

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Penelitian Terdahulu

Selama ini masih banyak sekolah yang belum secara maksimal memanfaatkan teknologi informasi. Sistem penyimpanan data akademik siswa masih banyak yang menggunakan metode konvensional. Akibatnya, pengelolaan penyimpanan data akademik siswa memakan banyak waktu dan tenaga.

Leli Deviana Putri (2014), mengembangkan aplikasi tentang "*Perancangan Sistem Informasi Penerimaan Siswa Baru Pada Sekolah Menengah Pertama Kejuruan Negeri 3 Pacitan*". Dalam penelitiannya disebutkan bahwa dalam pelaksanaan kegiatan penerimaan siswa baru pada saat awal tahun pelajaran sekolah tersebut sering menghadapi masalah karena sistem yang digunakan masih manual. Dengan adanya sistem informasi berbasis komputerisasi dan aplikasi basis data diharapkan dapat membantu dalam penyampaian informasi dan mempermudah dalam pelaksanaan pengolahan data penerimaan siswa baru. Metode yang digunakan dalam penelitiannya adalah wawancara, observasi, studi pustaka, analisis, perancangan sistem, uji coba, dan implementasi.

Ela Saraswati (2013), dalam penelitiannya yang berjudul "*Sistem Informasi Akademik Berbasis WEB Pada Sekolah Menengah (SMP) Negeri 3 Pringkuku*". Dalam penelitiannya ia menyebutkan bahwa selama ini sistem akademik di SMP Negeri 3 Pringkuku masih menggunakan metode konvensional, yakni setiap pengolahan data dicatat dalam buku besar, hal ini sering kali mempersulit dalam proses pencarian data, selain itu metode konvensional ini membutuhkan tempat dan waktu yang lebih banyak. Oleh karena itu, dalam penelitiannya Ela Saraswati membuat sebuah sistem informasi akademik yang bertujuan untuk menghasilkan sistem informasi yang mampu mengatasi kelemahan-kelemahan dari sistem yang ada sebelumnya. Metode pengumpulan data dilakukan dengan mengumpulkan data melalui wawancara dan observasi. Ela Saraswati dalam pembangunan sistem

informasi akademiknya menggunakan bahasa pemrograman PHP dan menggunakan basis data MySQL.

Eko Putra Membara, Liza Yulianti, dan Indra Kanedi (2014), mengembangkan aplikasi tentang “*Sistem Informasi Akademik SMP Negeri 2 Talang Empat Berbasis WEB*”. Dalam penelitian Eko Putra Membara dan kawan-kawan menyebutkan bahwa dengan adanya *website* di SMP Negeri 2 Talang Empat, maka sekolah telah memiliki media alternatif dalam hal penyedia informasi, sehingga hal ini diharapkan dapat mempermudah dalam pencarian data mengenai sekolah. Mereka mengembangkan *website* dengan menggunakan bahasa pemrograman PHP dan *database* MySQL. Metode perancangan sistem yang mereka gunakan yaitu analisis sistem aktual, analisis sistem baru, kemudian perancangan pengujian.

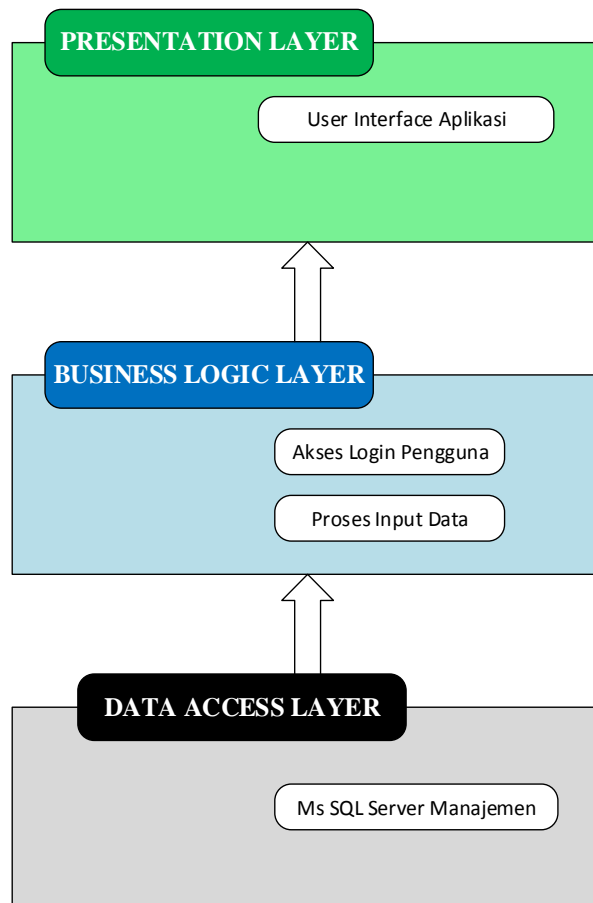
Dari ketiga penelitian sebelumnya, aplikasi yang dibuat memberikan manfaat yang menonjol dalam hal penyimpanan dan penyajian data. Berdasarkan penelitian yang sudah ada, maka penulis akan menambahkan beberapa spesifikasi yang menjadi perbedaan dengan sistem yang dikembangkan. Perbedaan tersebut adalah sebagai berikut:

1. Aplikasi dibangun dalam bentuk aplikasi berbasis *desktop*.
2. Aplikasi dapat mencetak data-data akademik siswa.
3. Aplikasi dapat menyimpan data-data akademik siswa ke dalam bentuk *file* dokumen *word (.docx)*.

2.2 Landasan Teori

2.2.1 Arsitektur Perangkat Lunak

Untuk membangun sebuah sistem, diperlukan rancangan penyusunan sistem. Oleh karena itu, arsitektur perangkat lunak ini bertujuan untuk menggambarkan bagaimana sistem akan dibangun. Adapun arsitektur perangkat lunak dalam penelitian dapat dilihat ada Gambar 2.1.



Gambar 2.1 Arsitektur Perangkat Lunak

2.2.2 Pengertian Aplikasi

Aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer untuk melakukan suatu tugas yang diinginkan pengguna. Umumnya, aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimiliki. Aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. (Wikipedia: 2016)

Menurut Puspita Dwi Astuti (2013), Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan beberapa aktifitas.

2.2.3 Pengertian Sistem

Dikutip dari Wikipedia (2016), sistem adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energy untuk mencapai suatu tujuan.

Berikut penjelasan dari elemen-elemen sistem yang membentuk sebuah sistem:

1. Elemen Tujuan

Setiap sistem memiliki *goal* (tujuan) yang berfungsi sebagai pemotivasi dalam mengarahkan sistem.

2. Elemen Masukan (*input*)

Elemen Masukan sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses.

3. Elemen Proses

Elemen Proses yaitu bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai.

4. Elemen Keluaran (*output*)

Elemen Keluaran merupakan hasil dari elemen proses. Pada sistem informasi, keluaran bisa berupa informasi, maupun cetakan laporan data.

5. Elemen Batas (*boundary*)

Yang disebut batas adalah pemisah antara sistem dan daerah di luar sistem (lingkungan).

6. Elemen Mekanisme Pengendalian dan Umpan Balik (*Control Mechanism*)

Pengendalian ini diwujudkan dengan menggunakan umpan balik (*feedback*), Umpan balik digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya yaitu untuk mengatur sistem agar sistem berjalan sesuai dengan tujuan.

7. Elemen Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan dapat berpengaruh terhadap operasi sistem dalam arti dapat merugikan maupun menguntungkan sistem.

2.2.4 Pengertian Informasi

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan oleh orang untuk menambah fakta-fakta yang ada (Oetomo, 2006:168)

2.2.5 Sistem Informasi

Sistem informasi adalah kombinasi dari teknologi informasi dan aktifitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Istilah sistem informasi sering digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara dimana orang berinteraksi dengan teknologi dalam mendukung proses bisnis (Wikipedia, 2016).

2.2.6 Bahasa pemograman C#

C# (dibaca: *C Sharp*) merupakan sebuah bahasa pemograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari kerangka *.NET Framework* (Wikipedia, 2016). Bahasa pemograman C# ini dikembangkan oleh sebuah tim pengembang di *Microsoft* yang dipimpin oleh Anders Hejlsberg.

2.2.7 Microsoft SQL Server 2014

Microsoft SQL Server merupakan sebuah produk RDBMS (*Relational Database Management System*) dari *Microsoft*. *Microsoft SQL Server* digunakan untuk mengkonfigurasi, dan mengelola komponen yang ada didalam *Microsoft SQL Server*. *Microsoft SQL Server* banyak digunakan pada bidang bisnis, pendidikan, maupun pemerintahan sebagai solusi untuk penyimpanan data.

Kelebihan *Microsoft SQL Server* salah satunya yaitu mudah dalam melakukan koneksi dengan komputer *client* yang pembangunan aplikasinya menggunakan *software* yang sama *platform* dengan *Microsoft SQL*, misalnya *Microsoft Visual Studio*.

Perintah dasar yang terdapat pada SQL (*Structured Query Language*) adalah sebagai berikut:

1. Membuat Database

```
CREATE DATABASE nama_database
```

2. Membuat Tabel

```
CREATE nama_tabel (nama_kolom tipe_data)
```

3. Memasukkan Data

```
INSERT INTO nama_tabel (nama_kolom) values ('isi_kolom')
```

4. Menghapus Data

```
DELETE FROM nama_tabel where condition
```

5. Memilih Data

```
SELECT * FROM nama_tabel
```

6. Mengubah Data

```
UPDATE nama_tabel SET nama_kolom='values'
```

2.2.8 *Unified Modeling Language (UML)*


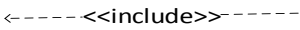


Menurut Dharwiyanti (2003:2), "*Unified Modeling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak". UML adalah metodologi untuk mengembangkan sistem berorientasi objek dan juga sekelompok *tool* untuk mendukung pengembangan sistem.

UML memiliki banyak diagram dengan berbagai fungsinya. Berikut diagram-diagram yang terdapat pada UML:

- a. *Use Case Diagram*, digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use Case* diagram mempresentasikan orang (*actor*) yang akan mengoperasikan atau orang

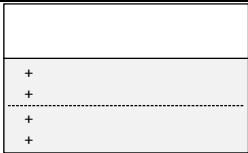

yang berinteraksi dengan sistem aplikasi (Wikipedia, 2016). Simbol-simbol yang digunakan pada *use case* dapat dilihat pada Tabel 2.1:



Tabel 2.1 Daftar Simbol *Use Case* Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

- b. *Class* Diagram, menggambarkan keadaan suatu atribut, struktur dan deskripsi, *class*, *package*, dan objek yang berhubungan satu sama lain dalam suatu sistem. Simbol-simbol yang digunakan pada *Class* Diagram dapat dilihat pada Tabel 2.2:

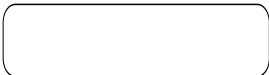


Tabel 2.2 Daftar Simbol *Class* Diagram

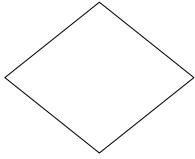

GAMBAR	NAMA	KETERANGAN
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

	<p><i>Composition</i></p>	<p>Bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> dibuat. Misal kelas <i>whole</i> dihapus maka kelas yang menjadi <i>part</i> ikut musnah</p>
	<p><i>Aggregation</i></p>	<p>Hubungan yang menyatakan bahwa suatu kelas menjadi atribut bagi kelas lain.</p>

- c. *Activity Diagram*, berfungsi menggambarkan bentuk visual dari alir kerja yang berisi aktifitas dan tindakan, yang dapat berisi pilihan, pengulangan, dan *concurrency*. Secara garis besar *Activity Diagram* menggambarkan alur kontrol. Simbol-simbol yang digunakan pada *Class Diagram* dapat dilihat pada Tabel 2.3:

Tabel 2.3 Daftar Simbol *Activity Diagram*

GAMBAR	NAMA	KETERANGAN
	<p><i>Action</i></p>	<p>State dari sistem yang mencerminkan eksekusi dari suatu aksi.</p>
	<p><i>Initial Node</i></p>	<p>Bagaimana objek dibentuk atau diawali.</p>
	<p><i>Activity Final Node</i></p>	<p>Bagaimana objek dibentuk dan diakhiri.</p>

	<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
	<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

2.2.9 Data

Data adalah catatan atas kumpulan fakta. Data kemudian diolah sehingga dapat diutarakan secara jelas dan tepat sehingga dapat dimengerti oleh orang lain. Data yang baik adalah data yang dapat dipercaya dan dapat memberikan penjelasan atau gambaran suatu masalah secara keseluruhan (Wikipedia, 2016).

Jenis-jenis data, yaitu:

1. Berdasarkan sifatnya, dibagi menjadi dua yaitu data kualitatif dan data kuantitatif. Data kualitatif merupakan data yang tidak berbentuk angka, misalnya kuesioner. Sedangkan Data kuantitatif adalah data yang berbentuk angka, misalnya harga saham.
2. Berdasarkan sumbernya, dibagi menjadi dua yaitu data internal dan data eksternal. Data internal adalah data dari dalam suatu organisasi yang menggambarkan keadaan organisasi tersebut, misalnya jumlah karyawannya. Data eksternal merupakan data dari luar suatu organisasi yang dapat menggambarkan faktor-faktor yang mungkin mempengaruhi hasil kerja suatu organisasi, misalnya pengaruh daya beli terhadap penjualan.
3. Berdasarkan cara memperolehnya, dibagi menjadi dua yaitu data primer dan data sekunder. Data primer adalah data yang dikumpulkan sendiri oleh perorangan/organisasi secara langsung dari objek yang diteliti dan untuk kepentingan studi dapat berupa interview maupun observasi. Data sekunder yaitu data yang diperoleh atau dikumpulkan oleh studi-studi sebelumnya

yang diterbitkan oleh berbagai instansi, biasanya berupa data dokumentasi dan arsip-arsip resmi.

4. Berdasarkan waktu pengumpulannya, dibagi menjadi dua yaitu data *cross section* dan data berkala. Data *cross section* yaitu data yang dikumpulkan pada waktu tertentu untuk menggambarkan keadaan dan kegiatan pada waktu tersebut, misalnya data penelitian menggunakan kuesioner. Data berkala adalah data yang dikumpulkan dari waktu ke waktu untuk melihat perkembangan suatu kejadian selama periode tersebut, misalnya perkembangan uang beredar.

2.2.10 SDLC (Software Development Life Cycle) Waterfall

Model *Waterfall* merupakan salah satu metode yang ada pada SDLC yang setiap pengerjaannya dilakukan secara berurutan atau secara linear. Jadi setiap tahap harus diselesaikan terlebih dahulu secara maksimal sebelum diteruskan ke tahap berikutnya, hal ini bertujuan untuk menghindari terjadinya pengulangan tahapan.

Berikut ini adalah tahapan-tahapan yang ada pada metode *Waterfall*:

1. Analisis Kebutuhan, seluruh kebutuhan perangkat lunak harus diperoleh pada tahapan ini, termasuk didalamnya kegunaan perangkat lunak yang diharapkan pengguna. Informasi analisis kebutuhan dapat diperoleh melalui wawancara, maupun survei. Dari informasi yang didapatkan tersebut kemudian dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna yang nantinya akan digunakan pada tahap selanjutnya.
2. Desain, sebelum memulai *coding*, tahapan desain bertujuan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk "*blueprint*". Desain harus dapat mengimplementasikan kebutuhan yang telah didapatkan pada tahapan analisis kebutuhan (Pressman, 2007).
3. Implementasi, untuk dapat dimengerti oleh mesin komputer, maka tahapan desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin. Dalam hal ini disebut dengan proses *coding*, secara teknis dikerjakan oleh *programmer*.

4. Pengujian, pada tahapan ini merupakan tahap uji coba pada perangkat lunak yang telah dibuat. Bertujuan untuk memastikan perangkat lunak yang telah dibuat sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya dan bebas dari *error*.
5. Pemeliharaan, perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada tahapan sebelumnya (Wardana, 2013).

2.2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu kegiatan yang dilakukan untuk memperoleh informasi serta mengevaluasi mengenai kualitas dari produk atau layanan yang sedang diuji. Tujuan dari pengujian sebuah program atau aplikasi adalah dapat memenuhi kebutuhan yang mendasari pada perancangan dan pengembangan perangkat lunak tersebut.

Berikut ini adalah tiga metode pengujian perangkat lunak:

1. *Black Box*, pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program. Secara singkat *Black Box* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem (Pendidikan, 2013).
2. *White Box*, pengujian dengan melihat ke dalam aplikasi untuk meneliti kode-kode program yang ada. Jika modul menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan maka baris-baris program, *variable*, parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-*compile* ulang (Johns, 2013).
3. *Grey Box*, merupakan kombinasi dari *black box* dan *white box* dengan mengambil kelebihan dan mengurangi kekurangan dari metode *black box* dan *white box* (Johns, 2013).

