

BAB II

LANDASAN TEORI

2.1 Definisi Sistem

Konsep dasar sistem memiliki banyak perbedaan definisi dari para ahlinya. Definisi mengenai sistem sesuai pendekatan sistem terdiri berasal sistem yang menekankan di mekanisme dan sistem yang menekankan pada elemennya. Pendekatan sistem yang menekankan pada prosedurnya didefinisikan sang Jerry Fitzgerald. Menurutnya, sistem bisa didefinisikan sebagai suatu jaringan kerja yang terdiri dari prosedur-prosedur yg saling berhubungan, kemudian berkumpul beserta-sama buat melakukan atau menuntaskan kegiatan serta mencapai suatu sasaran tertentu. Sedangkan sistem yang menekankan pada elemennya di definisikan sang Robert G. Murdick, yaitu sistem terdiri berasal elemen-elemen yg saling terintegrasi dengan tujuan yang sama buat mencapai sasaran yg di harapkan[4].

2.2 Defenisi Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataannya yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (event) adalah kejadian yang terjadi pada saat tertentu.

Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi sipenerima dan mempunyai nilai nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang. Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berguna bagi pengguna yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi”.

Informasi diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”. Jadi Informasi adalah data yang diproses kedalam bentuk yang lebih berarti bagi penerima dan berguna dalam pengambilan keputusan, sekarang atau untuk masa yang akan datang[5].

2.3 Defenisi Sistem Informasi

Sistem informasi merupakan suatu kumpulan atau himpunan dari unsur yang saling terorganisasi, saling berinteraksi dan saling bergantung satu sama lain. Sistem informasi adalah suatu rangkaian sistem yang dikelompokkan dalam suatu organisasi yang terdiri dari sekumpulan komponen baik yang berbasis komputer maupun manual yang dibuat untuk menghimpun dan menyiapkan data-data yang berisikan informasi keluaran untuk pemakai[6].

2.4 Pengertian Penerimaan Peserta Didik Baru (PPDB) Online

PPDB Online adalah sebuah sistem yang dirancang untuk melakukan otomatisasi seleksi penerimaan siswa baru (PSB), mulai dari proses pendaftaran, proses seleksi hingga pengumuman hasil seleksi, yang dilakukan secara online dan berbasis waktu nyata (realtime), yang bertujuan secara umum adalah untuk meningkatkan mutu layanan pendidikan, menciptakan sistem penerimaan siswa baru yang terintegrasi, akurat dan transparan, melaksanakan penerimaan siswa baru dengan lebih praktis dan efisien, menyediakan basis data sekolah yang akurat, memberi fasilitas akses informasi bagi masyarakat dengan cepat, mudah dan akurat[7].

2.5 Sosial Media Marketing (SMM)

Merujuk dari asal katanya, social media marketing (pemasaran media sosial) merupakan gabungan dari konsep social media dan marketing. Pada konteks pemasaran, media sosial merupakan sarana (*platform*) dimana orang-orang bisa membangun jejaring (*network*), berbagi informasi dan bahkan perasaan (*sentiment*). Social media marketing merupakan suatu bentuk pemasaran langsung ataupun tidak langsung yang digunakan untuk membangun kesadaran, pengakuan, daya ingat, dan tindakan untuk merek, bisnis, produk, orang, atau entitas lainnya dan dilakukan dengan menggunakan alat dari web sosial seperti blogging, microblogging, socialnetworking, social bookmarking, dan content sharing[8].

2.5.1. Facebook

Perkembangan Facebook sebagai salah satu situs yang paling mencolok dibandingkan dengan layanan lain yaitu jejaring sosial atau social network. Jejaring

sosial pada umumnya selalu dikaitkan dengan nilai-nilai, visi, ide, teman dan lainnya[9].

2.6 Website

Website adalah kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (*Uniform Resource Locator*) yang dapat diakses semua pengguna internet dengan cara mengetikkan alamatnya[10]. *Website* adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman[11].

2.7 Peralatan Pendukung (*tools system*)

Di dalam merancang sebuah sistem diperlukan suatu peralatan yang dapat mendukung terciptanya suatu rancangan. Peralatan pendukung (*tools system*) merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol, lambang, diagram yang menunjukkan secara tepat arti dan fungsinya[12].

2.7.1 ERD

Entity Relationship Diagram (ERD) merupakan sebuah cara untuk menggambarkan sebuah basis data yang menggunakan simbol-simbol beserta hubungan antara simbol-simbol tersebut. “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional[13].

Elemen-elemen diagram hubungan entitas adalah sebagai berikut :

a. *Entity*

Entity adalah apa saja yang ada di dalam sistem, baik sistem nyata maupun abstrak tempat data tersimpan. Entitas dinamakan dengan kata benda dan dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

b. *Relationship*

Relationship diberi nama kata kerja yang dapat memudahkan untuk

membaca relasinya. Relationship adalah hubungan yang terjadi antars entitas.

c. Attribute value





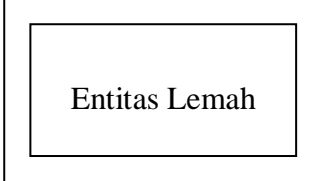
Attribut value atau nilai atribut adalah suatu occurrence tertentu dari sebuah attribute didalam suatu entity atau relationship.

d. Derajat Relationship (*Relationship Degree*)

Derajat relationship adalah jumlah entitas yang berpartisipasi dalam satu relationship.

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

Tabel 2.1 Simbol-Simbol ERD

No	Simbol	Deskripsi
1		Jenis Entitas dapat berupa suatu elemen lingkungan, sumber daya atau transaksi yang <i>field-fieldnya</i> dipergunakan dalam aplikasi program.
2		Menunjukkan nama relasi antar satu entitas dengan entitas lainnya
3		Atribut adalah karakteristik dari sebuah entitas
4		Menunjukkan hubungan (keterkaitan) antar entitas.
5		Entitas yang kemunculannya tergantung dari entitas lain yang lebih kuat.

2.7.2 UML

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. 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[14].



“UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek” diantaranya :

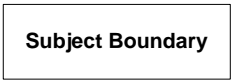

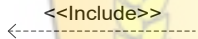


1. *Use Case Diagram*

Use Case Diagram bersifat statis, diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

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

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1	 Actor	<i>Actor/Role</i> , yaitu orang atau sistem lain yang berinteraksi dengan sistem saat ini.
2	 Use Case	<i>Use Case</i> adalah bagian utama dari fungsionalitas sistem. Bisa <i>extend</i> (memperluas) <i>use case</i> lainnya.


		Ditempatkan di dalam <i>system boundary</i> (batasan sistem). Dilabeli dengan kata kerja-frase kata benda.
3		<i>Subject Boundary</i> , Berisi nama dari sistem yang diletakkan di dalam atau dibagian atas <i>boundary</i> . Mewakili ruang lingkup sistem. <i>Actor</i> berada di luar ruang lingkup sistem.
4		<i>Association Relationship</i> , menghubungkan <i>actor</i> dengan <i>use case</i> . Menunjukkan komunikasi dua arah (menunjukkan komunikasi satu arah jika menggunakan tanda panah). Tanda * untuk keragaman dari asosiasi (<i>multiplicity of the association</i>). Namun umumnya hanya digambarkan garis saja.
5		<i>Include Relationship</i> , merupakan satu <i>use case</i> dalam <i>use case</i> lainnya, perilaku (<i>behavior</i>) yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya. Tanda panah mengarah dari <i>base use case</i> (pusat) menuju ke <i>use case</i> yang di-include.
6		<i>Extend Relationship</i> , memperluas <i>use case</i> untuk memasukkan perilaku opsional. Tanda panah mengarah dari <i>use case</i> tambahan ke <i>base use case</i> (pusat).
7		<i>Generalization Relationship</i> , mewakili <i>use case</i> khusus untuk <i>use case</i> yang lebih umum. Tanda panah mengarah dari <i>use case</i> khusus (<i>specialized</i>) ke <i>use case</i> yang umum.

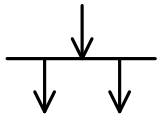
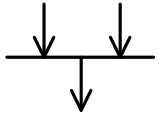

2. Activity Diagram

Activity diagram secara esensial mirip dengan *flowchart* atau diagram alur yang menunjukkan aliran kendali dari sebuah aktivitas ke aktivitas lainnya. Dalam

activity diagram terdapat aksi atau aktivitas, *activity nodes*, *flows* atau aliran, dan objek. Simbol-simbol yang dipakai dalam *activity* diagram yaitu:

Tabel 2.3 Simbol-Simbol *Activity* Diagram

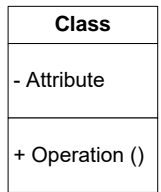
No	Simbol	Deskripsi
1		<i>Action</i> , perilaku yang sederhana dan tidak dapat diuraikan. Dilabeli dengan namanya.
2		<i>Activity</i> , digunakan untuk mewakili sekumpulan tindakan (<i>action</i>). Dilabeli dengan namanya.
3		<i>Object Node</i> , digunakan untuk mewakili sebuah objek yang terhubung ke sekumpulan arus objek. Dilabeli dengan nama <i>class</i> -nya.
4		<i>Control Flow</i> , menunjukkan urutan eksekusi.
5		<i>Object Flow</i> , menunjukkan aliran suatu objek dari satu aktivitas (atau tindakan) ke aktivitas (atau tindakan) lain.
6		<i>Initial Node</i> , menggambarkan awal dari serangkaian tindakan atau kegiatan.
7		<i>Final-Activity Node</i> , digunakan untuk menghentikan semua arus kontrol dan arus objek dalam suatu aktivitas (atau tindakan).
8		<i>Final-Flow Node</i> , digunakan untuk menghentikan aliran kontrol atau aliran objek tertentu.
9		<i>Decision Node</i> , digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Dilabeli dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.
10		<i>Merge Node</i> , digunakan untuk menyatukan kembali berbagai jalur keputusan yang dibuat menggunakan simpul keputusan.

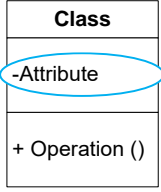
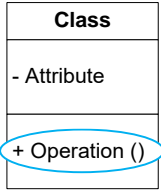

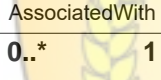


11		<i>Fork Node</i> , digunakan untuk membantu perilaku menjadi seperangkat aktivitas yang paralel atau bersamaan dari aktivitas (atau tindakan).
12		<i>Join Node</i> , digunakan untuk menyatukan kembali serangkaian arus aktivitas (atau tindakan) yang paralel atau bersamaan.
13		<i>Swimlane</i> , digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan kegiatan individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan). Dilabeli dengan nama individu atau objek yang bertanggung jawab.

3. Class Diagram

Class adalah sebuah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu system. *Class* diagram menggambarkan struktur dan deskripsi kelas, package beserta hubungan satu sama lain. Simbol-simbol yang digunakan dalam class diagram yaitu:

Tabel 2.4 Simbol-Simbol *Class* Diagram

No	Simbol	Deskripsi
1		<i>Class</i> , mewakili orang, tempat, atau hal-hal yang dibutuhkan sistem untuk menangkap dan menyimpan informasi. Memiliki nama yang diketik dengan huruf yang tebal dan berpusat dibagian atas kotak. Memiliki daftar atribut di kotak tengahnya. Memiliki daftar operasi di kotak bawahnya. Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas.



2	 <pre> classDiagram class Class { -Attribute +Operation() } </pre>	<p><i>Attribute</i>, merupakan properti yang menggambarkan keadaan suatu objek. Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.</p>
3	 <pre> classDiagram class Class { -Attribute +Operation() } </pre>	<p><i>Operation</i>, mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas. Dapat diklasifikasikan sebagai konstruktor, permintaan, atau operasi pembaruan. Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.</p>
4		<p><i>Generalization</i>, mewakili hubungan sejenis diantara beberapa kelas.</p>
5		<p><i>Association</i>, mewakili hubungan antar beberapa kelas atau kelas dan dirinya sendiri. Dilabeli menggunakan frasa kata kerja atau nama peran, yang lebih baik mewakili hubungan. Dapat ada di antara satu atau beberapa kelas. Berisi simbol multisplitas, yang mewakili waktu minimum dan maksimum <i>instance</i> kelas dapat dikaitkan dengan <i>instance</i> kelas terkait.</p>
6		<p><i>Aggregation</i>, merupakan hubungan sebagian (a part of) yang logis diantara beberapa kelas atau suatu kelas dengan kelas itu sendiri. <i>Aggregation</i> terbuat dari objek-objek yang bisa dibagi atau ditukar. Angka disebelah kanan boleh satu, boleh juga banyak (*).</p>
7		<p><i>Composition</i>, mewakili hubungan fisik antara beberapa kelas atau kelas dan kelas itu sendiri. <i>Composition</i> terdiri dari objek-objek yang tidak bisa dibagi atau ditukar dan hidup selama <i>composite</i> objek hidup. Angka disebelah kanan hanya boleh 1.</p>

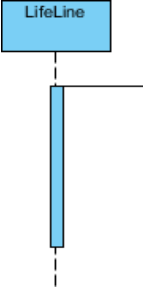
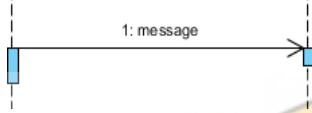
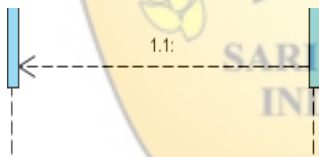

4. Sequence Diagram


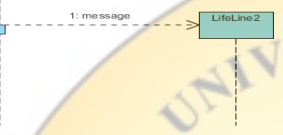

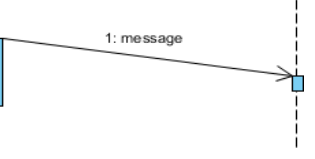
Sequence Diagram menggambarkan interaksi antara objek di dalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu, menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. *Collaboration* diagram juga menggambarkan interaksi antara objek seperti *sequence* diagram, akan tetapi lebih menekankan pada masing masing objek dan bukan pada waktu penyampaian message. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1.


Simbol-simbol yang digunakan dalam *sequence* diagram dijelaskan pada tabel :

Tabel 2.5 Simbol-Simbol *Sequence* Diagram

No	SIMBOL	DESKRIPSI
1		<i>Actor/Aktor</i> , Jenis peran yang dimainkan oleh entitas yang berinteraksi dengan subjek (Misalnya dengan bertukar sinyal dan data) Di luar subjek (maksudnya adalah batasan/kendala dari aktor bukan bagian dari batasan/kendala dari subjek yang sesuai).
2		<i>Lifeline</i> , atau Garis hidup mewakili peserta individu dalam Interaksi.

3		<p><i>Activation box</i>, atau kotak aktivasi berbentuk sebuah persegi panjang tipis pada <i>lifeline</i>, mewakili periode di mana suatu elemen melakukan operasi.</p> <p>Bagian atas dan bawah dari kotak aktivasi disejajarkan dengan inisiasi dan waktu penyelesaian masing-masing</p>
4		<p><i>Call message</i> atau pesan panggilan merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi.</p> <p><i>Call Message</i> adalah jenis pesan yang mewakili permintaan operasi dari target <i>lifeline</i>.</p>
5		<p><i>Return message</i> atau pesan balik adalah sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi.</p> <p><i>Return message</i> adalah jenis pesan yang mewakili informasi yang dikirimkan kembali ke pengirim pesan atau pemanggil berdasarkan pesan sebelumnya.</p>
6		<p><i>Self message</i> atau pesan mandiri adalah sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi.</p> <p><i>Self message</i> adalah jenis pesan yang mewakili permohonan pesan dari <i>lifeline</i> yang sama.</p>

7		<p><i>Recursive Message</i> atau pesan rekursif merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi.</p> <p><i>Recursive Message</i> merupakan jenis pesan yang mewakili permohonan pesan dari lifeline yang sama. <i>Recursive Message</i> ini menargetkan aktivasi di atas aktivasi tempat pesan itu berasal.</p>
8		<p><i>Create Message</i> merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi.</p> <p><i>Create message</i> atau membuat pesan adalah jenis pesan yang mewakili Instansiasi (target) <i>lifeline</i>.</p>
9		<p><i>Destroy Message</i> merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi.</p> <p><i>Destroy message</i> atau hancurkan pesan merupakan sebuah jenis pesan yang mewakili permintaan untuk menghancurkan siklus hidup target.</p>
10		<p><i>Duration message</i> atau pesan durasi yaitu sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi. Pesan Durasi menunjukkan jarak antara dua batasan waktu untuk permohonan pesan.</p>

11		<p><i>Note</i> atau Catatan (komentar) merupakan catatan atau lampiran berbagai komentar ke elemen. Sebuah komentar tidak memiliki kekuatan semantik, tetapi dapat berisi informasi yang berguna bagi pemodel.</p>
----	---	--

2.8 Bahasa Pemrograman

Bahasa pemrograman adalah sebuah perangkat lunak yang digunakan untuk menerjemahkan atau menuliskan algoritma dalam bentuk teks perintah-perintah yang dapat dimengerti oleh komputer untuk menyelesaikan suatu masalah. Bahasa pemrograman yang dapat menghasilkan file program komputer yang berdiri sendiri disebut dengan *compiler*, sedangkan bahasa pemrograman yang berperan sebagai penerjemah kode-kode program disebut dengan *interpreter*[15].

2.8.1 JavaScript

JavaScript adalah bahasa script yang ditempelkan pada kode HTML dan proses pada sisi klien, sehingga kemampuan dokumen HTML menjadi lebih luas[16].

2.8.2 PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa utama script serverside yang disisipkan pada HTML yang dijalankan di server dan juga bisa digunakan untuk membuat aplikasi desktop.

“PHP merupakan secara umum dikenal sebagai bahasa pemrograman script-script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*”[17].

2.9 Basis Data (*Database*)

Database merupakan komponen terpenting dalam pembangunan sistem informasi, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. Menurut para ahli terdapat pengertian

database antara lain : “*Database* atau sering disebut dengan basis data adalah sekumpulan informasi yang disimpan dalam komputer secara sistematis dan merupakan sumber informasi yang dapat diperiksa menggunakan suatu program komputer”.

“*Database* atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer”[18].

2.9.1 Xampp

XAMPP adalah perangkat lunak yang paling umum digunakan untuk menjalankan server *apache* dan melakukan pengembangan web berbasis PHP. XAMPP sebenarnya merupakan gabungan dari beberapa perangkat lunak yang berhubungan dengan server, web, dan pengembangannya. Pengertian XAMPP secara akronim berasal dari singkatan masing-masing program, yakni X (*Cross Operating System*), A (*Apache*), M(*MySQL*), P (*PHP*), dan P (*Perl*)[19].

2.9.2 Cascading Style Sheet (CSS)

Cascading Style Sheets (CSS) adalah teknologi yang berfungsi untuk memperindah tampilan website yang sudah digunakan selama bertahun-tahun. Pada sebuah website, selalu terdapat dokumen, gambar, dan teks. Dokumen di sini tidak seperti yang tersimpan di komputer (PDF, Word, dll), melainkan berbentuk HTML dan CSS.

Sementara itu, *style sheets* pada CSS adalah dokumen teks yang dapat diedit dalam banyak program untuk desain dokumen, baik untuk desain cetak maupun online. *Style sheets* dalam situs memiliki tujuan yang sama dengan desain cetak. Namun, diberi tugas tambahan seperti memberi tahu website bagaimana cara menerjemahkan dokumen yang sedang dilihat. CSS dapat memengaruhi seluruh tampilan yang ada di-website[20].

2.9.3 PHPMy Admin

Pengertian PHP MyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui website Jejaring Jagat Jembar (*World Wide Web*). Sedangkan pendapat lain mengatakan bahwa PHP MyAdmin adalah aplikasi web yang dibuat oleh phpmyadmin.net.

Dari beberapa pendapat para ahli diatas dapat disimpulkan bahwa, PHP MyAdmin merupakan aplikasi yang bertujuan untuk mengakses basis data secara mudah dan cepat[21].

