linux dan windows (MADCOMS, 2011: 288).

MySQL merupakan sistem *database* yang banyak digunakan untuk pengembangan aplikasi *Web*. Alasannya mungkin karena gratis, pengelolaan datanya sederhana, memiliki tingkat keamanan yang bagus, mudah diperoleh dan lain-lain (Raharjo, 2010: 216).

2.2.6. HTML

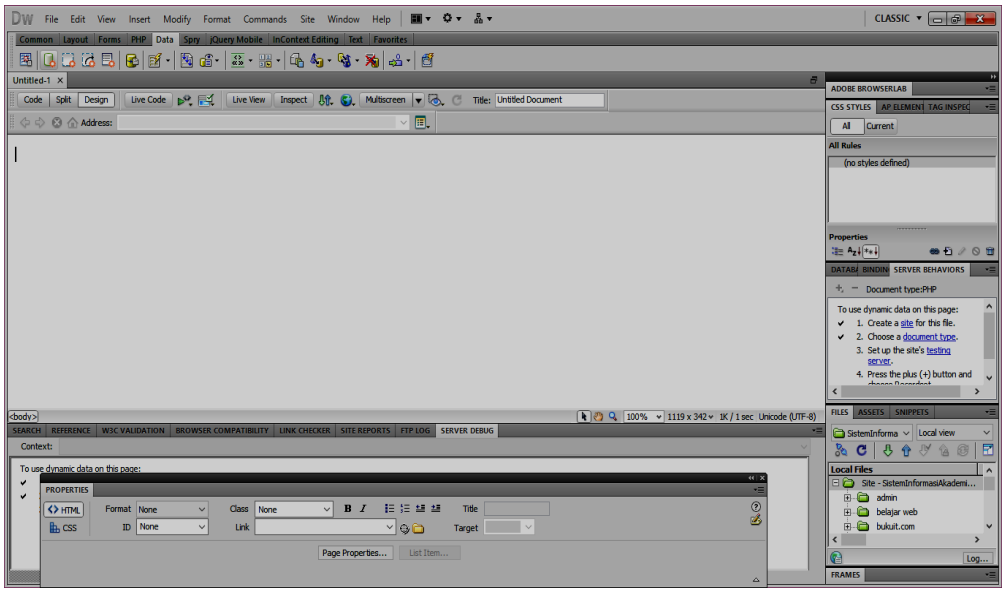
HTML kependekan dari *Hyper Text Markup Language*. Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan dalam browser *web surfer*. Dokumen ini umumnya berisi informasi atau *interface* aplikasi didalam internet (Sidik, 2009: 9).

HTML merupakan file teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke *user* melalui suatu aplikasi *web browser*. Setiap informasi yang tampil di web selalu dibuat menggunakan kode HTML. Oleh karena itu, dokumen HTML sering disebut juga sebagai *web page* (halaman *web*). Untuk membuat dokumen HTML, kita tidak tergantung pada aplikasi

tertentu, karena dokumen HTML dapat dibuat menggunakan aplikasi *Text Editor* apapun, bisa *Notepad*, *Emacs*, atau *Vi Editor* (Raharjo , 2010: 286).

HTML 5 merupakan generasi baru dari HTML, yang dirancang untuk memperbaiki teknologi HTML versi sebelumnya agar dapat mendukung teknologi multimedia terbaru dan tipe isi halaman *web* lainnya (Raharjo, 2011: 6).

2.2.7. Dreamweaver CS5



Gambar 2.1. Tampilan adobe dreamweaver cs5

1. *Aplication Bar*, berada di bagian paling atas jendela aplikasi Dreamweaver CS5. Baris ini berisi tombol workspace (*workspace switcher*), *CS Live*, menu, dan aplikasi lainnya.
2. *Toolbar Document*, berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen.
3. Jendela Dokumen, adalah lembar kerja tempat anda membuat dan mengedit desain halaman *web*.

4. *Workspace Switcher*, digunakan untuk mengubah tampilan ruang kerja (*workspace*) Dreamweaver CS5.
5. *Panel Groups*, adalah kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya.
6. *Tag Selector*, diletakkan 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.
7. *Property Inspector*, digunakan untuk melihat dan mengubah berbagai property objek atau teks pada jendela *design*.
8. *Toolbar Standard*, baris toolbar ini berisi tombol-tombol yang mewakili perintah pada menu File dan Edit, diantaranya perintah *New*, *Open*, *Save*, *Save All*, *Cut*, *Copy*, *Paste*, *Undo*, dan *Redo*.
9. *Toolbar Style Rendering* secara default disembunyikan. Toolbar ini berisi tombol-tombol untuk menampilkan desain anda dalam media yang berbeda.
10. *Toolbar Coding*, berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Toolbar ini hanya tampil pada jendela code.

11. *Toolbar Browser Navigation*, toolbar ini adalah toolbar baru yang ada di dalam Dreamweaver CS5, dan letaknya tepat berada di atas jendela dokumen.