



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Penelitian Sebelumnya

Penelitian – penelitian terdahulu berfungsi sebagai pendukung untuk melakukan penelitian. Penelitian – penelitian sebelumnya telah mengkaji masalah pembagian kerja dan upah yang masing–masing berpengaruh terhadap prestasi kerja, dan beberapa penelitian lain yang masih memiliki kaitan dengan variabel dalam penelitian ini.

UMMN

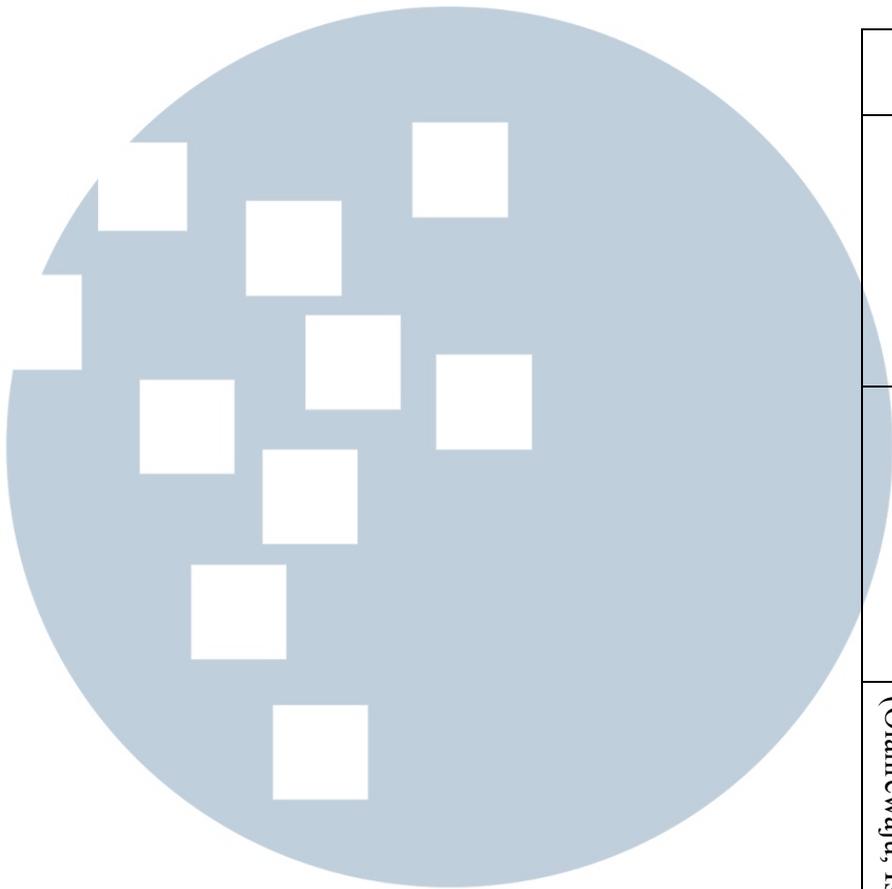
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tabel 2.1 Penelitian Sebelumnya

No	Nama Penulis (Tahun)	Judul	Metode	Hubungannya
1.	Weksi Budiaji (2013)	Skala Pengukuran dan Jumlah Respon Skala Likert	Skala likert adalah skala pengukuran yang dikembangkan oleh Likert (1932). Skala likert mempunyai empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor/nilai yang merepresentasikan sifat individu, misalkan pengetahuan, sikap, dan perilaku. Dalam proses analisis data, komposit skor, biasanya jumlah atau rata-rata, dari semua butir pertanyaan dapat digunakan. Penggunaan jumlah dari semua butir pertanyaan valid karena setiap butir pertanyaan adalah indikator dari variabel yang direpresentasikannya. (Budiaji, 2013)	Penggunaan skala Likert untuk menghitung <i>test score</i> pada UAT sistem informasi klaim ini.
2.	Emre Isikligil, Semih Samakay, Deniz Kilinc (2017)	<i>A Prototype Framework for High Performance Push Notifications</i>	Sistem yang diusulkan menggunakan layanan <i>cloud</i> GCM dan APNS untuk mengirim <i>mobile push notifications</i> ke perangkat yang menjalankan Android dan iOS, secara berurutan. Pertama, masing-masing perangkat mendaftarkan diri ke layanan <i>cloud</i> ini dan	Penggunaan Firebase <i>Cloud Messaging</i> sebagai pendukung fitur <i>push notification</i> yang akan digunakan untuk memberitahu atasan ketika ada klaim yang baru diajukan, memberitahu karyawan status klaim tersebut disetujui atau ditolak, dan

No	Nama Penulis (Tahun)	Judul	Metode	Hubungannya
3.	Ishaq Jound, Hamed Halimi (2016)	<i>Comparison of performance between Raw SQL and Eloquent ORM in Laravel</i>	<i>Load Testing</i> menempatkan permintaan pada sistem untuk mengukur respons, perilaku sistem yang ditentukan dengan meletakkannya di bawah beban maksimum dan normal. <i>Stress Testing</i> menempatkan sistem pada titik puncak untuk menentukan stabilitas sistem. Mereka menyajikan struktur database untuk aplikasi blog mereka menggunakan model data logis. Setiap tes kinerja memiliki <i>route</i> sendiri dan berisi operasi basis data yang mereka lakukan dengan mengunjungi <i>route</i> tersebut. (Jound & Halimi, 2016)	Penggunaan Eloquent ORM sebagai implementasi <i>ActiveRecord</i> sederhana untuk bekerja dengan database. Setiap tabel basis data memiliki "Model" yang sesuai yang digunakan untuk berinteraksi dengan tabel di <i>database</i> .
4.	Rashidah F. Olanrewaju, Thouhedul Islam, N. Ali (2015)	<i>An Empirical Study of the Evolution of PHP MVC Framework</i>	Untuk mengevaluasi kinerja empat kerangka kerja PHP yang disebutkan; CodeIgniter (CI), Symfony, CakePHP dan Laravel. Cara terbaik untuk melakukan perbandingan adalah	Penggunaan <i>framework</i> Laravel dalam mengembangkan API yang akan membatani aplikasi Android dengan <i>database</i> dan digunakan juga dalam

No	Nama Penulis (Tahun)	Judul	Metode	Hubungannya
			<p>dengan menerapkan beberapa kriteria seperti permintaan per detik, rata-rata beban sistem, penggunaan memori, jumlah panggilan fungsi dan jumlah file yang diperlukan pada masing-masing MVC. Hasil yang diperoleh dari mengevaluasi empat kerangka kerja: CodeIgniter (CI), Symfony, CakePHP dan Laravel menggunakan kriteria seperti permintaan per detik, rata-rata beban sistem, penggunaan memori, jumlah panggilan fungsi dan jumlah file yang diperlukan serta fasilitas yang tersedia di masing-masing kerangka. Ini menunjukkan bahwa Laravel menggunakan kerangka kerja MVC lainnya. Permintaan per detik setinggi 3.000 dicatat untuk Laravel dibandingkan dengan yang lain seperti CakePHP dengan serendah 750 permintaan per detik. Hasil yang diperoleh untuk semua parameter lain seperti penyimpanan, panggilan fungsi, jumlah <i>file</i>, waktu <i>respons</i> dll. Menunjukkan bahwa Laravel memiliki</p>	<p>mengembangkan <i>Dashboard</i>. <i>Reimbursement</i></p>



UMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

No	Nama Penulis (Tahun)	Judul	Metode	Hubungannya
			Flexibilitas yang sangat besar. (Olanrewaju, Islam, & Ali, 2015)	

Berdasarkan hasil penelitian yang terdahulu, penelitian ini menggunakan PHP sebagai bahasa pemrograman untuk membangun aplikasi berbasis *web* dan Java sebagai bahasa pemrograman yang dipakai untuk membangun aplikasi berbasis *mobile*.

## 2.2. Teori Terkait

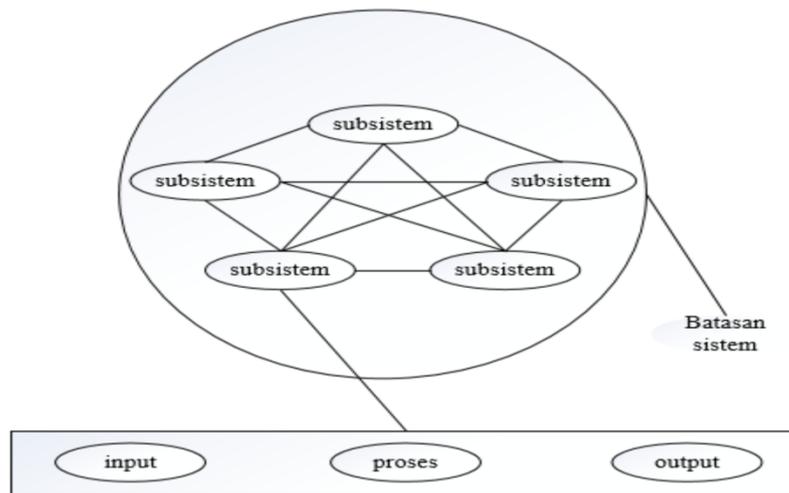
### 2.2.1 Pengertian Sistem

Dalam bidang sistem informasi, sistem diartikan sebagai sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur (Mulyanto, 2009).

### 2.2.2 Karakteristik Sistem

Suatu sistem mempunyai beberapa karakteristik, yaitu komponen atau elemen (*component*), batasan sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), pengolah (*process*), keluaran (*output*), sasaran (*objective*), atau tujuan (Mulyanto, 2009) seperti pada gambar 2.1

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



**Gambar 2.1 Karakteristik Sistem**

Sumber: (Mulyanto, 2009)

a. **Komponen (*component*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan.

Apabila suatu sistem merupakan salah satu dari komponen sistem lain yang lebih besar, maka akan disebut dengan *subsistem*, sedangkan sistem yang lebih besar tersebut adalah lingkungannya.

b. **Batasan (*boundary*)**

Batas sistem merupakan pembatas atau pemisah antara suatu sistem dengan sistem yang lainnya atau dengan

lingkungan luarnya. Batas suatu sistem juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar (*environment*)

Lingkungan luar adalah apapun diluar batas dari sistem yang dapat memengaruhi operasi sistem, baik pengaruh yang menguntungkan ataupun yang merugikan. Pengaruh yang menguntungkan ini tentunya harus dijaga sehingga akan mendukung kelangsungan operasi sebuah sistem. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan sebuah sistem.

d. Penghubung (*interface*)

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Penghubung inilah yang akan menjadi media yang digunakan data dari masukan (*input*) hingga keluaran (*output*). Dengan adanya penghubung, suatu subsistem dapat berinteraksi dan berintegrasi dengan subsistem yang lain membentuk satu kesatuan.

e. Masukan (*input*)

Masukan atau input merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah bahan yang dimasukkan agar sistem tersebut dapat beroperasi. *Signal input* adalah masukan yang diproses untuk mendapatkan keluaran.

f. Pengolah (*process*)

Pengolahan (*process*) merupakan bagian yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan. Dalam sistem informasi, pengolahan dapat berupa operasi penjumlahan, pengurangan, perkalian, pembagian, pengurutan, atau operasi lainnya yang nantinya akan mengubah masukan berupa data menjadi informasi berguna.

g. Keluaran (*output*)

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

Keluaran (*output*) merupakan hasil dari pemrosesan.

Keluaran dapat berupa informasi sebagai masukan pada sistem lain atau hanya sebagai sisa pembuangan.

h. Saran (*objective*) dan tujuan (*goal*)

Suatu sistem pasti memiliki sasaran (*objective*) atau tujuan (*goal*). Apabila sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Tujuan inilah yang mengarahkan suatu sistem. Tanpa adanya tujuan, sistem menjadi tidak terarah dan terkendali. Tujuan sistem informasi tergantung pada kegiatan yang ditangani.

### 2.2.3 Pengertian Informasi

McFadden dkk didalam Mulyanto (2009) menjelaskan, informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut .

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang menurut Davis didalam Mulyanto (2009).

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

#### 2.2.4 Kualitas Informasi

Kualitas informasi sangat dipengaruhi atau ditentukan oleh tiga hal pokok, yaitu akurasi (*accuracy*), relevansi (*relevancy*), dan tepat waktu (*timeliness*) (Mulyanto, 2009).

a. Akurasi (*accuracy*)

Sebuah informasi harus akurat karena dari sumber hingga penerima kemungkinan banyak terjadi gangguan yang dapat mengubah atau merusak informasi tersebut. Informasi dikatakan akurat apabila informasi tersebut tidak bias atau menyesatkan, bebas dari kesalahan-kesalahan dan harus mencerminkan maksudnya. Ketidakakuratan sebuah informasi dapat terjadi karena sumber informasi (data) mengalami gangguan atau kesengajaan sehingga merusak atau mengubah data-data asli tersebut.

b. Relevansi (*relevancy*)

Informasi dikatakan berkualitas jika relevan bagi pemakainya. Hal ini berarti bahwa informasi tersebut harus bermanfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. Misalnya informasi mengenai kerusakan infrastruktur laboratorium komputer

ditunjukkan kepada rektor universitas. Tetapi akan lebih relevan apabila ditunjukkan kepada penanggung jawab laboratorium.

c. Tepat waktu (*timeliness*)

Informasi yang dihasilkan dari suatu proses pengolahan data, datangnya tidak boleh terlambat (usang). Informasi yang terlambat tidak akan mempunyai landasan dalam pengambilan keputusan. Kesalahan dalam pengambilan keputusan akan berakibat fatal bagi perusahaan.

### **2.2.5 Pengertian Sistem Informasi**

Dalam buku Sistem Informasi Konsep dan Aplikasi didalam Mulyanto (2009), mendefinisikan sistem informasi sebagai sistem yang mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.

### **2.2.6 Konsep Sistem Informasi**

Sistem informasi terdiri dari 5 sumber daya yang dikenal sebagai komponen sistem informasi. Kelima sumber daya tersebut adalah manusia, *hardware*, *software*, data, dan jaringan. Kelima komponen tersebut memainkan peranan yang sangat penting dalam suatu sistem informasi.

Namun, dalam kenyataannya tidak semua sistem informasi mencakup kelima komponen tersebut (Mulyanto, 2009).

a. Sumber Daya Manusia

Manusia mengambil peranan yang penting bagi sistem informasi. Manusia dibutuhkan untuk mengoperasikan sistem informasi. Sumber daya manusia dapat dibedakan menjadi dua kelompok yaitu pengguna akhir dan pakar sistem informasi. Pengguna akhir adalah orang-orang yang menggunakan informasi yang dihasilkan dari sistem informasi, misalnya pelanggan, pemasok, teknisi, mahasiswa, dosen dan orang-orang yang berkepentingan dengan informasi dari sistem informasi tersebut. Sedangkan pakar sistem informasi adalah orang-orang yang mengembangkan dan mengoperasikan sistem informasi, misalnya sistem analis, *developer*, operator sistem, dan staf administrasi lainnya.

b. Sumber Daya *Hardware*

Sumber daya *hardware* adalah semua peralatan yang digunakan dalam pemrosesan informasi. Sumber daya *hardware* tidak hanya sebatas komputer saja, melainkan semua media data seperti lembaran kertas dan *disk* magnetik atau optikal.

c. Sumber Daya *Software*

Sumber daya *software* adalah semua rangkaian perintah (instruksi) yang digunakan untuk memproses informasi. Sumber daya *software* tidak hanya berupa program saja, tetapi juga berupa prosedur. Program merupakan sekumpulan instruksi untuk pemrosesan informasi. Sedangkan prosedur adalah sekumpulan aturan yang digunakan untuk mewujudkan aturan yang digunakan untuk mewujudkan pemrosesan informasi dan mengoperasikan perintah bagi orang-orang yang akan menggunakan informasi.

d. Sumber Daya Data

Sumber daya data bukan hanya sekedar bahan baku untuk memasukkan sebuah sistem informasi, melainkan sebagai dasar membentuk sumber daya organisasi. Seperti yang dijelaskan sebelumnya data dapat berbentuk teks, gambar, audio, maupun video.

e. Sumber Daya Jaringan

Sumber daya jaringan merupakan media komunikasi yang menghubungkan komputer, pemroses komunikasi, dan peralatan lainnya, serta dikendalikan melalui *software* komunikasi. Sumber daya jaringan dapat berupa media komunikasi seperti kabel, satelit, seluler dan dukungan jaringan seperti modem, *software* pengendali, serta prosesor antar jaringan. Kelima komponen tersebut digunakan oleh

sistem informasi untuk menjalankan aktivitas *input*, pemrosesan, *output*, penyimpanan, dan pengendalian yang mengubah sumber daya data menjadi produk informasi.

### **2.2.7 Sistem Klaim Penggantian Biaya**

Sistem adalah kumpulan/grup dari sub sistem/bagian/komponen apa pun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu (Susanto, 2013). *Reimburse* adalah kegiatan untuk melakukan restorasi atau pembayaran yang setara untuk mengganti biaya perjalanan (Oxford Learner's Pocket English Dictionary, 2008).

Dapat disimpulkan sistem *reimbursement* adalah kumpulan/grup dari sub sistem/bagian/komponen yang mempunyai tujuan untuk melakukan restorasi atau pembayaran yang setara untuk mengganti biaya perjalanan yang dikeluarkan oleh karyawan perusahaan.

### **2.2.8 Metode Pengumpulan Data**

#### **2.2.8.1 Observasi**

Proses pencatatan pola perilaku subyek (orang), obyek (benda), atau kejadian yang sistematis tanpa adanya pertanyaan atau komunikasi dengan individu-individu yang diteliti (Sangadji & Sopiah, 2010).

### **2.2.8.2 Wawancara**

Wawancara merupakan teknik pengumpulan data dalam metode *survey* yang menggunakan pertanyaan secara lisan kepada subyek penelitian. Teknik wawancara dilakukan jika peneliti memerlukan komunikasi atau hubungan dengan responden. Teknik wawancara dapat dilakukan dengan 2 cara yaitu: melalui tatap muka atau melalui telepon (Sangadji & Sopiah, 2010).

### **2.2.8.3 Studi Pustaka**

Studi pustaka, yaitu kajian teoritis, referensi serta literature ilmiah lainnya yang berkaitan dengan budaya, nilai dan norma yang berkembang pada ilmu yang diteliti (Sugiyono, 2012).

## **2.2.9 Rapid Application Development**

*Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak (Sukamto & Shalahuddin, 2013).

Jika kebutuhan perangkat lunak dipahami dengan baik dan lingkup perangkat lunak dibatasi dengan baik sehingga tim dapat menyelesaikan pembuatan perangkat lunak dengan waktu yang pendek. Model RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa komponen masing-masing tim pengerjaan dapat dilakukan secara paralel (Sukanto & Shalahuddin, 2013).

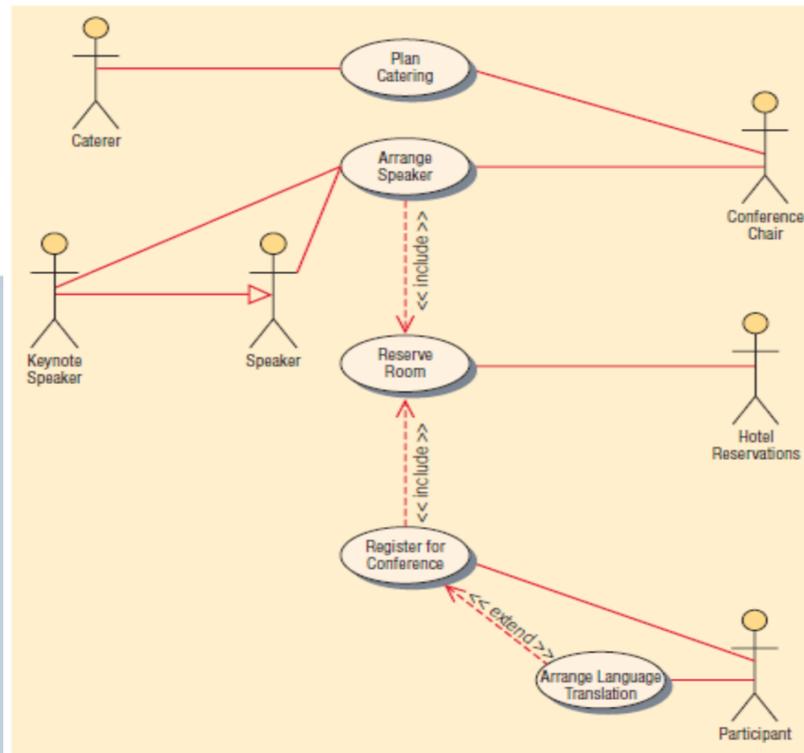
Model RAD cocok diterapkan apabila memenuhi kriteria proyek sebagai berikut (Sukanto & Shalahuddin, 2013):

- a. Anggota tim sudah berpengalaman mengembangkan perangkat lunak sejenis;
- b. Pengembang sudah memiliki komponen-komponen sistem yang bisa digunakan kembali dalam proyek tersebut.

#### 2.2.10 Diagram dan Notasi UML

##### a. *Use Case Diagram*

Merupakan penggambaran dari apa yang dilakukan sistem dengan merefleksikan sistem dari sudut pandang pengguna. *Use-case* terdiri dari aktor dan simbol *use-case* yang dihubungkan melalui sebuah garis (Kendall & Kendall, 2013). Pada gambar 2.2 menggambarkan ilustrasi *use case diagram*.



**Gambar 2.2 Contoh Use Case Diagram**

**Sumber: (Kendall & Kendall, 2013)**

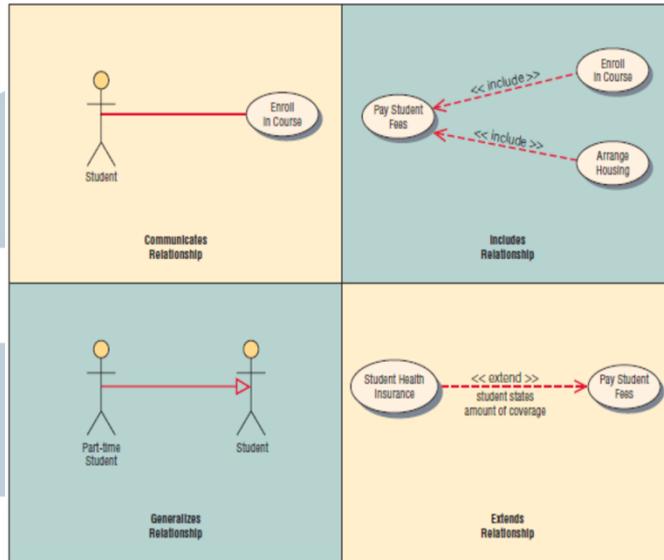
Aktor merupakan seseorang yang menggunakan sistem atau berinteraksi dengan sistem, contohnya seperti pekerja toko. Akan tetapi pekerja toko tersebut bisa juga menjadi seorang pembeli pada toko tersebut. Di dalam *use case diagram*, seseorang tersebut mempunyai dua simbol yang berbeda. Hal ini dikarenakan seseorang tersebut mempunyai peran yang berbeda di dalam sistem. Aktor dapat berinteraksi dengan satu atau lebih *use case* dan *use case* tersebut mungkin melibatkan aktor satu atau lebih.

Di dalam *use case* terdapat empat (4) *relationship* yang menghubungkan antar simbol di dalam *use case* seperti ilustrasi pada gambar 2.3, yakni:

- 1) *Communicates*, berguna untuk menghubungkan aktor ke sebuah simbol *use case*.
- 2) *Includes* atau *Uses*, merupakan *relation* yang menggambarkan dimana sebuah simbol *use case* memiliki satu atau lebih simbol *use case* lain.
- 3) *Extends*, merupakan *relation* yang menggambarkan sebuah simbol *use case* menangani variasi atau pengecualian.
- 4) *Generalize*, digunakan apabila terdapat simbol yang memiliki ciri atau tipikal yang sama seperti dua (2) aktor atau dua (2) simbol *use case*.

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



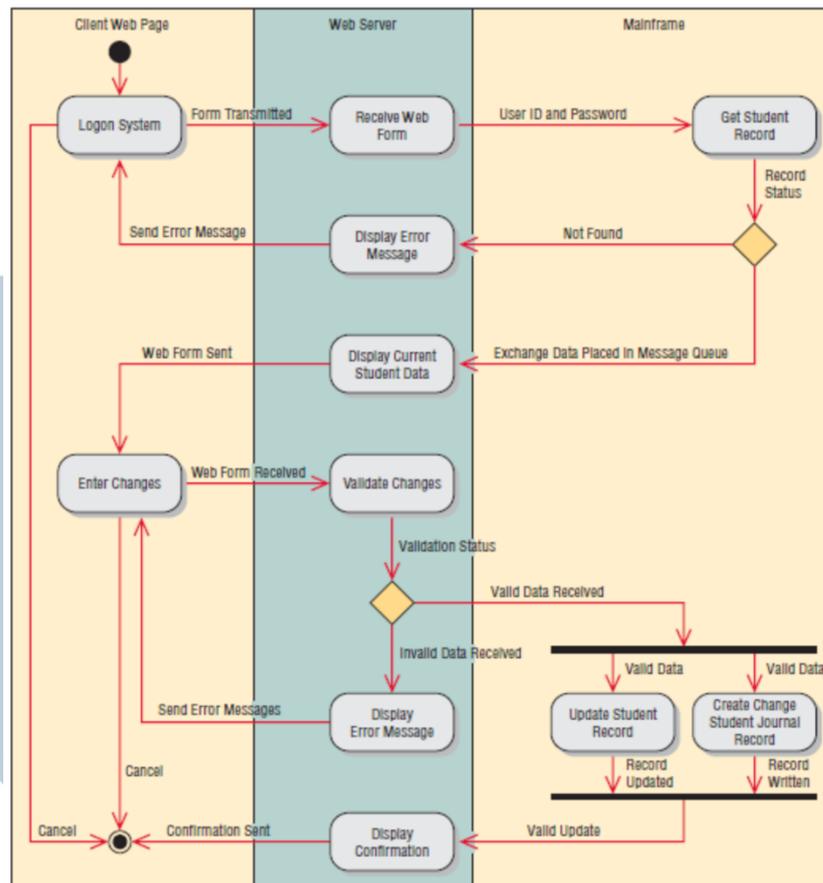
**Gambar 2.3 Use Case Relation**

**Sumber: (Kendall & Kendall, 2013)**

b. *Activity Diagram*

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 di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Sukamto & Shalahuddin, 2013). Pada gambar 2.4 dapat dilihat contoh *activity diagram*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



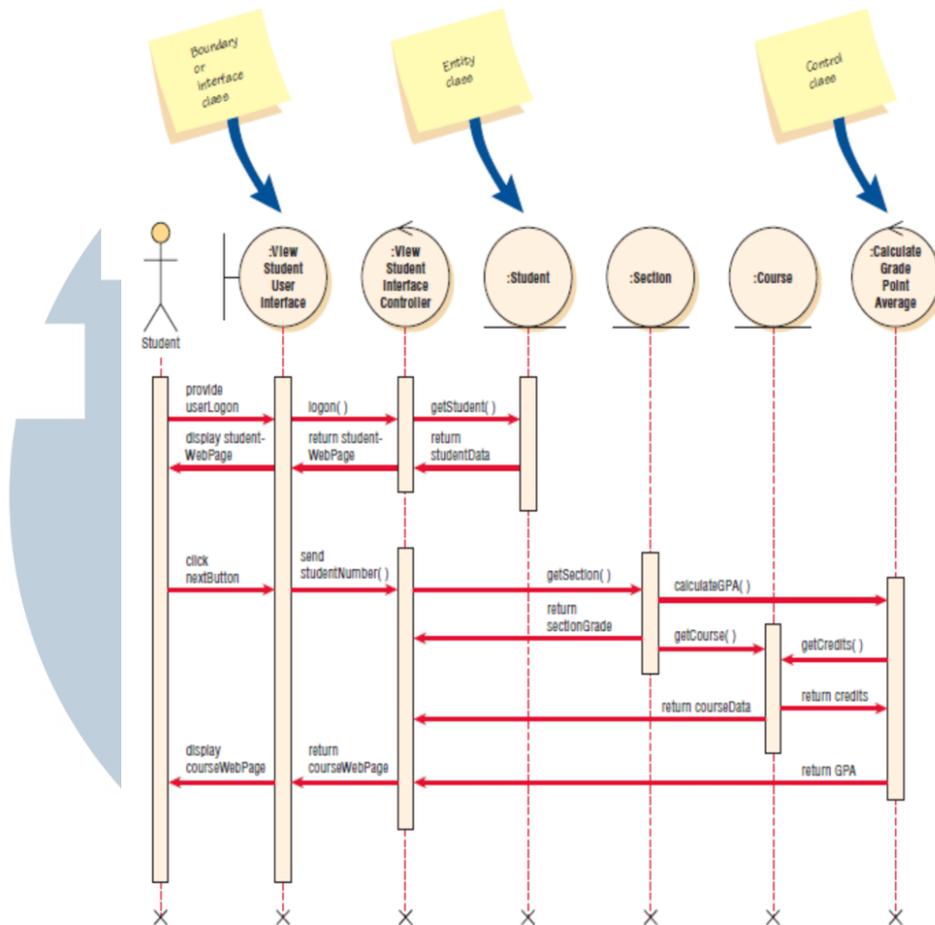
**Gambar 2.4 Contoh Activity Diagram**

(Kendall & Kendall, 2013)

c. *Sequence Diagram*

Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



**Gambar 2.5 Contoh Sequence Diagram**

**Sumber:** (Kendall & Kendall, 2013)

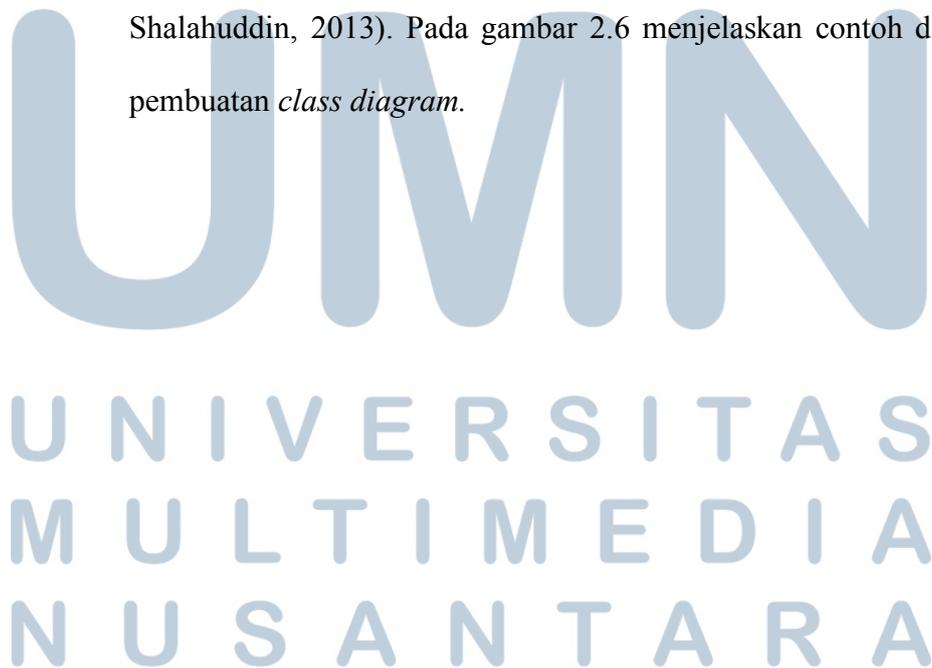
Pada gambar 2.5 menjelaskan contoh *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang di instansi menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

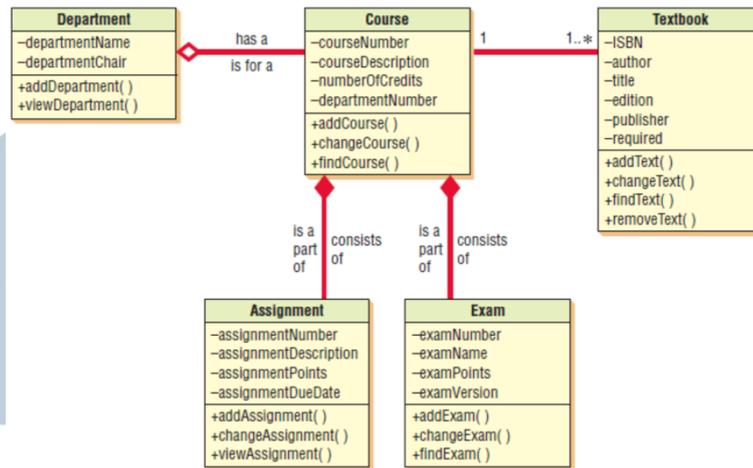
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Banyaknya diagram *sequence* yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang dimiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak (Sukamto & Shalahuddin, 2013).

d. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Sukamto & Shalahuddin, 2013). Pada gambar 2.6 menjelaskan contoh dari pembuatan *class diagram*.





**Gambar 2.6 Contoh Class Diagram**

**Sumber:** (Kendall & Kendall, 2013)

### 2.2.11 Java

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi baik *desktop*, *mobile*, *web* dan lainnya sebagaimana dibuat dengan bahasa pemrograman konvensional yang lain. Java adalah bahasa pemrograman yang berorientasi objek (OOP) yang dapat dijalankan pada berbagai *platform* sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source* (Wahana Komputer, 2009).

### 2.2.12 Javascript

Javascript adalah bahasa pemrograman tingkat tinggi dan dinamis (Flanagan, 2011).

### 2.2.13 PHP

Menurut Arief (Arief, 2011) PHP adalah bahasa *server-side –scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke *browser* dengan format HTML.

### 2.2.14 Laravel

Laravel merupakan salah *framework* PHP yang menawarkan *code modular*. Hal ini dicapai melalui kombinasi *driver* dan sistem *bundle*-nya. *Drivers* memungkinkan untuk dengan mudah mengganti dan memperpanjang *caching, session, database* dan fungsi *authentication*. Dengan menggunakan *bundle* dapat memungkinkan untuk membungkus berbagai kode untuk dapat digunakan kembali. Hal ini sangat bagus dikarenakan kode yang ditulis dapat dibungkus menjadi *bundle* dari *simple library* ke dalam berbagai aplikasi web. *Bundle* website Laravel memungkinkan untuk mencari *bundles* yang telah dibuat oleh komunitas sesuai dengan keinginan. Hal ini menjadikan sumber *third party libraries* menjadi bernilai dan memberikan kemudahan untuk mengembangkan aplikasi *web* (McCool, 2012).

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 2.2.15 Android SDK

Android SDK adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, Google Inc. mengakuisi Android Inc. yang mengembangkan *software* untuk ponsel yang berada di Palo Alto, California Amerika Serikat. (Nazruddin, 2015)

### 2.2.16 Database

Menurut Winarno dkk (Winarno, 2013) mendefinisikan bahwa “Basis data (database) secara bahasa memiliki arti tempat atau wadah sebuah data. Secara istilah database yaitu kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diambil datanya atau diolah menggunakan suatu program.

### 2.2.17 MySQL

MySQL adalah *software* atau program *Database Server*. Sedangkan *SQL* adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database server* termasuk dalam MySQL itu sendiri. *SQL* juga dipakai dalam *software database server* lain, seperti *SQL Server*, *Oracle*, *PostgreSQL* dan lainnya (Nugroho, 2013).

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 2.2.18 XAMPP

XAMPP merupakan perangkat yang menyediakan beberapa perangkat lunak kedalam satu paket. Perangkat lunak tersebut seperti *Apache* untuk menjalankan *web server*, *MySQL* sebagai basis data, *PHP* sebagai *server side scripting*, *PERL*, *FTP Server*, *PHP MyAdmin* dan berbagai perangkat lainnya (Suprianto, 2008).

### 2.2.19 Android Studio

Android Studio adalah IDE untuk android yang diumumkan di bulan Mei 2013 pada acara pengembang Google I/O dan diharapkan sebagai alternatif Eclipse. Android Studio memberikan kemudahan akses pratinjau dengan versi terbaru 0.0.5. Pada saat ini Android Studio belum secara penuh *end to end* aplikasi pengembangan Android, akan tetapi secara berkala akan diperbaharui per bulannya. Android Studio berbasis Java IDE yang disebut IntelliJ yang merupakan produk dari JetBrains seperti RedMine, PyCharm, PhpStorm, WebStorm atau AppCode. Meskipun Android Studio adalah merek baru IDE, hal penting untuk dicatat untuk kemampuan IDE dari Eclipse diterapkan di Android Studio. *Tools* di dalam Android Studio sangat mirip dengan Eclipse seperti *shortcut*, *designer*, dan *code editors*. Android Studio dapat mengekspor *APK*, *view logcat* dan *code virtual* sama seperti halnya dengan Eclipse (Wolfson, 2013).

### 2.2.20 Black Box Testing

*Black-box testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses *testing* di bagian luar. Jenis *testing* ini memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Sebagai contoh, jika terdapat sebuah perangkat lunak yang merupakan sebuah sistem informasi inventori di sebuah perusahaan. Pada jenis *Black-Box Testing*, perangkat lunak tersebut akan dieksekusi kemudian berusaha di tes apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar *listing* programnya (Wicaksono, 2017)

