

BAB II

LANDASAN TEORI

2.1 Ayat Al-Quran yang Berhubungan dengan Penelitian

Banyak ayat-ayat Al-Qur'an yang membahas tentang ilmu pengetahuan salah satunya terdapat pada Firman Allah dalam Surah Al-Kahfi Ayat 66 yang berbunyi:

قَالَ لَهُ مُوسَىٰ هَلْ أَتَّبِعُكَ عَلَىٰ أَنْ تُعَلِّمَنِي مِمَّا عَلَّمْتَ رُشْدًا

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).

Dari ayat ini dapat diambil beberapa pokok pemikiran sebagai berikut: Kaitan ayat ini dengan aspek pendidikan bahwa seorang pendidik hendaknya: Menuntun anak didiknya. Dalam hal ini menerangkan bahwa peran seorang guru adalah sebagai fasilitator, tutor, tentor, pendamping dan yang lainnya. Peran tersebut dilakukan agar anak didiknya sesuai dengan yang diharapkan oleh bangsa neraga dan agamanya.

Ada juga ayat Al-Qur'an yang berkaitan dengan ilmu pengetahuan yang berbunyi :

وَإِذْ قَالَ لَقْمَنُ لِبَنِيهِ وَهُوَ يَعِظُهُ يَا بُنَيَّ لَا تُشْرِكْ بِاللَّهِ إِنَّ الشِّرْكَ لَظُلْمٌ عَظِيمٌ

Artinya: *"Dan (Ingatlah) ketika Luqman Berkata kepada anaknya, di waktu ia memberi pelajaran kepadanya: "Hai anakku, janganlah kamu mempersekutukan Allah, Sesungguhnya mempersekutukan (Allah) adalah benar-benar kezaliman yang besar"*(QS. Luqman:13).

Dari ayat ini dapat diambil kesimpulan : Dalam mendidik hendaknya menggunakan pendekatan yang bersifat kasih sayang, sesuai makna seruan Lukman kepada anak-anaknya, yaitu "Yaa Bunayyaa" (Wahai anak-anakku), seruan tersebut menyiratkan muatan kasih sayang/sentuhan kelembutan dan kemesraan, tetapi dalam koridor ketegasan dan kedisiplinan, bukan berarti mendidik dengan keras. (Kompetensi Personal).

2.2 Teori yang Berhubungan dengan Penelitian

2.2.1 Sistem

Sistem merupakan seperangkat unsur yang saling terkait dalam suatu relasi diantara unsur-unsur tersebut dengan lingkungan (Ludwig Von Bartalanty dalam Paryati dan Ardhana, 2008:2). Syarat-syarat sistem yaitu :

1. Sistem harus dibentuk untuk menyelesaikan tujuan.
2. Elemen sistem harus mempunyai rencana yang diterapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

Sistem dapat diartikan sebagai serangkaian komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu. Jadi dalam istilah sistem terkandung tiga elemen penting, seperti rangkaian komponen, interaksi dan kerja sama serta tujuan (Soeherman dan Pinontoan, 2008).

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

2.2.2 Informasi

Informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan tersebut bisa menjadi informasi, hasil pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut (Darmawan dan Fauzi, 2013:2).

Informasi adalah hasil pemrosesan, memanipulasi, dan pengorganisasian/penataan dari sekelompok data yang mempunyai nilai pengetahuan bagi penggunanya. (Sutabri, 2014:2) .

Berdasarkan kutipan yang peneliti ambil dapat disimpulkan bahwa informasi adalah data yang diolah menjadi bentuk yang berguna untuk membuat keputusan. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode ataupun cara-cara tertentu. Kualitas suatu informasi tergantung dari 3 (tiga) hal, yaitu : informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*). Berikut penjelasan kualitas informasi(Sutabri, 2012:41):

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*timeline*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk orang satu dengan yang lain berbeda.

2.2.3 Sistem Informasi

Sistem informasi merupakan kombinasi teratur dari orang-orang, perangkat lunak (*software*), perangkat keras (*hardware*), jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi (Yakub, 2012:17).

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi operasi bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu

sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran (Sutabri, 2012:46).

Berikut adalah keenam blok bangunan (*building block*) diantaranya :

1. Blok masukan (*input block*).
2. Blok model (*model block*).
3. Blok keluaran (*Output block*).
4. Blok teknologi (*technology block*).
5. Blok basis data (*database block*).
6. Blok kendali (*control block*).

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegrasi dan saling melengkapi dengan menghasilkan output yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

2.2.4 Sistem Informasi 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. Sedangkan 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 (Liatmaja, 2013).

Sistem informasi akademik merupakan bagian dari pengendalian internal suatu kegiatan akademik yang meliputi pemanfaatan sumber daya manusia, dokumen, teknologi, dan prosedur oleh manajemen untuk memecahkan masalah akademik. Sistem informasi akademik ini merupakan sistem informasi berdasarkan pada aktivitas manajemen. Sistem ini menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) komputer, prosedur pedoman, model manajemen dan keputusan, dan sebuah “*database*” (Suzanto dan Sidharta, 2015).

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi akademik adalah suatu disiplin akademik atau bidang studi, juga merupakan suatu cabang pengetahuan yang diajarkan atau diteliti ditingkat sekolah dan perguruan tinggi.

2.2.5 MVC (*Model View Controller*)

Sebuah *framework* aplikasi web biasanya mengimplementasikan pola desain yang disebut *Model*, *View* dan *Controller* atau yang biasa disebut sebagai MVC.

Modul *model* membuat kelas-kelas yang mewakili tabel pada *database* yang mempunyai *instances* yang digunakan untuk memanipulasi database. *Model* biasanya digunakan sebagai penghubung antara modul *controller* dengan *database* ketika *controller* ingin mengambil dan menggunakan data di *database*.

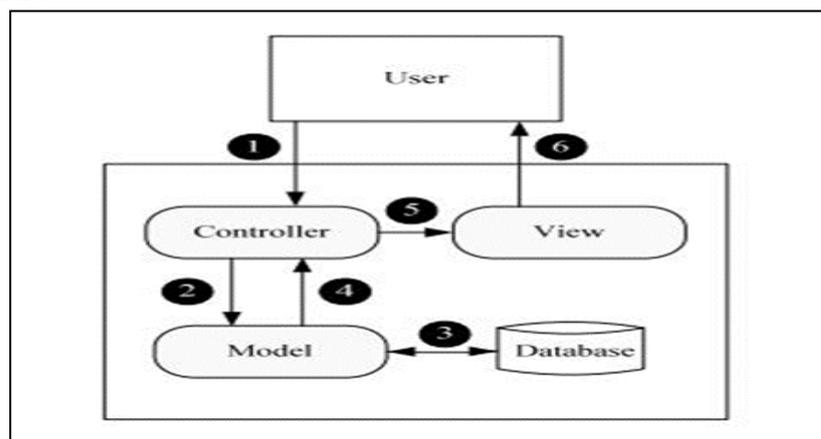
Modul *controller* adalah kelas-kelas yang dibuat oleh programmer untuk menangani logika program dan *user events*. Pada aplikasi yang menggunakan pola MVC *controller* bertindak sebagai otak dari sistem, menjembatani hubungan

antara *model* dan *view*. *Controller* juga berfungsi menerima *request* dari *user* dan kemudian memprosesnya.

Modul *view* berfungsi untuk menerima dan menampilkan data yang dikirim oleh *controller*. *View* dalam aplikasi berbasis *website* biasanya berbentuk kumpulan halaman HTML.

MVC membantu mengurangi kompleksitas dari pembuatan desain dan menambah fleksibilitas dan pemakaian kembali (re-use) kode (Cui et al, 2009).

Berikut adalah ilustrasi dari konsep MVC seperti yang ditunjukkan pada Gambar 2.1 :



Gambar 2.1 Ilustrasi MVC

2.2.6 Framework Laravel

Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. *Laravel* dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di *Github*. Sama seperti *framework* PHP lainnya, *Laravel* dibangun dengan basis MVC (*Model-ViewController*). *Laravel* dilengkapi *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging*

bundle dan instalasi bundle. *Framework Laravel* dibuat oleh Taylor Otwell, proyek *Laravel* dimulai pada April 2011. Awal mula proyek ini dibuat karena Otwell sendiri tidak menemukan *framework* yang *up-to-date* dengan versi PHP. Mengembangkan *framework* yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya. Dikarenakan beberapa keterbatasan tersebut, Otwell membuat sendiri *framework* dengan nama *Laravel*. Oleh karena itu *Laravel* menisyaratkan PHP versi 5.3 keatas (Rohman, 2014).

Laravel merupakan *framework* aplikasi web yang ekspresif dan *syntax* yang elegan. Kami membangun *framework* ini dengan menyenangkan dan dari pengalaman-pengalaman kreatif agar menjadi produk yang memuaskan. *Laravel* berusaha mencoba menjadi *framework* yang mudah digunakan dengan mengurangi tugas-tugas umum yang sering digunakan dalam sebagian besar proyek-proyek web seperti otentikasi, *routing*, *session*, dan *caching*. *Laravel* membuat proses *development* yang menyenangkan bagi pengembang tanpa mengurangi fungsionalitas aplikasi. Dengan harapan, pengembang dapat membuat rangkaian kode-kode terbaik. *Laravel* berusaha untuk menggabungkan yang terbaik dari apa ada dalam *framework* web lain, termasuk *framework* yang menggunakan bahasa lain, seperti Ruby on Rails, ASP.NET MVC, dan Sinatra. *Laravel* merupakan *framework* yang mudah diakses, powerful dan menyediakan tools yang diperlukan untuk skala aplikasi besar. *Laravel* juga merupakan sebuah aplikasi luar biasa dari sebuah kumpulan program kontrol, sistem migrasi yang ekspresif dan dukungan tools yang Anda butuhkan dalam menguji aplikasi Anda yang terintegrasi dengan beberapa aplikasi lainnya.

2.2.7 Mysql

MySQL adalah salah satu jenis database server yang menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Dengan menggunakan script PHP dan PERL software database ini dapat berfungsi atau berjalan pada semua platform system operasi yang bias digunakan (Windows, Linux, OS/2, berbagai varian Unix) (Kadir,2008).

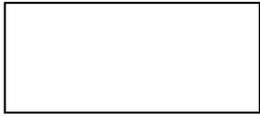
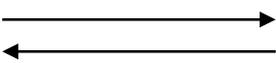
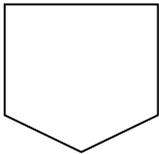
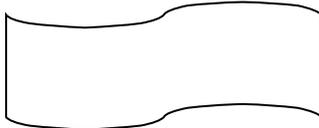
2.3 Teori yang berkaitan dengan *Tools* yang digunakan

2.3.1 *Flowchart*

Flowchart (Bagan Alir) adalah bagan (*Chart*) yang menunjukkan alir (*Flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan dokumentasi. *Flowchart* terbagi atas lima jenis, yaitu :

- 1 *Flowchart* Sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang di kerjakan dalam sistem secara keseluruhan dan menjelasakan dari prosedur-prosedur yang di dalam sistem.
- 2 *Flowchart* Dokumen, yaitu bagan alir yang menunjukkan arus dari laporan.
- 3 *Flowchart* Skematik, yaitu untuk menggambarkan prosedur di dalam sistem.
- 4 *Flowchart* Program, yaitu bagan yang menjelaskan secara rinci langkah-langkah proses program.
- 5 *Flowchart* Proses, yaitu teknik penggambaran dan menganalisis langkah-langkah dalam suatu prosedur atau sistem.

Tabel 2.1 Simbol-simbol *Flowchart*

No	Gambar	Nama	Keterangan
1		Terminator	Menggambarkan asal data atau tujuan data
2		Proses	Menunjukkan kegiatan proses dari operasi program komputer
3		Garis Alir	Menunjukkan arus dari proses
4		Penghubung	Penghubung bagian-bagian <i>flowchart</i> yang berbeda pada halaman berbeda
5		Dokumen	Menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau komputer

Sumber : (Jogiyanto H.M, 2005: 796).

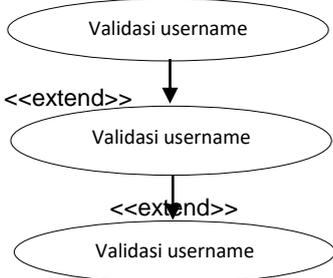
2.3.2 UML (*Unified Modelling Language*)

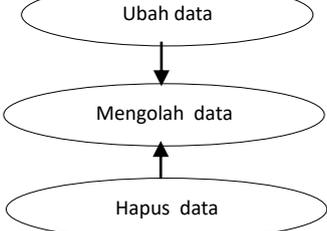
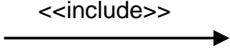
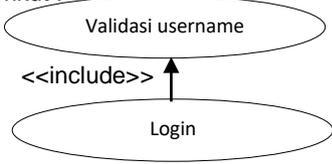
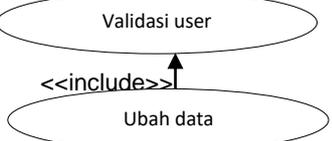
2.3.2.1 *Use Case Diagram*

Menurut Rosa & Shalahuddin (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*.

1. Aktor merupakan 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 orang, tetapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor

Tabel 2.2 Simbol *Use case* Diagram

No.	Simbol	Keterangan
1	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
2	<p>Aktor /actor</p> 	Orang atau proses atau sistem lain yang berinteraksi dengan 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 <i>Actor</i>
3	<p>Asosiasi/<i>Association</i></p> 	Komunikasi antara aktor dengan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<p>Ekstensi/<i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal :</p> 

5	<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya :</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
6	<p>Include</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :</p> <ul style="list-style-type: none"> • Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan yang dijalankan misal pada kasus berikut :  <ul style="list-style-type: none"> • Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut :  <p>Kedua interpretasi di atas dapat dianut salah satu keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

(Sumber : Rosa&Shalahuddin, 2015 hal 156).

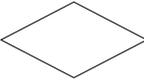
2.3.2.2 Activity Diagram

Diagram *activity* adalah 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. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan *interface* tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Tabel 2.3 Simbol *Activity Diagram*

No	Simbol	Keterangan
1	Status Awal 	Status awal aktivitas system, sebuah diagram aktivitas memiliki sebuah awal
2	Aktivitas 	Aktivitas yang dilakukan system, aktivitas biasanyaawali dengan kata kerja
3	Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/ join 	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.
	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Rosa&Shalahuddin, 2015 hal 162-163).

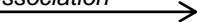
2.3.2.3 Class Diagram

Diagram kelas atau *class* diagram adalah menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas diagram memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Diagram kelas dibuat agar programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Tabel 2.4 Simbol *Class* Diagram

Simbol	Deskripsi
	Kelas pada struktur sistem
Antarmuka/interface  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi/association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/dependency 	Relasi antar kelas dengan makna Kebergantungan antar kelas
	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

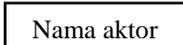
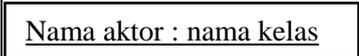
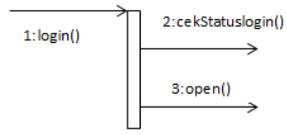
(Sumber : Rosa&Shalahuddin, 2015 hal 146-147).

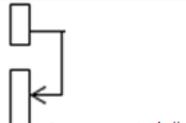
2.3.2.4 Sequence Diagram

Diagram sekuen adalah menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan

diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 2.5 Simbol *Sequence* Diagram

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p> <p>Atau</p>  <p>Nama aktor</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p>  <p>Nama aktor : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu Aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya misalnya</p>  <p>Maka cek Status Login () dan open () dilakukan di dalam metode login () aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i> << create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarahkan pada objek yang dibuat</p>
<p>Pesan tipe call 1: nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>

	 <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
Pesan tipe send 1: masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 1:keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy << destroy >> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber : Rosa&Shalahuddin, 2015 hal 165-167).

2.3.3 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah bentuk paling awal dalam melakukan perancangan basis data relasional. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional (Patmanthara, 2012:28).

Adapun simbol-simbol yang digunakan dalam ERD (*Entity Relationship Diagram*) adalah sebagai berikut :

Tabel 2.6 Simbol-simbol ERD (*Entity Relationship Diagram*)

No	Simbol	Deskripsi
1.	Entitas / <i>entity</i> 	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.

2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
4.	Asosiasi / <i>association</i> 	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakai.

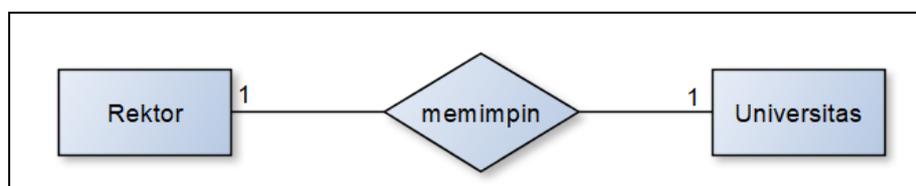
Sumber: (Rosa & Shalahuddin, 2016: 50-51).

Dalam ERD terdapat Kardinalitas, Menurut Fathansyah (2015:82-84) Kardinalitas mengacu pada berapa kali instance dari suatu entitas dapat berelasi dengan instance lain di entitas yang berbeda, yaitu:

1. Satu ke Satu (*One to One*)

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B dan begitu juga sebaliknya.

Contoh : Satu Rektor memimpin Satu Universitas dan sebaliknya Satu Universitas dipimpin oleh Satu Rektor. Berikut pada Gambar 2.2 contoh relasi satu ke satu (*One to One*) :

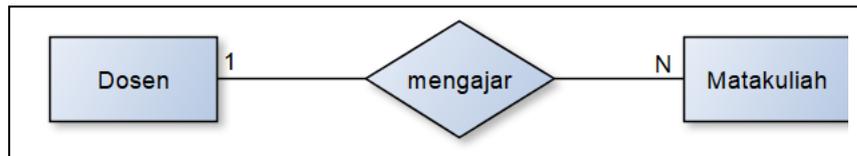


Gambar 2.2 Satu ke Satu

2. Satu ke Banyak atau Banyak ke Satu (*One to Many* atau *Many to One*)

Yang berarti satu entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya.

Contoh : Satu Dosen mengajar banyak Matakuliah, dan banyak Matakuliah diajarkan oleh satu Dosen. Berikut pada Gambar 2.3 contoh relasi satu ke banyak atau banyak ke satu (*One to Many* atau *Many to One*) :

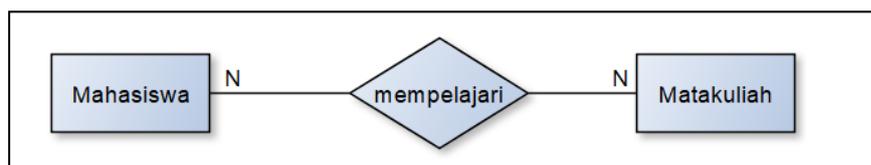


Gambar 2.3 Satu ke Banyak

3. Banyak ke Banyak (*Many to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, begitu juga dengan sebaliknya.

Contoh: Banyak Mahasiswa mempelajari banyak Matakuliah, dan banyak Matakuliah yang dipelajari oleh banyak Mahasiswa. Berikut pada Gambar 2.4 contoh relasi banyak ke banyak (*Many to Many*) :



Gambar 2.4 Banyak ke Banyak

2.4 Pengujian Sistem

Pengujian perangkat lunak merupakan persentase terbesar dari upaya teknis dalam proses perangkat lunak. Apapun jenis perangkat lunak yang anda bangun, strategi untuk perencanaan pengujian yang sistematis, pelaksanaan, dan kontrol

dimulai dengan mempertimbangkan elemen-elemen kecil dalam perangkat lunak dan bergerak keluar terhadap program secara keseluruhan. Tujuan pengujian perangkat lunak adalah untuk menemukan kesalahan (Roger S. Presman, 2012:580).

2.4.1 *BlackboxTesting*

Pengujian kotak hitam (*Black-Box Testing*) berkaitan dengan pengujian-pengujian yang dilakukan pada antarmuka perangkat lunak. Pengujian kotak hitam mengkaji beberapa aspek fundamental dari suatu sistem/perangkat lunak dengan sedikit memperhatikan struktur logis internal dari perangkat lunak. Pengujian kotak hitam berupaya menemukan kesalahan dalam kategori yaitu fungsi salah satu atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan perilaku atau kinerja dan kesalahan inisialisasi dan penghentian. Tidak seperti pengujian kotak putih, yang dilakukan pada awal proses pengujian, pengujian kotak hitam cenderung diterapkan selama tahap pengujian (Roger S. Presman, 2012:597).

Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut :

1. Fungsi yang salah atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses basis data eksternal
4. Kesalahan perilaku atau kinerja
5. Kesalahan inisialisasi dan penghentian.

2.5 Penelitian Sebelumnya

Agar penelitian ini dapat dipertanggung jawabkan, maka peneliti memberikan gambaran penelitian terdahulu yang berhubungan dengan Metode *Prototype* dan *Frame Laravel*, yaitu:

Penelitian yang dilakukan Ichsan dkk (03 Desember 2016) yang berjudul “Membangun Sistem Informasi Manajemen Kerja Praktik Berbasis *Website* Dengan Metode Iterative Incremental (Modul Mahasiswa)”. Menghasilkan sebuah kesimpulan bahwa penelitian ini Untuk mengatasi permasalahan tersebut, dibangun sebuah aplikasi *website* kerja praktik dengan menggunakan metode *iterative & incremental*. *Website* tersebut dapat diakses oleh mahasiswa yang melakukan kerja praktik. *Website* ini memberikan informasi terkait kerja praktik berupa tata cara pelaksanaan beserta pembuatan laporan. Sehingga memudahkan mahasiswa dalam menemukan informasi tentang kerja praktik dan memudahkan fakultas dalam menata dokumen setiap mahasiswa yang melakukan kerja praktik.

Penelitian yang dilakukan Lilis Yuningsih (10 Agustus 2017) yang berjudul “Implementasi Framework *Laravel* Pada Aplikasi Digitalisasi Arsip Sekretariat Organisasi Mahasiswa STMIK STIKOM Bali”. Menghasilkan sebuah kesimpulan bahwa penelitian ini membahas tentang salah satu bidang kepanitiaan yang memiliki peran penting dalam inti pelaksanaan kegiatan adalah bidang kesekretariatan. Bidang ini berfungsi sebagai penanggung jawab administrasi dalam seluruh kegiatan, mulai dari surat-menyurat, proposal kegiatan, hingga dalam penyusunan laporan pertanggungjawaban. Saat ini data arsip dari setiap kegiatan masih hanya berupa hardcopy.

Penelitian yang dilakukan Firma dkk (Juni 2016) yang berjudul “Implementasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel”. Menghasilkan sebuah kesimpulan bahwa Penelitian ini berfungsi untuk penggunaan Sistem Informasi Akademik yang dibutuhkan semua instansi sekolah. Selain memudahkan dalam dokumentasi, Sistem Informasi akademik juga mampu mengurangi kinerja waktu dalam memasukkan data masih menggunakan cara manual.

Penelitian yang dilakukan Ahmad dkk (Juni 2017) yang berjudul “Rancangan Bagun Aplikasi Sistem Informasi Manajemen Laboratorium Biomedik Fakultas Kedokteran Universitas Brawijaya”. Menghasilkan sebuah kesimpulan bahwa Penelitian ini berfokus pada sistem informasi manajemen laboratorium biomedik yang dirancang ulang sesuai dengan kebutuhan pengguna berdasarkan tahapan analisis kebutuhan komprehensif dan perancangan yang mendetail agar nantinya dapat diimplementasi secara lengkap sesuai dengan kebutuhan pengguna.

Penelitian yang dilakukan Stefan dkk (29 Desember 2016) yang berjudul “Rancangan Bangun Sistem Informasi Manajemen Mutu Perusahaan Menggunakan *Framework Laravel* dan *Materialize*”. Menghasilkan sebuah kesimpulan bahwa Penelitian ini bertujuan merancang aplikasi sistem informasi manajemen mutu untuk membantu dalam melakukan monitoring capaian mutu dan temuan audit, pendistribusian jadwal audit, serta membantu dalam pengelolaan dokumen.

Penelitian yang dilakukan O.K Saddam dkk (Juli 2018) yang berjudul “Sistem Informasi Deteksi Kehadiran dan Media Penyampaian Pengumuman Dosen Menggunakan Tekni Pengenalan *QR CODE*”. Menghasilkan sebuah kesimpulan bahwa penelitian ini merupakan sebuah aplikasi berbasis web yang dapat mendukung proses penyebaran informasi di kampus Universitas Muhammadiyah Riau.

Penelitian yang dilakukan A.A. Gede dkk (28 Oktober 2017) yang berjudul “Sistem Informasi Pembimbing Skripsi Online Berbasis Web (Studi Kasus : FTK, UNDIKSHA)”. Menghasilkan sebuah kesimpulan bahwa Penelitian ini bertujuan untuk membuat rancangan serta mengimplementasikan rancangan sistem skripsi online berbasis web. Penelitian ini dilakukan dengan model pengembangan SDLC (*System Development Life Cycle*) yaitu melalui tahapan requirement analysis, system design, implementation, testing, maintenance. Pengembangan sistem skripsi online berbasis web ini menggunakan bahasa pemrograman PHP dan basis data MySQL.

Penelitian yang dilakukan Farizan (Mei 2017) yang berjudul “Penggunaan Framework Laravel Dalam Rancangan Bangun Modul Back-End Artikel Website Bisnisbisnis.ID”. Menghasilkan sebuah kesimpulan bahwa Penelitian ini lebih difokuskan pada pembuatan Modul Back-End Artikel dari website Bisnisbisnis.ID. Website dibangun dengan menggunakan Framework Laravel dan MySql sebagai database. Dari hasil penelitian ini didapatkan hasil yaitu modul artikel telah berhasil dibuat dan berdasarkan hasil pengujian menggunakan

metode Black-Box testing didapati kesimpulan bahwa fungsi yang dibuat sudah sesuai dengan requirement system.

Penelitian yang dilakukan Rendi dkk (2017) yang berjudul “Rancang Bangun Sistem Informasi Manajemen Administrasi Surat Menggunakan *Framework* Laravel Pada Jurusan Teknik Elektro”. Menghasilkan sebuah kesimpulan bahwa Tujuan dari penelitian adalah merancang sistem informasi manajemen yang dapat membantu administrasi serta pengelolaan surat yang ada pada Jurusan Teknik Elektro dengan pengembangan menggunakan framework Laravel. Metode pengembangan yang digunakan adalah metode waterfall.

Penelitian yang dilakukan Andrie dkk (2016) yang berjudul “Sistem Penjadwalan Kuliah Menggunakan Metode Algoritma Genetika (Studi Kasus: Fakultas Kedokteran Dan Kesehatan Universitas Muhammadiyah Jakarta)”. Menghasilkan sebuah kesimpulan bahwa Pada Penelitian ini bertujuan merancang Sistem Informasi Penjadwalan Perkuliahan Berbasis Web dengan Metode Algoritma Genetika yang merupakan sarana terkait proses penentuan jadwal perkuliahan. Sistem ini digunakan untuk FKK UMJ bidang akademik dalam membuat jadwal perkuliahan, serta memudahkan staf akademik, dosen dan mahasiswa mengakses data penjadwalan kuliah.

Penelitian yang dilakukan Achmad dkk yang berjudul Implementasi *Framework Laravel* Pada Sistem Informasi Pemesanan Penggunaan Lapangan Futsal Berbasis Web Di Zona6 Futsal Semarang”. Menghasilkan sebuah kesimpulan bahwa pada penelitian ini adalah permasalahan tentang penyewaan

lapangan futsal di zona6 futsal semarang, proses pengelolaan lapangan futsalnya masih dilakukan secara konvensional.

Pada penelitian ini bertujuan untuk membangun sistem informasi akademik berbasis web menggunakan *Framework Laravel* pada SMK Kesehatan Bhakti Persada. Menghasil sebuah sistem yang memberikan informasi yang dibutuhkan sekolah guna untuk mengambil keputusan yang tepat, ada beberapa informasi seperti persentase nilai siswa, pemberian nilai terendah yang di lakukan guru, jumlah siswa minat dari smp mana sajah informasi.