

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang telah dilakukan oleh beberapa peneliti yang bertema “Sistem Informasi Akademik” telah terangkum dalam Tabel 2.1.

Tabel 2.1 Tabel Perbandingan Penelitian

Author	Judul	Metode	Permasalahan	Tujuan	Hasil
Agung Tri Laksono (2019)	“Sistem Informasi Akademik Berbasis Web (Studi Kasus Lembaga Bimbingan Belajar Cendika Cilacap)”	<i>Waterfall</i>	Semakin bertambahnya data akademik dan dapat meningkatkan layanan	Mengembangkan sistem informasi akademik	Mencakup proses pendaftaran siswa, proses penyampaian informasi jadwal bimbingan, proses penyampaian hasil nilai, dan proses pelaporan kepada Pimpinan LBCC[2]
I Pangaribuan, F Subakti (2019)	“Sistem Informasi Akademik Berbasis Web pada SMK (Sekolah	<i>Prototype</i>	Belum menggunakan sistem informasi yang terkomputerisasi sehingga memiliki	Merancang sistem informasi akademik yang mencakup pendaftaran, penjadwalan kelas	Kegiatan akademik di SMK Teknologi Industri Pembangunan Cimahi dapat

Author	Judul	Metode	Permasalahan	Tujuan	Hasil
	Menengah (Kajuruan) Teknologi Industri Pembangunan Cimahi”		kendala-kendala yang dihadapi seperti proses pendaftaran ulang siswa baru, proses penjadwalan, dan proses penilaian	dan penilaian	berjalan lancar dan dapat meminimalisir kendala pada proses pendaftaran, penjadwalan, dan penilaian[3].
Chandra Kesuma, Desiana Nur Kholifah (2019)	“Sistem Informasi Akademik Berbasis Web pada LKP Rejeki Cilacap”	<i>Waterfall</i>	Sistem yang masih manual antara lain : pengolahan nilai masih menggunakan tulisan serta pemberitahuan nilai dan pengumuman akademik peserta didik harus datang ke LKP	Peserta didik yang ingin mendapatkan informasi akademik tidak lagi datang ke LKP dan lebih praktis tanpa meluangkan waktu ke instansi tersebut.	Sistem ini dapat mencapai suatu kegiatan yang efektif dalam menunjang aktifitas LKP yang lebih kondusif dibanding dengan sistem manual[4]

Penelitian yang akan dilakukan memiliki perbedaan dengan penelitian-penelitian sebelumnya. Perbedaan tersebut diantaranya dapat menginformasikan pemberitahuan jatuh tempo pembayaran kepada peserta didik dengan menggunakan *email* dan dapat memuat status pembayaran di halaman *interface*. Perbedaan lain dari sistem ini yaitu penggunaan *framework Laravel* dalam pengembangan sistem tersebut.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan[5]. Akademik adalah suatu bidang yang mempelajari tentang kurikulum atau pembelajaran dalam fungsinya untuk meningkatkan pengetahuan dalam segi pendidikan atau pembelajaran yang dapat dikelola oleh suatu sekolah atau lembaga pendidikan. Sistem informasi akademik adalah perangkat lunak yang digunakan untuk menyajikan informasi dan menata administrasi yang berhubungan dengan kegiatan akademik. Dengan penggunaan perangkat lunak seperti ini diharapkan kegiatan administrasi akademik dapat dikelola dengan baik dan informasi yang diperlukan dapat diperoleh dengan mudah dan cepat[6].

Sistem Informasi adalah suatu sistem dalam organisasi yang memenuhi kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan laporan yang dibutuhkan kepada pihak luar tertentu [7].

Kualitas suatu informasi tergantung dari 3 (tiga) hal, yaitu[8]:

a. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan serta tidak menyimpang.

b. Tepat waktu (*timeline*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah lama tidak akan memiliki nilai lagi karena informasi menjadi landasan dalam pengambilan keputusan.

c. Relevan (*relevance*)

Informasi tersebut mempunyai kaitan dengan masalah yang dihadapi.

Sistem Informasi mencakup dalam kegiatan akademik. Sistem Informasi Akademik adalah sistem yang dibuat untuk melakukan pengolahan data akademik baik melalui *software* maupun *hardware* sehingga proses kegiatan akademik mampu dikelola dengan baik sehingga menjadi informasi yang berguna untuk manajemen pendidikan[9]. Dengan adanya sistem informasi akademik diharapkan dapat melakukan beberapa hal sebagai berikut[10] :

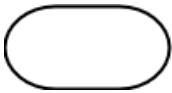

- a. Dapat menyebarkan informasi secara luas, seragam dan cepat.
- b. Dapat menunjang kegiatan belajar mengajar.
- c. Dapat menambah keanekaragaman sumber maupun kesempatan belajar.




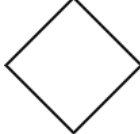
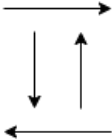

- d. Dapat menambah daya tarik untuk belajar.
- e. Dapat menghemat biaya.




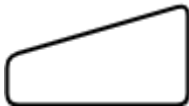
2.2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah suatu disiplin ilmu yang membahas semua aspek dari perangkat lunak, mulai dari tahap awal yaitu *communication*, *requirements capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), desain, *coding*, *testing* sampai *maintenance* (pemeliharaan sistem) setelah digunakan[11]. Analisa kebutuhan pengguna yang terdapat dalam rekayasa perangkat lunak mencakup *flowchart*. *Flowchart* merupakan diagram yang memberikan urutan tindakan dalam menyelesaikan suatu masalah[12]. Berikut Tabel 2.2 yaitu simbol-simbol pada *flowchart*:

Tabel 2.2 Simbol- Simbol *Flowchart*[12]

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Awal atau akhir dari aliran proses
2.		<i>Process</i>	Menggambarkan sebuah proses

No.	Simbol	Nama	Keterangan
3.		<i>Input-output</i>	Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya
4.		<i>Predefined Process (sub program)</i>	Permulaan/proses menjalankan program
5.		<i>Preparation</i>	Proses inialisasi
6.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada
7.		<i>Flow</i>	Menggambarkan alur kerja
8.		<i>Document</i>	Menggambarkan output dalam

No.	Simbol	Nama	Keterangan
			bentuk dokumen
9.		<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> dalam satu halaman
10.		<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang berbeda
11.		<i>Manual Operation</i>	Menggambarkan masukan secara manual tanpa bantuan komputer/pc
12.		<i>Manual Input</i>	Menggambarkan input yang dimasukkan secara manual dari <i>keyboard</i>


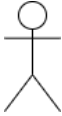

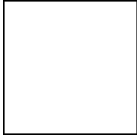
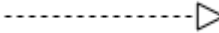
Tahap *coding* dalam rekayasa perangkat lunak diimplementasikan dengan kode program. Konsep pemograman berupa Pemograman Berorientasi Objek. Pemograman berorientasi objek (PBO) merupakan sebuah konsep pemrograman yang menggambarkan suatu proses penyelesaian masalah pada program yang menggunakan *object* dan *class*[13]. *Object* merupakan sekumpulan data berupa *field* atau *variabel*. *Class* merupakan penjelasan dari objek. Contoh dari perangkat pemodelan berbasis objek salah satunya adalah UML.


UML (*Unified Modeling Language*) merupakan metode pemodelan visual yang dipakai dalam perancangan dan pembuatan *software* yang berorientasikan pada objek[14]. Ada beberapa diagram dalam UML yang sering digunakan dalam pengembangan sebuah sistem, yaitu:

a. *Use Case Diagram*

Use case diagram merupakan gambaran mengenai interaksi antara aktor dan sistem. Aktor dalam use case diagram merupakan gambaran entitas dari manusia. Berikut Tabel 2.3 yaitu simbol-simbol yang digunakan dalam *use case diagram*:

Tabel 2.3 Simbol - Simbol *Use Case Diagram*[14]

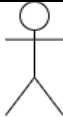
No.	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Mengidentifikasi scenario dalam use case
2.		<i>Actor</i>	Mengidentifikasi aktor yang berperan dalam use case
3.		<i>Association</i>	Penghubung antara use case dan aktor
4.		<i>System</i>	Pembatas sistem yang terhubung dengan aktor yang menggunakannya
5.		<i>Include</i>	Mengarahkan bahwa use case sumber secara tersirat
6.		<i>Extend</i>	Menunjukkan bahwa use case target


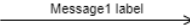
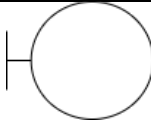

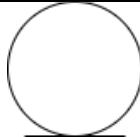
No.	Simbol	Nama	Keterangan
			memperluas perilaku dari use case sumber pada suatu titik yang diberikan

b. Sequence Diagram

Sequence diagram merupakan salah satu diagram UML yang mendiskripsikan hubungan antar objek sesuai urutan waktu. Dalam *sequence diagram* harus mengetahui semua objek yang terlibat dalam sebuah *use case* untuk menjelaskan tahapan yang akan dicapai dalam penggunaan sistem. Berikut Tabel 2.4 yang menunjukkan simbol-simbol pada *sequence diagram* :

Tabel 2.4 Simbol - Simbol *Sequence Diagram*[14]

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menggambarkan pengguna

No.	Simbol	Nama	Keterangan
2.		<i>Life Line</i>	Objek entity, antarmuka yang saling berinteraksi
3.		<i>Message</i>	Menggambarkan pesan atau hubungan antar objek sesuai urutan kejadian
4.		<i>Boundary</i>	Menggambarkan sebuah form
5.		<i>Control Class</i>	Penghubung boundary dengan tabel
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang dilakukan

Pengembangan rekayasa perangkat lunak meliputi rekayasa web didalamnya. Rekayasa web adalah sebuah aplikasi yang

menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi, dan pemeliharaan aplikasi berbasis Web (*Web-based applications*)[15].

2.2.3 Hypertext Preprocessor (PHP)

Hypertext preprocessor atau lebih dikenal dengan sebutan PHP merupakan bahasa pemrograman *script server-side* yang digunakan untuk pengembangan web. PHP disebut bahasa pemrograman *server-side* karena diproses pada komputer *server*. Bahasa pemrograman ini dibuat untuk para pengembang web agar dapat membangun suatu halaman web yang bersifat dinamis. Hal ini berbeda dengan bahasa pemrograman *client-side* seperti *Javascript* yang diproses didalam *web browser*[16].

2.2.4 Laravel

Laravel merupakan salah satu *framework* PHP yang dikembangkan oleh *Taylor Otwell*. *Framework* ini dirilis dibawah MIT dan dirancang dengan konsep MVC (*model view controller*). *Laravel* merupakan pengembangan *website* berbasis MVP yang ditulis dalam PHP yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan[17].

MVC (*model view controller*) merupakan sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika

dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi[18].

- a. *Model* mewakili struktur data. Biasanya *model* berisikan fungsi-fungsi yang membantu pengembang dalam pengelolaan basis data, seperti memasukkan data ke basis data, pembaruan data.
- b. *View* merupakan bagian yang mengatur tampilan ke pengguna.
- c. *Controller* merupakan bagian yang menghubungkan antara *model* dan *view*.

2.2.5 Basis Data

Basis data merupakan sekumpulan data yang terbentuk dari berkas atau file yang saling berkaitan satu sama lain dengan tata cara tertentu untuk membentuk data baru atau informasi. Dalam basis data diperlukan sebuah perangkat lunak yaitu sistem manajemen basis data atau database management system (DBMS)[19]. Salah satu perangkat lunak DBMS adalah SQL (*Structured Query Language*). Secara umum SQL mempunyai dua bahasa yaitu[20] :

- a. DDL (Data Definition Language)

Data definition language merupakan kumpulan perintah SQL yang digunakan untuk membuat, mengubah, dan menghapus struktur dan definisi metadata dari objek-objek database. DDL mempunyai beberapa skema antara lain :

- a) *Create* : membuat data baru berupa database atau tabel.
- b) *Drop* : menghilangkan suatu objek di database.
- c) *Alter* : mengubah struktur tabel.

b. DML (*Data Manipulation Language*)

Data manipulation language merupakan bahasa yang memungkinkan pengguna untuk mengakses atau memanipulasi data dalam sistem basis data yang bersifat relational. DML mempunyai beberapa skema antara lain :


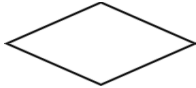
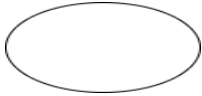
- a) *Insert* : menambahkan data satu baris atau lebih pada tabel.
- b) *Update* : mengubah data satu baris atau lebih pada tabel.
- c) *Select* : menampilkan data yang terdapat pada tabel baik secara keseluruhan atau berdasarkan pilihan data.
- d) *Delete* : menghapus data pada setiap baris ataupun secara keseluruhan.


Gabungan antara DBMS dengan basis data akan menghasilkan suatu sistem yang disebut sistem basis data. Sistem basis data bertujuan untuk menyederhanakan hubungan antara pengguna dengan sistemnya.

Pemodelan awal basis data yang paling banyak dipergunakan ialah menggunakan ERD (*Entity Relationship*

Diagram). ERD merupakan alat yang dipergunakan untuk memodelkan struktur data dengan menggambarkan entitas dan hubungan antara entitas (*relationship*) secara abstrak (konseptual)[21]. ERD berisikan simbol-simbol yang sesuai dengan hubungan antara masing-masing entitas. Berikut Tabel 2.5 mengenai simbol-simbil ERD :

Tabel 2.5 Simbol - Simbol Entity Relationship Diagram[21]

No.	Simbol	Nama	Keterangan
1.		Entitas	Kumpulan dari objek yang dapat diidentifikasi secara unik
2.		Relasi	Hubungan yang terjadi antara salah satu atau lebih entitas. Jenis hubungan antara lain one to one, one to many, dan many to many
3.		Atribut	Karakteristik dari entitas atau relasi yang merupakan penjelasan detail mengenai entitas

No.	Simbol	Nama	Keterangan
4.		Garis atau alur	Penghubung antara entitas dengan atributnya dan humpunan entitas dengan himpunan relasinya

2.2.6 Black Box

Pengujian *black box* merupakan sebuah metode yang dipakai untuk menguji sebuah perangkat lunak yang berfokus pada spesifikasi fungsional. Hasil untung menghitung berapa persentase keberhasilan dari pengujian *black box* dapat dihitung dengan rumus :

$$\text{Hasil} = \frac{\text{proses berhasil}}{\text{proses seluruhnya}} \times 100\%$$

Pada pengujian *black box* hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Keuntungan menggunakan pengujian *black box* sebagai berikut[22] :

- a. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
- b. Pengujian dilakukan dari sudut pandang pengguna.

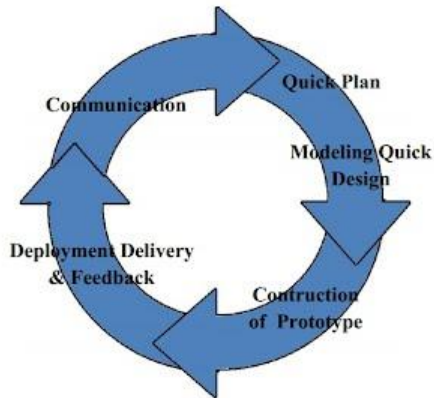
- c. Pengembang dan pengguna saling bergantung satu sama lain.

2.2.7 Metode *Prototype*

Metode *prototype* merupakan salah satu model perangkat pengembangan perangkat lunak. *Prototyping* adalah proses pembuatan model sederhana *software* yang mengizinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal[23]. Berikut tahapan-tahapan dalam model *prototype*[24]:

- a. *Communication* atau komunikasi dan pengumpulan data awal, yaitu analisis terhadap kebutuhan pengguna.
- b. *Quick plan*, yaitu tahapan perencanaan kebutuhan.
- c. *Modelling Quick Design*, tahapan pembuatan desain.
- d. *Contruction of prototype*, yaitu pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan.
- e. *Deployment Delivery & Feedback*, yaitu mengevaluasi *prototype* dan merinci analisis terhadap kebutuhan pengguna. Perbaiki *prototype*, yaitu pembuatan tipe yang sebenarnya berdasarkan hasil evaluasi *prototype* dan selanjutnya produksi akhir, yaitu memproduksi

sistem secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.1 Metode *Prototype*[24]

2.2.8 Skala *Likert*

Skala *Likert* merupakan skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang ataupun kelompok pada sebuah peristiwa ataupun fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Biasanya tersedia lima pilihan skala, misalnya dengan format seperti[25]:

- a. Sangat tidak (setuju/baik/sependapat)
- b. Tidak (setuju/baik/sependapat)
- c. Cukup/netral
- d. Setuju (baik/sependapat)

e. Sangat (setuju/baik/sependapat)