

BAB II

LANDASAN TEORI

II.1. Penelitian Terkait

Sebagai bukti penelitian yang akan dibuat, maka penelitian akan dibandingkan terhadap penelitian sejenis yang pernah dilakukan. Adapun Penelitian sebelumnya yang penulis angkat yaitu :

1. Penelitian dengan judul “Analisa Dan perancangan Web Portal Sistem Informasi Akademik Beberapa Perguruan Tinggi Menggunakan Basis Data Terdistribusi Pada Suatu Yayasan”. Tujuan dari penelitian ini adalah menganalisa dan mendesain web portal sistem informasi akademik pada perguruan tinggi yang dapat diakses kapanpun dimanapun tanpa terkendala waktu yang terintegrasi pada suatu domain tetapi memiliki basis terdistribusi dalam satu yayasan. (Agi Candra Bramantia, dkk, 2017)

2. Penelitian dengan judul “Faktor-Faktor Pengembangan Sistem Informasi Akademik Berbasis Web Pada Perguruan Tinggi Swasta Palembang”. Penelitian ini bertujuan untuk meneliti kontribusi teknologi dan infrastruktur, desain, sumber daya manusia dan budaya dalam pengembangan sistem informasi akademik berbasis web pada perguruan tinggi di Palembang. (Muhammad Soekarno Putra, 2018)

3. Penelitian yang berjudul “ Sistem Informasi Akademik Lembaga Pendidikan Pada Bimbingan Belajar Matrik Palembang Berbasis Web” Bahwa menggunakan

metode SDLC untuk membangun aplikasi menerapkan strategi yang dapat dilaksanakan dalam rangka memenuhi tuntutan dalam pelayanan akademik pada bimbingan belajar matrik dengan menerapkan sistem informasi yang berbasis komputer atau penggunaan sistem komputerisasi sebagai alat pengolah data yang tepat dengan akses dengan ketelitian maupun data tersimpan cepat di banding akses kerja manual. (Endah Saputri, 2016)

4. yang berjudul “Penerapan sistem informasi akademik sekolah dasar negeri 2 Ilung pasar malam berbasis web” Dihasilkan aplikasi sistem ini dapat lebih efisien dari segi biaya, tenaga, dan waktu, Sehingga efektif dalam mencapai tujuan serta dapat mempermudah dalam pengaksesan sistem yang selama ini sering mengalami hambatan dalam melihat informasi, seperti jadwal penerimaan siswa baru serta persyaratan dan tata cara yang harus dipersiapkan. (agus salim, 2019)

5. Penelitian yang berjudul “Rancang Bangun Aplikasi E-COMMERCE Berbasis Web Menggunakan Metode SDLC” Membuat aplikasi keamanan untuk mengatasi masalah keamanan sistem dapat diretas oleh penyusup dapat menimbulkan kerugian, bisa berupa kerugian materi, kerugian kepercayaan konsumen dan kehilangan data. Bahwa metode SDLC diterapkan untuk membuat sebuah rancangan aplikasi *e-commerce* yang mampu mengatasi masalah keamanan di perusahaan supaya berkas tidak akan hilang dan tidak merugikan perusahaan tersebut. (Danang Sudrajat, 2019)

II.2. Uraian Teoritis

II.2.2. Sistem Informasi

Sistem informasi adalah suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan (Agus Mulyanto, 2009:29). Sedangkan definisi lain menyatakan bahwa sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (Al Bahra, 2005:13). Berdasarkan pendapat para ahli diatas pengertian sistem informasi adalah suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

II.2.3 Murid

Murid yaitu manusia yang hidup dalam satu lingkungan sosial yang mikro, yaitu keluarga sangat besar sekali pengaruhnya. Peranan keluarga sebagai pendorong perkembangan daya Penalaran individu dipengaruhi oleh interaksi sosialnya yang dinamis dan wajar, status sosial ekonomi yang berpengaruh. Lingkungan materil yang dihadapi siswa dalam keluarganya itu lebih luas, maka ia dapat kesempatan yang luas pula untuk mengembangkan berbagai kecakapannya. Sudah tentu ada konflik - konflik yang akan berakibat pada perkembangan kejiwaan siswa tersebut yang jelas

kesemuanya ini berpengaruh pada prestasi -prestasi belajar yang dicapainya, tergantung pada bagaimana pola kehidupannya. Maka dari itu peneliti dapat menyimpulkan bahwa definisi dari implementasi sistem Integrasi pendaftaran siswa baru secara online adalah sebuah media yang bertujuan untuk mempermudah siswa yang ingin meneruskan ke jenjang sekolah tingkat menengah untuk melakukan pendaftaran siswa baru tanpa perlu datang ke sekolah karena media ini merupakan media tanpa batas waktu dan tempat (Irva Oktavia: 2015).

II.2.4.Sistem Informasi Akademik (SIA)

Konsep Sistem Hanif Al Fatta menjelaskan bahwa “sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama. Hal ini juga sama dikemukakan oleh Indrajit dalam buku Jeperson Hutapean bahwa sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya. Sistem menurut Lani Sidharta dalam buku Jeperson Hutapean sistem adalah himpunan dari bagian-bagian yang saling berhubungan yang secara bersama mencapai tujuan-tujuan yang sama.

Dari pengertian di atas, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sistem dapat pula diartikan sebagai sekumpulan subsistem, komponen ataupun elemen yang saling bekerja sama

dengan tujuan yang sama untuk menghasilkan output yang sudah di tentukan sebelumnya

II.2.4 SDLC (*System Development Life Cycle*)

SDLC merupakan proses logika yang digunakan oleh seorang analis sistem untuk mengembangkan sebuah sistem informasi yang melibatkan *requirements*, *validation*, *training* dan pemilik sistem. SDLC berisi tahapan-tahapan yang dikembangkan untuk tujuan tertentu. Berikut ini tujuh tahapan yang harus dilewati.

1. Tahapan Analisis Sistem

Tahapan pertama, yaitu analisis sistem. Pada tahap ini, sistem akan dianalisis bagaimana akan dijalankan nantinya. Hasil analisis berupa kelebihan dan kekurangan sistem, fungsi sistem, hingga pembaharuan yang dapat diterapkan. Bagian ini termasuk dalam bagian perencanaan. Bagian lain yang termasuk dalam perencanaan ialah lokasi sumber daya, perencanaan kapasitas, penjadwalan proyek, *estimasi* biaya, dan penetapan. Dengan demikian, hasil dari tahap perencanaan ialah rencana proyek, jadwal, estimasi biaya, dan ketentuan. Idealnya manajer proyek dan pengembang dapat bekerja maksimal pada tahap ini.

2. Tahapan Perancangan Sistem

Setelah persyaratan dipahami, perancang dan pengembang dapat mulai mendesain *software*. Tahapan ini akan menghasilkan *prototype* dan

beberapa *output* lain meliputi dokumen berisi desain, pola, dan komponen yang diperlukan untuk mewujudkan proyek tersebut. Setelah spesifikasi, kemudian dilakukan perancangan sistem sebagai tahapan kelanjutannya. Tahap ini ialah tahap di mana seluruh hasil analisis dan pembahasan tentang spesifikasi sistem diterapkan menjadi rancangan atau cetak biru sebuah sistem. Tahap ini disebut sebagai cetak biru, di mana sistem sudah siap untuk dikembangkan mulai dari implementasi, analisis sistem, hingga tenaga pendukung sistem yang akan dikembangkan.

3. Tahap Pembangunan Sistem

Pengembangan sistem ialah tahap di mana rancangan mulai dikerjakan, dibuat, atau diimplementasikan menjadi sistem yang utuh dan dapat digunakan. Jika diibaratkan bangunan, tahap ini merupakan tahap membangun. Tahap ini memakan waktu cukup lama karena akan muncul kendala-kendala baru yang mungkin dapat menghambat jalannya pengembangan sistem. Pada tahapan ini, perancangan bisa saja berubah karena satu atau banyak hal. Tahap selanjutnya ialah memproduksi perangkat lunak di bawah proses pengembangan. Menurut metodologi yang sudah digunakan, tahap ini dapat dilakukan dengan cepat. *Output* yang dihasilkan pada tahap ini ialah perangkat lunak yang telah berfungsi dan siap diuji.

4. Tahapan Pengujian Sistem

Sesudah sistem selesai dikembangkan, sistem harus melalui pengujian sebelum digunakan atau dikomersialisasikan. Tahap pengujian sistem harus dijalankan untuk mencoba apakah sistem yang dikembangkan dapat bekerja optimal atau tidak. Pada tahap ini, ada beberapa hal yang harus diperhatikan, seperti kemudahan penggunaan sampai pencapaian tujuan dari sistem yang sudah disusun sejak perancangan sistem dilakukan. Jika ada kesalahan, tahap pertama hingga keempat harus diperbarui, diulangi, atau pun dirombak total. Tahap tes SDLC ialah bagian paling penting dalam rangkaian pembuatan sebuah perangkat lunak. Karena sangat tidak mungkin mempublikasikan sebuah *software* tanpa melalui pengujian terlebih dahulu. Beberapa pengujian yang harus dilewati, antara lain kualitas kode, tes fungsional, tes *integrasi*, tes performa, dan tes keamanan. Untuk memastikan pengujian berjalan teratur dan tidak ada bagian yang terlewat, tes dapat dilakukan menggunakan perangkat Continuous Integration seperti Codeship. Dari tahap ini, akan dihasilkan perangkat lunak yang telah dites dan siap untuk disebarkan ke dalam proses produksi.

5. Implementasi

Implementasi dan pemeliharaan merupakan tahap akhir dalam pembuatan SDLC. Di tahap ini sistem sudah dibuat, diuji coba, dan dipastikan dapat bekerja optimal. Setelah tahap pembuatan selesai, dilakukan implementasi dan pemeliharaan oleh pengguna. Pemeliharaan sangat penting untuk memastikan

sistem bekerja dengan optimal setiap saat. Untuk implementasi, langkah yang harus dilakukan adalah sebagai berikut :

- A. Melakukan survei dan penilaian terhadap kelayakan sistem yang sudah dikembangkan.
- B. Menganalisis dan mempelajari sistem yang sudah ada dan sedang berjalan.
- C. Melakukan pemecahan masalah dalam pengembangan sistem.
- D. Menentukan penggunaan *hardware* dan *software* yang tepat.
- E. Merancang dan mengembangkan sistem baru.
- F. Memelihara dan meningkatkan sistem yang baru jika diperlukan.

Fase ini disebut juga sebagai tahap penyebaran. Pada tahap ini, *software* disebarkan setelah melewati proses yang melibatkan beberapa persetujuan manual. Tahap ini dilakukan sebelum menurunkan *software* ke produksi. Proses penyebaran dapat dilakukan menggunakan Application Release Automation (ARA) sebelum masuk ke proses produksi. *Output* yang didapat dari tahap ini ialah perangkat lunak yang siap untuk diproduksi secara massal.

6. Pemeliharaan Sistem

Pemeliharaan sistem yang sudah dibuat sangat penting untuk *referensi* di kemudian hari. Pemeliharaan ialah tahap akhir yang menjadi permulaan fase yang baru yaitu penggunaan. SDLC belum berakhir di tahap ini. *Software* yang

dihasilkan harus terus dipantau untuk memastikan ia berjalan sempurna. Celah dan kerusakan yang ditemukan pada proses produksi harus dilaporkan dan diselesaikan. Jika ditemukan sebelum diproduksi massal, ini akan lebih baik daripada menyelesaikan dengan merombak semuanya dari awal ke akhir.

II.2.5 *Unified Modeling Language (UML)*

UML adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek. UML dapat menyediakan bahasa pemodelan yang mudah dimengerti oleh pengembang dan dapat dikomunikasikan dengan pemakai. Metode perancangan yang digunakan pada penelitian ini adalah metode berorientasi objek menggunakan UML. (Yulianti: 2015)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam *industry* untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem. UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem, karena tergolong bahasa visual yang lebih mudah dan lebih cepat dipahami dibandingkan dengan bahasa pemrograman.

Unified Modelling Language (UML) biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.

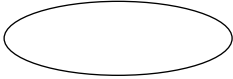
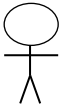
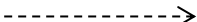
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams* UML.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.

Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan *berbasis visual*. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Yuliati: 2015)

II.2.6.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada tabel II.5 dibawah ini :

Tabel II.5. Simbol *Use Case*

Gambar	Keterangan	Deskripsi
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

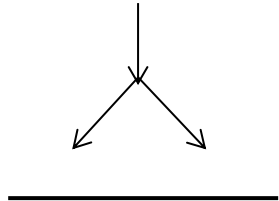
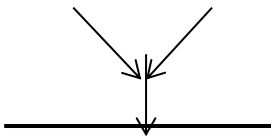
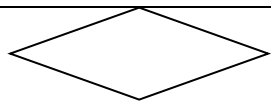
(Sumber :Dinda Atika, 2015)

II.2.6.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Dinda Atika Komariah: 2015).

Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.6 dibawah ini:

Tabel II.6. Simbol Activity Diagram

Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .

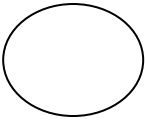
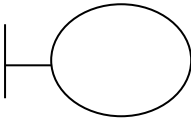
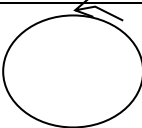
	<i>Swimlane</i>	Pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.
---	-----------------	--

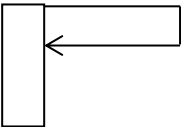
(Sumber :Dinda Atika, 2015

II.2.6.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.7 dibawah ini :

Tabel II.7. Simbol Sequence Diagram

Gambar	Keterangan	Deskripsi
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .

	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber :Dinda Atika, 2015)

II.2.6.4 Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek *eksternal* kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

Tabel II.8. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber :Dinda Atika, 2015)

II.2.7 PHP

PHP merupakan bahasa pemrograman yang berjalan dalam sebuah web server dan berfungsi sebagai pengolah data pada sebuah server. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di server (Yulianti: 2015).

II.2.8 MySql

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat *database* yang bersifat *open source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik windows maupun *linux*. MySQL juga

merupakan sebuah perangkat lunak sistem manajemen basis data SQL yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi user.

MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database* server yang gratis dengan lisensi *GNU General Public License* (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (*Relational Database Management System*). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah database terdapat satu atau beberapa tabel. SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server* (Dinda Atika Komariah :2015).

II.2.9 Database

Database adalah kumpulan data atau informasi yang diperoleh dan selanjutnya disimpan dalam suatu media, umumnya adalah di komputer. Pengolahan database dalam media komputer ditujukan untuk mempermudah dan tentunya mengikuti perkembangan zaman yang semakin menerapkan era komputerisasi. Suatu pengelolaan

sistem database dalam dunia IT biasa dikenal dengan istilah DBMS (*Database Management System*). Suatu database juga dapat didefinisikan terdiri dari kumpulan tabel – tabel yang menyimpan data serta informasi (Yulianti :2015).