

## BAB II

### LANDASAN TEORI

#### 2.1 Ayat Al-Qur'an Tentang Pendidikan.

Al-Qur'an merupakan firman Allah SWT yang mengandung berbagai aspek kehidupan baik aspek hukum, sejarah, ilmu, akidah (keimanan) maupun isyarat tentang pengetahuan. Semua itu dipermudahkan bagi manusia agar dijadikan pedoman hidup sehingga kehidupannya lebih baik dan mendapatkan rahmat dari Allah SWT. Berikut ini adalah ayat Al-Qur'an yang berisi isyarat tentang pengetahuan dan ilmu dari Allah SWT, yakni Surah Ali Imran Ayat 18 dan surah Al-Kahfi Ayat 66.

Di dalam Al-Qur'an terdapat banyak ayat yang menjelaskan tentang pendidikan yaitu sebagai berikut:

شَهِدَ اللَّهُ أَنَّهُ لَا إِلَهَ إِلَّا هُوَ وَالْمَلَائِكَةُ وَأُولُو الْعِلْمِ قَائِمًا بِالْقِسْطِ لَا إِلَهَ إِلَّا هُوَ الْعَزِيزُ الْحَكِيمُ ١٨

Artinya : *“Allah menyatakan bahwasanya tidak ada Tuhan melainkan Dia (yang berhak disembah), Yang menegakkan keadilan. Para Malaikat dan orang-orang yang berilmu (juga menyatakan yang demikian itu). Tak ada Tuhan melainkan Dia (yang berhak disembah), Yang Maha Perkasa lagi Maha Bijaksana”*.

(QS Ali Imran : 18).

Selanjutnya dalam Al-Qur'an Surah Al-Kahfi ayat 66 Allah berfirman:

قَالَ لَهُ مُوسَىٰ هَلْ أَتَّبِعُكَ عَلَىٰ أَنْ تُعَلِّمَ مِنِّي مِمَّا عَلَّمْتَ رُشْدًا ۖ ٦٦

Artinya : “Musa berkata kepada Khidhr: "Bolehkah aku mengikutimu supaya kamu mengajarkan kepadaku ilmu yang benar di antara ilmu-ilmu yang telah diajarkan kepadamu?" (QS Al-Kahfi : 66).

## **2.2 Teori Yang Berhubungan dengan Sistem Secara Umum**

### **2.2.1 Sistem**

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama (Pratama, 2014:7). Sedangkan menurut (Hutahaean, 2012:2), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berkaitan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu.

Dengan demikian jelas bahwa pengertian ini mempunyai peran yang penting untuk mempelajari suatu sistem. Pengertian sistem yang telah dijelaskan dapat dilihat bahwa pendapat-pendapat tersebut ada dasarnya mempunyai satu pengertian yang sama pada akhirnya adalah pencapaian suatu sasaran atau tujuan tertentu secara bersama-sama.

### **2.2.2 Informasi**

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data, data kenyataan menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah kejadian terjadi pada saat tertentu (Hutahaean, 2012:9). Sedangkan dilain sisi informasi merupakan hasil pengolahan data dari satu atau berbagai

sumber, yang kemudian diolah, sehingga memberikan nilai, arti, dan manfaat (Pratama, 2014:9).

Dari definisi-definisi tersebut, disimpulkan bahwa informasi merupakan hasil dari proses pengolahan data mentah yang berupa sesuatu yang lebih berguna dan digunakan untuk membantu dalam pengambilan keputusan.

### **2.2.3 Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategis dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang di perlukan. (Sutabri, 2012: 46)

Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan sumber daya manusia (SDM) yang terlatih. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengolah perencanaan, kontrol, koordinasi dan pengambilan keputusan. (Iputu, 2014: 10)

Berdasarkan kutipan yang penulis ambil mengenai definisi sistem informasi dapat disimpulkan bahwa komponen yang saling berkaitan dan bekerjasama dalam mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi yang bermanfaat untuk penggunaanya dengan tujuan mengambil keputusan

#### **2.2.4 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 (Ilhamsah, 2012). Sedangkan menurut (Risetyawan, 2010:6), Akademik adalah suatu bidang yang mempelajari tentang kurikulum dalam fungsinya untuk meningkatkan pengetahuan dalam segi pendidikan yang dapat dikelola oleh suatu sekolah.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa Akademik adalah suatu bidang untuk meningkatkan pengetahuan dalam sebuah pendidikan disekolah maupun instansi.

#### **2.2.5 Sistem Informasi Akademik**

Sistem Informasi Akademik (SIKAD) merupakan sistem informasi yang secara khusus dirancang untuk memenuhi kebutuhan akademik dalam sebuah instansi pendidikan, dimana sistem tersebut kaya akan data akademik dan hanya dapat digunakan oleh pihak yang memiliki hak akses ke dalam sistem informasi tersebut (Noviandi, 2012). Sedangkan Sistem Informasi Akademik adalah segala macam hasil interaksi antara elemen di lingkungan akademik untuk menghasilkan informasi yang kemudian dijadikan landasan pengambilan keputusan, melaksanakan tindakan, baik oleh pelaku proses itu sendiri maupun dari pihak luar. (Agustin, 2012).

Berdasarkan kutipan yang penulis ambil dapat disimpulkan mengenai sistem informasi akademik adalah suatu perangkat lunak yang dikembangkan

sesuai dengan kebutuhan akademis yang diinginkan baik itu sekolah maupun instansi lainnya.

### **2.2.6 Kurikulum 2013**

Kemendikbud (2013: 210) menyatakan bahwa Kurikulum 2013 menekankan pada dimensi pedagogik modern dalam pembelajaran, yaitu menggunakan pendekatan ilmiah (*scientific approach*) dalam pembelajaran semua mata pelajaran (tematik terpadu), dan proses mendapatkan dan mengumpulkan informasi dilakukan dengan penilaian otentik. Sedangkan menurut Mulyasa (2013: 163) bahwa Kurikulum 2013 menggunakan pendekatan yang bersifat ilmiah, karena berangkat, berfokus dan bermuara pada hakekat peserta didik untuk mengembangkan berbagai kompetensi sesuai dengan potensinya masing - masing. Dalam hal ini siswa merupakan subjek belajar, dan proses belajar berlangsung secara alamiah dalam bentuk bekerja berlangsung secara alamiah dalam bentuk bekerja dan mengalami berdasarkan kompetensi tertentu, bukan transfer pengetahuan (*transfer of knowledge*).

Berdasarkan pendapat diatas, dapat disimpulkan bahwa Kurikulum 2013 adalah suatu rencana yang disusun untuk melancarkan proses belajar - mengajar berbasis karakter dan kompetensi dengan karakteristik pembelajaran menerapkan pendekatan ilmiah (*scientific approach*), pembelajaran bersifat tematik terpadu, dan penilaian otentik.


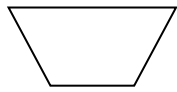
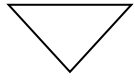
## 2.3 Teori Yang Berhubungan Dengan Teknik Analisa Yang Digunakan.


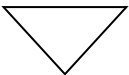
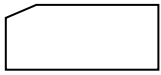
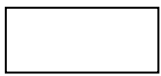
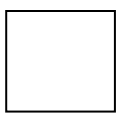




### 2.3.1 *Flowchart*.



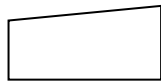
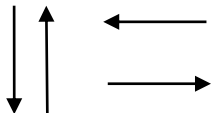
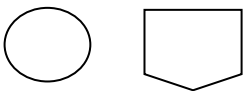
*Flowchart* merupakan gambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program (Pressman, 2006). Sedangkan menurut (Jogiyanto, 2005:795) bagan alir (*Flowchart*) adalah bagan (*Chart*) yang menunjukkan alir (*Flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi serta pada waktu akan menggambarkan suatu bagan alir

Bagan alir sistem (*System Flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada didalam sistem, bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Bagan alir sistem digambar dengan menggunakan simbol-simbol yang tampak seperti tabel 2.1 berikut ini (Jogiyanto, 2005:796).

**Tabel 2.1** simbol-simbol *flowchart*.

No	Simbol	Nama	Keterangan
1		Dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik dan komputer.
2		Kegiatan manual	Menunjukkan pekerjaan manual
3		Simpanan <i>Offline</i>	<i>File</i> non-komputer yang diarsipurut angka ( <i>numercial</i> )


4		Simpanan <i>Offline</i>	<i>File</i> non-komputer yang diarsip urut huruf ( <i>alphabetical</i> )
5		Simpanan <i>Offline</i>	<i>File</i> non-komputer yang diarsip urut tanggal ( <i>cronological</i> )
6		Kartu plong	Menunjukkan <i>input/output</i> yang menggunakan kartu plong
7		Proses	Menunjukkan kegiatan proses dari operasi program komputer
8		Operasi luar	Menunjukkan operasi yang dilakukan di luar proses operasi komputer.
9		Pengurutan <i>Offline</i>	Menunjukkan proses pengurutan data di luar proses operasi komputer
10		Pita magnetik	Menunjukkan <i>input/ouput</i> yang menggunakan pita magnetik.
11		<i>Har disk</i>	Menunjukkan <i>input/ouput</i> yang menggunakan pita <i>Har disk</i> .
12		<i>Discette</i>	Menunjukkan <i>input/ouput</i> yang menggunakan pita <i>discette</i> .

13		<i>Drum magnetik</i>	Menunjukkan <i>input/ouput</i> yang menggunakan pita <i>Drum magnetik</i> .
14		Pita kertas berlubang	Menunjukkan <i>input/ouput</i> yang menggunakan pita kertas berlubang.
15		<i>Keyboard</i>	Menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i> .
16		Garis alir	Menunjukkan arus dari proses.
17		Penghubung	Menunjukkan penghubung ke halaman yang masih sama atau kehalaman lain.





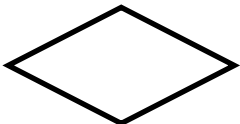

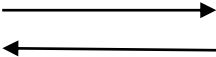
Sumber : Jogyanto (2005:796-799).

Berdasarkan kutipan yang penulis ambil dapat disimpulkan mengenai pengertian *Flowchart* adalah suatu aliran yang menggambarkan data yang akan dibuat dalam suatu program atau aplikasi.

**Tabel 2.2** Simbol-simbol *Flowchart*

No	Simbol	Keterangan
1.		Simbol titik terminal yang digunakan untuk awal dan akhir dari suatu proses



2.		Simbol input/output yang digunakan untuk mewakili data input/output
3.		Simbol proses digunakan untuk menunjukkan pengeluaran yang dilakukan oleh komputer
4.		Simbol penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang masih sama
5.		Simbol untuk database yang digunakan dalam program
6.		Simbol keputusan yang digunakan untuk suatu penyelesaian kondisi didalam program
7.		Dokumen merupakan simbol untuk data yang berbentuk kertas maupun informasi
8.		Simbol aliran data



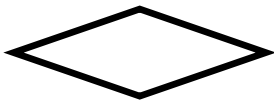
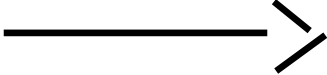
Sumber: (Pressman, 2006)

### 2.3.2 ERD (*Entity Relationship Diagram*)

*Entity Relationship Diagram* (ERD) adalah bentuk paling awal dalam melakukan perancangan basis data relasional. (Rosa & Shalahuddin (2016: 50).

Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen dapat dilihat pada tabel. 2.3;

**Tabel. 2.3** Simbol-simbol ERD

Simbol	Deskripsi
<p>Entitas / <i>Entity</i></p> 	<p>Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama table.</p>
<p>Atribut</p> 	<p>Field atau kolom data yang butuh disimpan kedalam suatu entitas</p>
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja</p>
<p>Asosiasi / <i>Associaation</i></p> <p>N</p> 	<p>Penghubung antar relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakai</p>

Sumber: (Rosa & Shalahuddin, 2016: 50-51)

Menurut Widodo dan Saftian (2005), Ada tiga tipe Relationship yang dikenal, yaitu:

1. *One to one* (1:1)

Pada bentuk relasi ini, suatu *entity* hanya berelasi dengan satu *entity* lainnya, misalnya 1 Direktur memimpin 1 Kantor

2. *One to Many* (1:M)

Pada relasi ini suatu *entity* bisa mempunyai lebih dari satu relasi pada *entity* lainnya, misal seorang dosen mengajar lebih dari satu mata kuliah

3. *Many to Many* (M:M)

Pada relasi ini banyak *entity* yang bisa berelasi dengan banyak *entity* lainnya, sebagai contoh banyak pembeli bisa membeli banyak barang disuatu pasar.

### 2.3.3 UML (*Unified Modeling Language*).

UML adalah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman objek, yaitu *Unified Modeling Language*. *Unified Modeling Language* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. *Unified Modeling Language* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *Unified Modeling Language* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *Unified Modeling Language* tidak terbatas pada

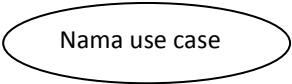
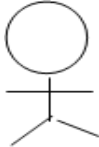


metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa, 2014:137-138).

### **2.3.3.1 Use case Diagram.**

*Use case diagram* menyajikan interaksi antara *case* dan aktor dalam sistem yang akan dikembangkan. *Use case* sendiri adalah fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut menurut pandangan pemakai sistem. Sedangkan aktor bisa berupa orang, peralatan, atau sistem lain yang berinteraksi terhadap sistem yang akan dibangun. *Use case* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi (Tohari, 2014:47).

Menurut (Rosa, 2014:155), *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 aktor dengan sistem informasi yang akan dibuat, *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 aktor dan *use case*. Elemen-elemen yang digunakan dalam pemodelan *Use Case* sistem ditunjukkan pada tabel. 2.4 berikut:

**Tabel 2.4** Tabel simbol *Use Case diagram*

No	Simbol	Keterangan
1	<p><i>Use case</i></p>  <p>Nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	<p>Aktor / <i>actor</i></p>  <p>Nama aktor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.
4	<p>Ekstensi / <i>extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> 	Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu: mirip

		dengan prinsip <i>inheritence</i> pada pemograman berorientasi objek; biasa <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan
--	--	--

Sumber : Rosa (2016:156).






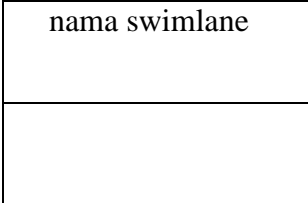
### 2.3.3.2 Activity Diagram.

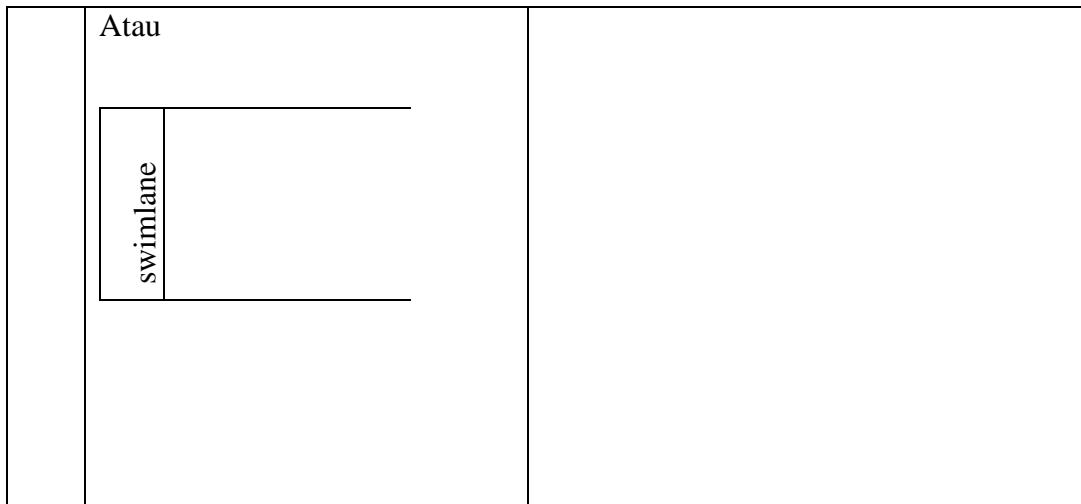
*Activity Diagram* memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses (Tohari, 2014:114). *Activity Diagram* menggambarkan aliran fungsionalitas sistem. Ada dua kegunaan diagram aktivitas dalam pemodelan dengan UML. Dua kegunaan tersebut yaitu sebagai berikut:

- a. Pada tahap pemodelan bisnis, *activity diagram* dapat digunakan untuk menunjukkan alur kerja bisnis (*businnes workflow*).
- b. Pada tahap pemodelan sistem, *activity diagram* dapat dijelaskan untuk menjelaskan aktivitas yang terjadi di dalam sebuah *use case*.

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2016:161). Berikut adalah simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel. 2.5 berikut :

Tabel. 2.5 Simbol-simbol *Activity diagram*

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3	Percabangan ( <i>decision</i> ) 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4	Penggabungan ( <i>join</i> ) 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi



(Sumber : Rosa, 2016:162-163)

### 2.3.3.3 Class Diagram.

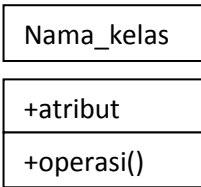



*Class diagram* digunakan untuk menampilkan kelas-kelas atau paket-paket dalam sistem dan relasi antar mereka. Biasanya, dibuat beberapa diagram kelas untuk sistem. Satu class diagram menampilkan subset dari kelas-kelas dan relasinya. Diagram kelas lainnya, mungkin menampilkan kelas-kelas termasuk atribut dan operasi dari kelas-kelas pembentuk diagram. Sedangkan *class diagram* yang lainnya lagi, mungkin menampilkan paket-paket kelas dan relasi antar paket-paket. *Class diagram* adalah alat perancangan terbaik untuk tim pengembang perangkat lunak. *Class diagram* membantu tim pengembang mendapatkan pola kelas-kelas dalam sistem, struktur sistem sebelum menuliskan kode program, dan membantu untuk memastikan bahwa sistem adalah rancangan terbaik dari beberapa alternatif rancangan (Sholiq, 2010:149-150).




Diagram kelas atau *class diagram* menggunakan striktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan



variabel-variabel yang dimiliki oleh suatu kelas. Kemudian operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa dan Shalahuddin, 2016:141-142). Berikut adalah simbol-simbol yang ada pada *class diagram* dapat dilihat pada tabel. 2.6 berikut:

**Tabel 2.6** Simbol-simbol *class diagram*

No	Simbol	Deskripsi
1	<p>Kelas</p> 	Kelas pada struktur sistem
2	<p>antarmuka atau <i>interface</i></p>  <p><b>nama_interface</b></p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	<p>asosiasi atau <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4	<p>asosiasi berarah (<i>directed association</i>)</p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5	<p>Generalisasi</p>	Relasi antarkelas dengan makna generalisasi

		
6	kebergantungan atau <i>dependency</i> 	Kebergantungan antarkelas
7	Agregasi atau <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

(Sumber : Rosa, 2016:146-147)

## 2.4 Teori Yang Berhubungan Dengan Alat Bantu Pembuatan Sistem.

### 2.4.1 DBMS (*Database Management System*).

Rosa dan Shalahuddin (2015:44) *Database Management System* (DBMS) adalah perangkat lunak komputer khusus yang disediakan dari vendor-vendor komputer yang digunakan untuk membuat, mengakses, mengontrol, dan mengelola *database*. Lain lagi menurut Raharjo (2015:12) *Database Management System* adalah *Software* yang berguna untuk membuat dan mengelola *database*.

Dari uraian diatas dapat disimpulkan bahwa *Database Management System* adalah *Software* yang digunakan untuk membuat, mengakses, mengontrol, dan mengelola *database*.

### 2.4.2 Database.

Menurut Sutarman (2012:15) *Database* adalah sekumpulan file yang saling berhubungan dan terorganisasi atau sekumpulan *record-record* yang menyimpan data dan berhubungan di antaranya. Lain lagi menurut Rosa dan Shalahuddin (2015:43) sistem basis data adalah sistem terkomputerisasi yang bertujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Dari uraian diatas dapat disimpulkan bahwa data adalah bahan mentah yang diolah untuk menghasilkan suatu informasi.

### 2.4.3 MySQL.

Menurut (Hidayatullah, 2015:231) MySQL adalah salah satu aplikasi *Database Management System* (DBMS) yang sudah sangat banyak digunakan oleh para pemrograman aplikasi web. Adapun yang dimaksud dengan DBMS adalah aplikasi yang dipakai untuk mengelola basis data. Contoh DBMS lainnya adalah: PostgreSQL, SQL Server, MS Access, DB2, Dbase, Foxpro, dsb. MySQL adalah sistem manajemen database SQL yang bersifat Open Source dan paling populer saat ini. Sistem database MySQL mendukung beberapa fitur seperti *multithreaded*, *multiuser*, dan *SQL database Manajemen Sistem* (DBMS). Sedangkan menurut (Raharjo, 2015 : 212) *MySQL* merupakan sistem database yang banyak digunakan untuk pengembangan aplikasi web.

Berdasarkan kutipan yang penulis ambil dapat disimpulkan mengenai pengertian dari MySQL adalah suatu alat atau aplikasi yang dibuat untuk

mengelola data-data yang penting ke dalam sistem dengan keamanan yang sangat kuat.

#### **2.4.4 XAMPP.**

*Xampp* adalah kompilasi *software* yang membungkus *apache HTTP server*, *MySQL*, *PHP*, dan *Perl* (Raharjo, 2015:25). Lain lagi menurut Nugroho (2008:2) *Xampp* adalah program yang berisi paket *PHP* berbasis *open source* yang dikembangkan oleh sebuah komunitas *open source*.

Dari uraian tersebut dapat disimpulkan bahwa *Xampp* adalah kompilasi *software* yang berisi paket *apache HTTP server*, *MySQL*, dan *PhpMyadmin*.

#### **2.4.5 PHP**

Menurut Hidayatullah (2015:231) *Hypertext Processor* atau sering disebut *PHP* adalah bahasa scripting khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan *PHP* harus menggunakan *webserver*. *PHP* juga dapat diintegrasikan dengan *HTML*, *Javascript*, *Jquery*, dan *Ajax*. Namun, pada umumnya *PHP* lebih banyak digunakan bersama dengan file bertipe *HTML* dengan menggunakan *PHP* kita bisa membuat *website powerfull* yang dinamis dengan disertai manajemen database-nya.

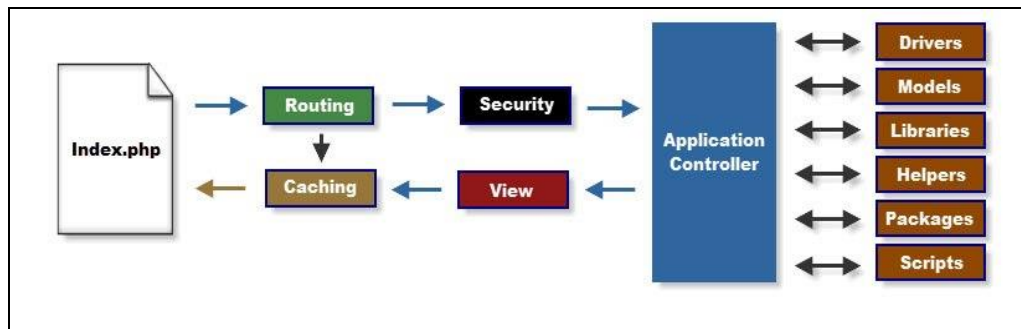
Sedangkan pendapat dari Anhar (2010:3) *PHP* merupakan *script* yang terintegrasi dengan *HTML* dan berada pada *server (server side HTML embedded scripting)*. *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis.

Berdasarkan kutipan yang penulis ambil dapat disimpulkan mengenai pengertian PHP adalah suatu alat bantu dalam pengkodean sebuah aplikasi yang mana untuk mengetahui letak syntax error dalam suatu sistem atau aplikasi.

#### 2.4.6 *CodeIgniter*.

*Codeigniter* adalah *framework* PHP yang memakai sistem Model *View Controller* (MVC). Jadi secara garis besar, disana ada 3 komponen utama yaitu, Model, *View* dan *Controller* (Basuki, 2016:12). Menurut Basuki (2014:12), *codeigniter* merupakan salah satu dari sekian banyak *framework* PHP, kini mulai banyak digunakan dalam mengembangkan aplikasi berbasis web. Saat menjalankan aplikasi berbasis *codeigniter*, ada aliran data dan proses yang dijalankan *Codeigniter*.

Berikut aliran data *codeigniter* yang akan dijelaskan pada Gambar 2.1 sebagai berikut:



(Sumber : Basuki, 2014:16)

**Gambar 2.1.** Aliran Data *Codeigniter*

Keterangan gambar:

- a. *index.php* berfungsi sebagai *front controller* yang bertugas untuk menginisialisasi sumber daya dasar untuk menjalankan *CodeIgniter*.

- b. *Router* memeriksa *HTTP request* yang datang dari user untuk menentukan apa yang harus dilakukan dengan request tersebut.
- c. Jika file *chace* ada, maka akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
- d. *Security* sebelum *controller* dimuat, *HTTP request* dan data yang dikirimkan *user* akan difilter untuk keamanan.
- e. *Controller* memanggil model, *core libraries*, *helpers*, dan sumber daya lainnya yang diperlukan untuk memproses data berdasarkan *HTTP request*.
- f. Akhirnya *view* yang dihasilkan akan dikirim ke *browser*. Jika *cache* diaktifkan, maka *view* akan disimpan sebagai *cache* dahulu, sehingga pada *request* berikutnya langsung dapat ditampilkan.

Menurut Supono dan Viridiandry Putratama (2016:112), kelebihan *framework Codeigniter* (CI) dibandingkan dengan *framework* PHP lain:

- a. Performa sangat cepat: salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scratch*, tapi *Codeigniter* sangat cepat bahkan mungkin bisa dibilang *codeigniter* merupakan *framework* yang paling cepat dibanding *framework* yang lain.
- b. Konfigurasi yang sangat minim (*nearly zero configuration*): tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan *codeigniter* dengan *setting standard*, anda hanya perlu mengubah sedikit saja file pada folder *config*.

- c. Memakai konsep MVC: Codeigniter memakai konsep MVC (*Model View Controller*), konsep modern yang banyak dipakai oleh *framework* PHP lainnya. Dengan adanya MVC, pengerjaan antara logika dengan *layout* telah dipisahkan, sehingga antara si *programmer* dan *designer* dapat santai melakukan tugasnya.
- d. Dokumentasi yang sangat lengkap: setiap paket *instalasi codeigniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami

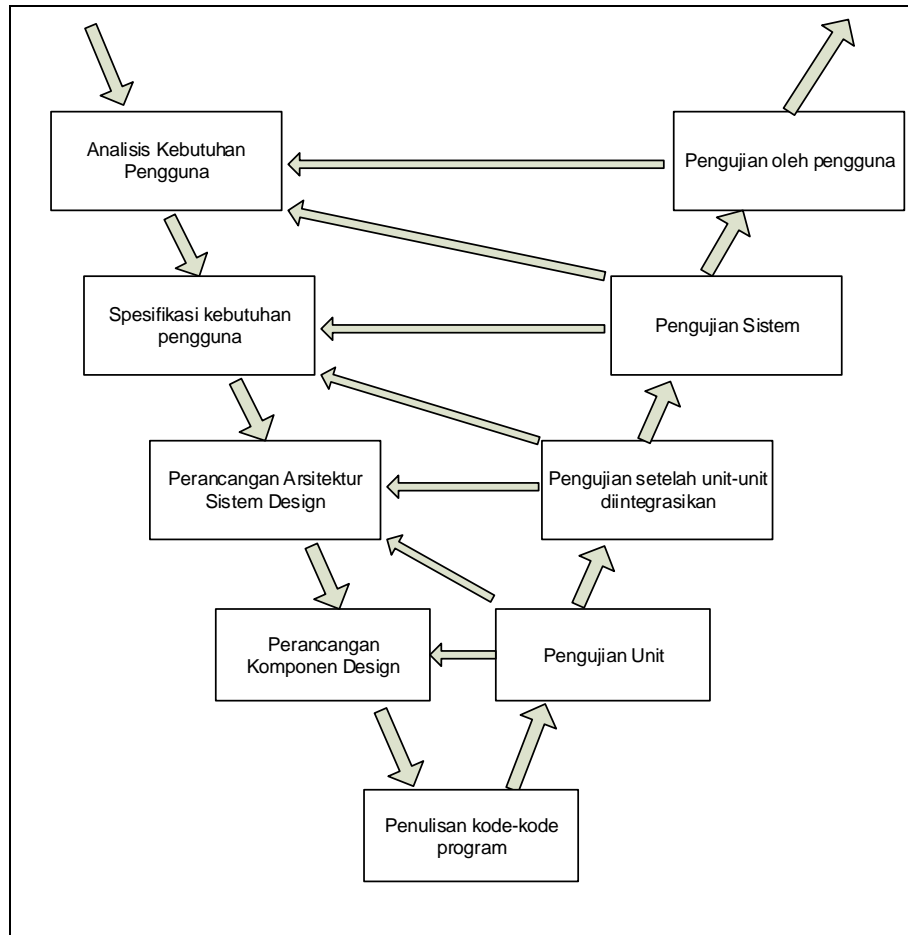
#### **2.4.7 Website.**

Menurut Hidayat (2010:2) *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam dan bergerak, animasi, suara, dan atau gabungan teks, gambar baik bersifat statis maupun *dinamis* yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

### **2.5 Metode Pengembangan Sistem**

Model-V ini merupakan perluasan dari model *waterfall*. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model *waterfall*. Jika dalam model *waterfall* proses dijalankan secara linear, maka dalam V-Model proses dilakukan bercabang. Disebut bercabang yaitu jika ditahap pengujian terjadi pesan error atau tidak sesuai dengan perancangan pengembangan software maka akan di koreksi ditahap sebelumnya begitupun dengan tahap

selanjutnya. Dalam V-Model ini digambarkan hubungan antara tahap pengembangan *software* dengan tahap pengujiannya (Anonim, 2016). Sedangkan menurut (Munnasar, 2010) Tahapan V-Model hampir sama dengan *waterfall*, hanya pada model ini tahapan pengujian dirinci untuk masing-masing tahap pada gambar. 2.2



**Gambar. 2.2** Model-V

Sumber : Pressman., “*Rekayasa Perangkat Lunak – Pendekatan Praktisi Edisi 7 (Buku 2)*”, 2017, hal. 47.

Berikut ini penjelasan mengenai tahapan pada metode pengembangan yang digunakan (Windi Eka Y.R, Saiful Bukhori, D. I. 2013), yaitu:



1. Analisis kebutuhan pengguna & Pengujian oleh pengguna

Tahap analisis kebutuhan pengguna sama seperti yang terdapat dalam model *waterfall*. Keluaran dari tahap ini adalah dokumentasi kebutuhan pengguna. Pengujian oleh pengguna merupakan tahap yang akan mengkaji apakah dokumentasi yang dihasilkan tersebut dapat diterima oleh para pengguna atau tidak.

2. Perancangan spesifikasi pengguna & Pengujian sistem keseluruhan

Dalam tahap ini analisis sistem mulai merancang sistem dengan mengacu pada dokumentasi kebutuhan pengguna yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi software yang meliputi organisasi sistem secara umum, struktur data, dan yang lain.

Pengujian sistem keseluruhan adalah pada tahap ini akan dilakukan pengujian dengan mencocokkan sistem dan kebutuhan yang diinginkan user

3. Perancangan arsitektur design & Pengujian unit integrasi.

Sering juga disebut *High Level Design*. Dasar dari pemilihan arsitektur yang akan digunakan berdasar kepada beberapa hal seperti: pemakaian kembali tiap modul, ketergantungan tabel dalam basis data, hubungan antar *interface*, detail teknologi yang dipakai.

Pengujian unit integrasi merupakan pengujian pada tahap ini dilakukan oleh pengembang pada seluruh unit sistem yang berkaitan dengan *input* dan *output* sistem. Jika ditemukan kesalahan, maka akan dilakukan pengujian lagi terhadap seluruh sistem.

4. Perancangan komponen design & Pengujian unit.

Sering juga disebut sebagai *Low Level Design*. Perancangan dipecah menjadi modul-modul yang lebih kecil. Setiap modul tersebut diberi penjelasan yang cukup untuk memudahkan programmer melakukan coding. Tahap ini menghasilkan spesifikasi program seperti: fungsi dan logika tiap modul, pesan kesalahan, proses input-output untuk tiap modul, dan lain-lain.

Pengujian unit ialah pengujian dilakukan pada semua unit untuk menguji alur sistem minimal satu kali.

## 5. Coding

Dalam tahap ini dilakukan pemrograman terhadap setiap modul yang sudah dibentuk.

## 2.6 Pengujian Sistem

Pengujian perangkat lunak sebenarnya merupakan sebuah proses verifikasi dan validasi. Verifikasi adalah tahapan dari rekayasa perangkat lunak untuk memastikan produk yang di hasilkan dari aktivitas pengembangan sesuai dengan spesifikasi yang di tentukan. Sedangkan validasi merupakan tahapan untuk memberikan penilaian produk sesuai dan memuaskan keinginan dari pemangku kepentingan (Pressman, 2010 : 550). Sedangkan menurut (Rosa dan Shalahuddin, 2016 : 272) Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan.

Proses pengujian perangkat lunak dilakukan untuk memastikan bahwa perangkat lunak yang dikembangkan sudah berjalan dengan semestinya. Pengujian perangkat lunak dilakukan dengan 4 tahap dapat dilihat pada tabel;

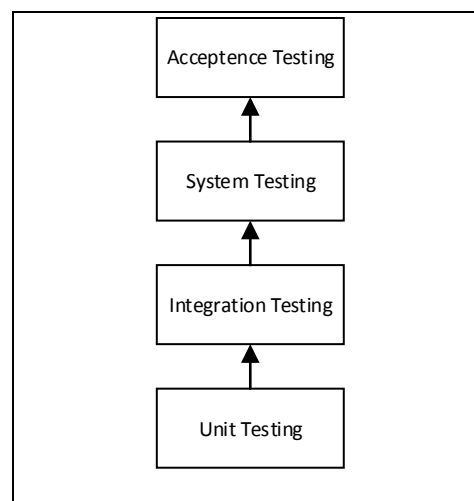
**Tabel 2.7** Metode Pengujian

<b>Teori Pressman</b>	<b>Model-V</b>	<b>Teknik Pengujian</b>	<b>Aspek Uji</b>
Verifikasi	<i>Unit Testing</i>	<i>White Box</i>	<i>Functionality</i>
	<i>Integration Testing</i>	<i>Black Box</i>	
Validasi	<i>System Testing</i>	<i>Stress testing</i>	<i>Compatibility</i>
	<i>Acceptance Testing</i>	<i>Playability Test</i>	<i>Playability</i>

Pada Tabel 2.7 merupakan tahapan pengujian pada metode pengembangan sistem menurut teori Pressman dan dihubungkan dengan metode pengembangan sistem Model-V, yang mana Verifikasi bagian dari pengujian *Unit Testing* dan *Integration Testing* teknik pengujian yang di pakai yaitu *white box* dan *black box* serta aspek yang diuji *Functionality*. Kemudian Validasi merupakan bagian dari pengujian *System Testing* dan *Acceptance Testing*, teknik pengujian yang dipakai *Stress Testing* dan *Playability*, aspek yang diuji *Compability* dan *Playability*.

### 2.6.1 Fase Pengujian

Pada Gambar 2.3 akan menggambarkan fase pengujian yang ada pada metode pengembangan sistem Model-V, berikut gambar fase pengujian;



**Gambar 2.3** Fase Pengujian Model-V

Terdapat empat fase pengujian didalam metode pengembangan sistem Model-V yaitu;

## 1. *Unit Testing*

*Unit testing* adalah proses pengujian perangkat lunak dimana masing-masing unit/komponen diuji. Tujuannya untuk meverifikasi bahwa setiap unit perangkat lunak sudah melakukan seperti apa yang telah dirancang. Menurut Pressmen, *unit testing* berfokus pada upaya verifikasi terhadap unit terkecil dari perancangan perangkat lunak. Pengujian unit berfokus pada logika pemrosesan internal dan struktur data didalam komponen.

Pada *unit testing*, pengujian dilakukan dengan dengan memeriksa bagian kode program secara terpisah dari bagian yang lain. Pengujian dapat langsung dilakukan setiap kali sebuah kode unit (*event, procedure, functon*) selesai dibuat. *Unit testing* dapat dilakukan dengan menggunakan metode *whitebox testing (functionality)*

## 2. *Integration Testing*

*Integration Testing* adalah proses pengujian perangkat lunak dimana unit individu digabungkan dan diuji sebagai sebuah kelompok. Sehingga pengujian ini mampu menampilkan kesalahan dalam interaksi antar unit. Menurut Pressman, pengujian integrasi adalah teknik untuk membangun arsitektur perangkat lunak, sementara pada saat yang sama melakukan pengujian untuk menemukan kesalahan terkait antarmuka. Tujuannya adalah untuk mengambil komponen yang diuji dan membangun struktur program yang telah ditentukan oleh perancangan.

Setelah melakukan *Unit/Component Testing*, langkah berikutnya adalah memeriksa bagaimana unit-unit tersebut bekerja sebagai suatu kombinasi, bukan lagi sebagai suatu unit yang individual. Sebagai contoh, kita memiliki sebuah proses yang dikerjakan oleh dua *function*, di mana satu *function* menggunakan hasil output dari *function* yang lainnya. Kedua *function* ini telah berjalan dengan baik secara individu pada *Unit Testing*.

Pada tahap *Integration Testing*, kita memeriksa hasil dari interaksi kedua *function* tersebut, apakah bekerja sesuai dengan hasil yang diharapkan. Kita juga harus memastikan bahwa seluruh kondisi yang mungkin terjadi dari hasil interaksi antar unit tersebut menghasilkan output yang diharapkan. *Integration testing* dapat dilakukan dengan pengujian *blackbox (functionality)*.

### 3. *System Testing*

*System Testing* adalah proses pengujian dimana perangkat lunak yang diuji sudah lengkap dan terintegrasi. Tujuan dari pengujian ini adalah untuk mengevaluasi kesesuaian sistem dengan persyaratan yang telah ditentukan. Menurut Pressman, pengujian sistem merupakan serangkaian pengujian yang berbeda-beda yang bertujuan untuk memverifikasi bahwa semua elemen sistem telah terintegrasi dengan baik dan menjalankan fungsi yang telah ditetapkan. *System Testing* mencakup pengujian aplikasi yang telah selesai dikembangkan. Karena itu, aplikasi harus terlihat dan berfungsi sebagaimana mestinya terhadap pengguna akhir.

*System Testing* dapat dilakukan menggunakan metode *Blackbox*, *nonfunctional test* (*configuration test*, *compability test*, *stress test*, *performance test* dan lain sebagainya

#### 4. *Acceptence Testing*

*Acceptance Testing* atau uji penerimaan adalah pengujian formal dilakukan untuk menentukan apakah sistem menerima kriteria penerimaan dan memastikan jika pengguna dapat menerima sistem. Tujuan dari pengujian ini adalah untuk mengetahui tingkat kelayakan dari perangkat lunak.

Seperti *Integration Testing*, *Acceptance Testing* juga meliputi pengujian keseluruhan aplikasi. Perbedaannya terletak pada siapa yang melakukan testing. Pada tahap ini, *end-user* yang terpilih melakukan testing terhadap fungsi-fungsi aplikasi dan melaporkan permasalahan yang ditemukan. Pengujian yang dilakukan merupakan simulasi penggunaan nyata dari aplikasi pada lingkungan yang sebenarnya. Proses ini merupakan salah satu tahap final sebelum pengguna menyetujui dan menerima penerapan sistem aplikasi yang baru. Karena itu pada tahap ini sudah tidak difokuskan untuk mengangkat permasalahan kecil seperti kesalahan pengetikan. Hal-hal minor seperti di atas sudah seharusnya ditangani selama *Unit/Component Testing* dan *Integration Testing*. *Acceptance test* dapat dilakukan menggunakan metode *Blackbox*, *usability* atau *playbility* .

Menurut Pressmen (2010) pengujian perangkat lunak adalah proses verifikasi dan validasi perangkat lunak yang diuji. Pada *unit testing* dan *integration testing* merupakan tahapannya verifikasi pengujian perangkat lunak yang dilakukan dengan metode *whitebox* dan *blackbox*. Kemudian pada tahap validasi pengujian perangkat lunak yaitu pada *system testing* dan *acceptance testing* dilakukan pengujian alpha untuk mengukur *compability* dan *playbility* perangkat lunak.

Variabel penelitian yang ada dalam penelitian Sistem Informasi Akademik berbasis web di SMK Negeri Kayu Agung ini adalah;

1. *Functionality* (fungsionalitas) merupakan kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
2. *Compability* (kompatibel) merupakan kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.
3. *Playability* (kemampuan untuk dimainkan) merupakan kemampuan perangkat lunak untuk mudah dipahami, dipelajari, digunakan, dan menarik bagi pengguna ketika digunakan dalam kondisi tertentu.

## **2.7 Tinjauan Pustaka**

Berikut ini merupakan jurnal yang peneliti ambil dari penelitian sebelumnya untuk digunakan sebagai referensi atau acuan penelitian yang akan dilakukan. Abdussomad, dkk yang berjudul “Sistem Informasi Akademik Berbasis Web Pada SMK PGRI Karawang” (2016). Maka dapat disimpulkan dari penelitian ini adalah untuk mempermudah guru dan siswa/i SMK PGRI 2 dalam

memperoleh informasi akademik khususnya pengolahan data nilai. Menghemat waktu dalam mendapatkan informasi akademik, karena tidak perlu datang ke sekolah untuk mendapatkan informasi mengenai sekolah dan mewujudkan sistem informasi berbasis IT (*information technology*). Metode pengembangan yang dipakai adalah *Waterfall*.

Firdaus, dkk yang berjudul “Sistem Informasi Akademik (SIA) Sekolah Menengah Kejurusan (SMK) Cendana Padang Panjang Dengan Menggunakan Bahasa Pemrograman yang Berbasis Object Oriented Programming (OOP)” (2010). Dari penelitian ini maka dapat dihasilkan sistem yang dibuat dapat memberikan kemudahan terhadap bagian tata usaha dalam mencetak laporan yang dibutuhkan oleh tata usaha pada SMK Cendana Padang Panjang. dengan menerapkan 5 outputnya yaitu; Laporan Nilai per mata pelajaran, laporan nilai per Siswa, laporan nilai Per Kelas, Raport Siswa dan Laporan data peringkat Siswa per kelas. Metode pengembangan yang dipakai adalah *Object Oriented Programming* (OOP).

Muh. Hasbi yang berjudul “Perancangan Sistem Informasi Akademik Pada SMK Negeri 2 Simbang Maros.” (2015). Hasil penelitian yang dihasilkan berupa sebuah aplikasi perangkat lunak yang dibangun menggunakan *macromedia dreamweaver 8* yaitu *html, php, MySql* berbasis web sehingga membentuk sistem informasi akademik pada SMK Negeri 2 Simbang Maros. Staf tenaga kependidikan yang sistem informasi akademik pada SMK Negeri 2 Simbang Maros Kabupaten Maros dapat mengelola data-data siswa, data pelajaran, data guru dan data nilai-nilai siswa. Metode pengembangan yang dipakai adalah *Prototype*.



Yustina Meisella Kristania, dkk “Sistem Informasi Akademik Berbasis Web pada SMK Negeri 2 Banyumas” (2017). Tujuan penelitian ini adalah untuk membuat website yang dapat digunakan untuk mengelola data guru, data siswa dan data akademik. Metode penelitian yang digunakan yaitu pengembangan perangkat lunak menggunakan model *waterfall*, pengamatan, wawancara, dan studi pustaka. Website dibangun menggunakan bahasa pemrograman *HTML*, *PHP*, *CSS*, *Javascript* dan *JQuery*. Serta menggunakan *MySQL* sebagai *database* dan *Apache* sebagai *web server*. Dengan adanya website ini, diharapkan dapat memudahkan proses penyebaran informasi, memudahkan guru dalam mengolah data akademik sehingga dapat meminimalisir terjadinya kerangkapan data keamanan data menjadi lebih terjamin.

Wiwid Rachma Pitaloka, dkk yang berjudul “Sistem Informasi Akademik Berbasis Web Pada SMK Muhammadiyah 4 Palembang” (2015). Sistem informasi akademik ini dapat membantu Tata Usaha dalam mengelola data siswa, menyusun jadwal mengajar Guru, penyampaian informasi akademik, membantu Guru dalam mengelola nilai raport siswa, serta membantu siswa dalam mendapatkan informasi akademik dengan mudah dan cepat. Metodologi yang digunakan adalah metodologi RUP (*Rational Unified Process*) dan metode perancangan sistem menggunakan pemodelan UML (*Unified Modeling Language*) yang terdiri dari *Fase Inception*, *Fase Elaboration*, *Fase Construction* dengan menerapkan sistem menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan database menggunakan *DBMS MySQL*, dan *Fase Transition*. Untuk mengidentifikasi masalah penulis menggunakan kerangka *PIECES*.

Dian Mulyani penelitiannya yang berjudul “Sistem Informasi Akademik Berbasis Web (Studi Kasus SMK Informatika Bina Generasi)” (2014). Tujuan penulisan ini adalah membangun sistem informasi akademik berbasis web yang dapat digunakan sebagai salah satu fasilitas di SMK Informatika Bina Generasi untuk penyajian informasi kepada siswa mengenai data nilai dan data absensi, serta menyajikan informasi kepada pengguna mengenai daftar pegawai, daftar pengajar, daftar kelas serta daftar jadwal mata pelajaran. Metode pengembangan yang dipakai adalah *Waterfall*.

Chandra Kesuma dan Lucky Rahmawati yang penelitiannya berjudul “Sistem Informasi Akademik Berbasis Web Pada SMK Purnama 2 Banyumas.” (2017). Pada penelitian ini maka dapat dihasilkan adalah Sistem Informasi Akademik berbasis web pada SMK Purnama 2 Banyumas dapat membantu mengolah dan menyajikan informasi bagi *user* baik itu guru, siswa dan masyarakat luas serta membantu guru dalam mengolah nilai dan menginformasikan nilai tersebut kepada siswa dan mampu mempermudah siswa dalam memperoleh informasi akademik dan mengakses nilai meskipun berada diluar sekolah. Serta metode pengembangan yang dipakai adalah *Waterfall*.

Cecep Bunyadin penelitiannya yang berjudul “Sistem Informasi Akademik Berbasis Web Pada SMK N 2 Adiwerna – Tegal” (2013). Hasil penelitian menunjukkan bahwa perancangan sistem informasi akademik di SMK N 2 Adiwerna dapat membantu staf, guru, dan bagian yang terkait pada sistem informasi akademik dalam pengolahan data akademik yang ada. Mengadakan pengujian terhadap sistem aplikasi dengan menggunakan metode Blackbox. Metode yang digunakan adalah metode *Waterfall*.

Marga Hartono, dkk yang berjudul “Sistem Informasi Akademik Berbasis Web Di SMK Negeri 1 Semboro” (2012). Hasil dari penelitian ini adalah untuk mempermudah dalam pengolahan data didalam manajemen sekolah sehingga informasi yang bermutu dapat disampaikan kepada masyarakat umum dan para siswa dapat mengakses informasi-informasi sekolah sehingga siswa lebih efektif dalam mendapatkan informasi serta dengan adanya sistem ini untuk dunia usaha dapat mengetahui kompetensi siswa-siswi sehingga dunia usaha dapat menyerap tenaga kerja dari SMK Negeri 1 Semboro.

Berdasarkan penjelasan tinjauan pustaka diatas adalah yang membedakan antara peneliti yang peneliti ambil dengan penelitian sebelumnya adalah kebanyakan penelitian sebelumnya menggunakan metode pengembangan *Waterfall*, *OOP*, dan *RUP* kemudian didalam pembahasannya hanya membahas data guru, siswa, jadwal pelajaran serta penilaian. Sedangkan penelitian yang dibuat penulis beberapa membedakan diantaranya metode pengembangan yaitu metode pengembangan dari *Waterfall* yaitu metode Model *V* metode ini adalah pengembangan dari metode *Waterfall*, serta didalamnya sedikit membedakan yaitu pendaftaran siswa baru serta pengolaan data alumni.