

BAB II TINJAU PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang relevan dilakukan peneliti sebagai berikut :

Penelitian yang dilakukan oleh [3] yang melakukan proses penyaluran dana beasiswa masih sering terjadi dalam dunia pendidikan salah satunya yaitu kurang tepat sasaran dalam proses penerimaannya. Terdapat beberapa kriteria yang harus dipenuhi oleh siswa sekolah dalam menentukan penerima beasiswa, antara lain keadaan rumah tempat tinggal, fasilitas air dan listrik, penghasilan orang tua serta kendaraan yang digunakan untuk berangkat ke sekolah. Tujuan penelitian ini adalah membantu Dinas Sosial dalam menentukan siswa yang layak menerima beasiswa tidak mampu. Dalam proses pengambilan keputusan tersebut digunakan metode *Multi Objective Optimization on the Basis of Ratio*. Berdasarkan hasil perhitungan, maka diperoleh 6 (enam) data siswa sebagai penerima beasiswa dengan range nilai antara 25 s/d 27 yang merupakan nilai tertinggi dari setiap kriteria.

Penelitian yang dilakukan oleh [5] menggunakan metode *Extreme Programming* (XP) di mana metode ini adalah metode pengembangan perangkat lunak yang cepat, efisien, tingkat risiko rendah, fleksibel dalam penggunaan, diprediksi dengan jelas dan terperinci. Dan metode *Extreme Programming* (XP) diterapkan dalam pembuatan aplikasi. Dengan pembuatan aplikasi mengenai layanan kendaraan operasional, tentu saja bertujuan untuk mengubah semua masalah menjadi keunggulan yaitu efisiensi dan efektivitas. Dalam aplikasi hanya perlu mengkonfirmasi peminjaman dan pengembalian dan pencatatan pemeliharaan dan pengelolaan pajak kendaraan operasional yang telah dimasukkan oleh sistem. Dimana dalam alurnya ketika memasukkan semua data yang diinginkan dapat dikelola secara singkat, cepat dan tepat.

Penelitian yang dilakukan oleh [6] menunjukkan pengalaman menggunakan *Extreme Programming* (XP) untuk proyek tertentu. XP mempraktikkan Pengembangan Berbasis Tes, pemrograman berpasangan, interaksi singkat, kepemilikan kode tim, dan tes penerimaan. Makalah ini menunjukkan kesesuaian penggunaan metode pengembangan perangkat lunak tangkas untuk aplikasi konkret yang dirancang untuk pengguna dengan gangguan tertentu. Metode tangkas

ini lebih menyukai reaksi cepat terhadap perubahan sebelum penyelesaian rencana, yang terbukti penting dalam kasus perangkat lunak yang dikembangkan. Ada banyak perubahan dan persyaratan baru selama pengembangan, dan solusi mereka lebih penting untuk kualitas output daripada mengikuti jadwal waktu pengembangan.

Penelitian oleh [7] yang melakukan pengolahan data yang terdapat di Politeknik Negeri Sriwijaya khususnya pada Jurusan Manajemen Informatika masih sederhana karena menggunakan aplikasi ms.excel yang tentu saja belum optimal. Sehingga dibutuhkan perubahan metode dari informasi kertas ke bentuk digital Perangkat lunak komputer dapat dijadikan alternatif untuk memudahkan proses kompensasi. Kegiatan yang meliputi pengolahan data memasukan data kompensasi yang akan di proses untuk menginformasikan berapa banyak kompensasi kepada mahasiswa. Oleh karena itu, penulis membuat sistem informasi yang dapat digunakan oleh Jurusan Manajemen Informatika untuk menginformasikan kompensasi dengan Menggunakan Metode *Extreme Programming* (XP) agar para mahasiswa dapat mengetahui banyaknya kompensasi dan menjadi disiplin.

Penelitian yang dilakukan oleh [8]. Uang Kuliah Tunggal (UKT) adalah biaya yang dikenakan kepada setiap mahasiswa untuk digunakan dalam proses pembelajaran, yang diberlakukan sejak tahun 2013. Universitas Negeri Gorontalo (UNG) pada awalnya menggunakan metode wawancara dengan mahasiswa untuk menentukan UKT. Namun kendalanya adalah jika banyak mahasiswa yang akan diwawancara akan berpengaruh pada ketelitian akibat faktor kelelahan pewawancara. Selanjutnya UNG menentukan besaran UKT mahasiswa didasarkan pada jumlah penghasilan orang tua. Pada penelitian ini telah dirancang Sistem Pendukung Keputusan (SPK) berbasis logika fuzzy untuk menentukan besaran UKT mahasiswa. Terdapat tiga input sistem fuzzy, yaitu penghasilan orang tua (empat level), jumlah tanggungan (tiga level) dan jumlah rekening air, listrik, internet (tiga level) dan satu output yaitu besaran UKT mahasiswa (delapan level). Aturan JIKA-MAKA sejumlah 36 aturan. Digunakan tiga input (tidak hanya penghasilan orang tua), sehingga hasil penentuan UKT lebih adil. Proses defuzifikasi menggunakan centroid. Dalam simulasi, data uji diambil sebanyak 25 mahasiswa angkatan 2020. Hasil simulasi menunjukkan tingkat akurasi model fuzzy yang dirancang sebesar 76%, yang artinya terdapat 19 UKT mahasiswa yang tepat sama dengan UKT sebenarnya.

Dari beberapa penelitian yang telah dilakukan, maka perbedaan penelitian yang akan saya lakukan adalah sistem ini menggunakan kriteria yang berbeda, yakni jumlah penghasilan orang tua, *assets*, prodi, jumlah tanggungan, jarak dari rumah ke kampus, daya listrik, sumber air, jenis pekerjaan ayah, dan jenis pekerjaan ibu. Selain itu, kelebihan dari sistem ini berdasarkan fitur adalah adanya tambahan fitur pencarian, kelola data peninjauan, dan perhitungan SPK menggunakan metode MOORA. Sistem dibuat dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

2.2 Landasan Teori

2.2.1 SISTEM PENDUKUNG KEPUTUSAN

Menurut jurnal [9] Dalam kehidupan, proses pengambilan keputusan merupakan sesuatu yang akan selalu dihadapi oleh manusia. Keputusan yang diambil biasanya karena ada pertimbangan tertentu atau atas dasar logika, ada alternatif terbaik dari beberapa alternatif yang harus dipilih, dan ada tujuan yang harus dicapai. Keputusan merupakan hasil pemikiran berupa pemilihan satu diantara beberapa alternatif yang dapat digunakan untuk memecahkan masalah yang dihadapi.

Pengambilan keputusan dilakukan pimpinan untuk menyelesaikan masalah yang dihadapi dalam organisasi yang dipimpinya dengan melalui pemilihan satu alternatif pemecahan masalah terbaik dengan berdasarkan pertimbangan-pertimbangan (kriteria) tertentu. Pengambilan keputusan harus dilakukan secara sistematis, mengumpulkan fakta-fakta, kemudian ada penentuan yang matang dari alternatif yang dihadapi, dan selanjutnya mengambil tindakan yang menurut perhitungan merupakan tindakan yang paling tepat.

Kegiatan *Support System* merupakan bagian dari *Management Support System* yang terdiri dari :

1. *Decision Support Systems* (DSS).
2. *Group Support Systems* (GSS), termasuk *Group DSS* (GDSS).
3. *Executive Information Systems* (EIS).
4. *Expert Systems* (ES).
5. *Artificial Neural Networks* (ANN).
6. *Hybrid Support Systems*.

Pembuat keputusan sering kali dihadapkan pada kerumitan dan lingkup pengambilan keputusan dengan data yang begitu banyak. Oleh karena itu, para pembuat keputusan harus mempertimbangkan resiko

manfaat/biaya, dan dihadapkan pada suatu keharusan mengandalkan seperangkat sistem yang mampu memecahkan masalah secara objektif berdasarkan kriteria ataupun pertimbangan yang telah diberikan sebelumnya, sistem ini kemudian disebut Sistem Pendukung Keputusan (SPK).

2.2.2 Metode MOORA

Menurut jurnal [10] *Multi-Objective Optimization by Ratio Analysis* (MOORA) adalah salah satu metode yang dapat digunakan untuk membantu proses pengambilan keputusan dalam SPK. Metode MOORA dikembangkan pertama kali oleh Brauers yang menerapkannya dalam proses pengambilan keputusan dengan multikriteria. Kelebihan metode ini salah satunya adalah fleksibilitas yang tinggi dan tingkat selektifitas yang baik. Hal ini disebabkan MOORA mampu menentukan tujuan dari kriteria yang saling bertolak belakang, dimana kriteria dapat bernilai menguntungkan (*benefit*) atau yang tidak menguntungkan (*cost*). Selain itu, MOORA juga memiliki kemampuan memisahkan unsur subjektif dari suatu proses evaluasi secara mudah ke dalam kriteria bobot keputusan yang memiliki beberapa atribut pengambil keputusan. Langkah-langkah dalam metode MOORA terdiri dari :

1. Membuat *matriks* keputusan. *Matriks* keputusan mewakili semua informasi yang tersedia untuk setiap atribut dalam bentuk matriks. Persamaan *matriks* menunjukkan sebuah *matriks* $X_{m \times n}$, Dimana x_{ij} adalah pengukuran kinerja dari alternatif ke- i pada atribut ke- j , m adalah jumlah alternatif dan n adalah jumlah atribut/kriteria. Selanjutnya, dilakukan perbandingan antara setiap kinerja dari alternatif yang ada pada atribut dengan penyebut yang mewakili semua alternatif dari atribut tersebut menggunakan persamaan.

$$X = \begin{bmatrix} x_{11} & \dots & x_{1i} & \dots & x_{1n} \\ \vdots & & \vdots & & \vdots \\ x_{ji} & \dots & x_{ij} & \dots & x_{jn} \\ \vdots & & \vdots & & \vdots \\ x_{mi} & \dots & x_{mi} & \dots & x_{mn} \end{bmatrix}$$

Gambar 2. 1 Persamaan *Matriks*

Keterangan :

X_{ij} : respon alternatif j pada kriteria i .

i : 1, 2, 3, ..., n adalah inisialisasi urutan kriteria atau atribut.

j : 1, 2, 3, ..., m adalah inisialisasi urutan alternatif

X : matriks keputusan.

2. Normalisasi. Tahapan ini dilakukan untuk mendapatkan nilai element *matriks* yang seragam dengan cara menyatukan setiap element *matriks*. Persamaan digunakan untuk menghitung *matriks* normalisasi.

$$X^*_{ij} = \frac{x_{ij}}{\sqrt{\left[\sum_{j=1}^m x_{ij}^2 \right]}}$$

Gambar 2. 2 *Matriks* Normalisasi

Keterangan :

x_{ij} : *Matriks* alternatif j dengan kriteria i .

i : 1, 2, 3, ..., n adalah inisialisasi urutan kriteria atau atribut.

j : 1, 2, 3, ..., m adalah inisialisasi urutan alternatif.

X^*_{ij} : *Matriks* Normalisasi alternatif j dengan kriteria i .

3. Mengurangi nilai *maximax* dan *minimax*. Pada tahapan ini terdapat dua kondisi yang mungkin terjadi yang masing-masing memiliki perhitungan yang berbeda. Kondisi tersebut yaitu :
- (1) Tidak adanya nilai bobot yang dimiliki atribut atau kriteria di setiap alternatif. Pada situasi ini, maka dilakukan pengurangan nilai maksimum dan minimum pada setiap baris. Hal ini dilakukan untuk mendapatkan *rangking* dari setiap baris. Persamaan digunakan dalam proses ini.

$$y_j^* = \sum_{i=1}^{i=g} x_{ij}^* - \sum_{i=g+1}^{i=n} x_{ij}^*$$

Gambar 2. 3 *maximax* dan *minimax*

Keterangan :

- i : 1,2,3, ..., g adalah kriteria atau atribut *maximized*.
 j : g+1, g+2, g+3, ..., n adalah kriteria atau atribut *minimized*.
 y * j : Matriks normalisasi hasil pengurangan *maximize* dan minimize alternatif j.

- (2) Atribut atau kriteria di setiap alternatif memiliki atau diberikan nilai bobot kepentingan. Pada kondisi ini maka pemberian nilai bobot dilakukan dengan ketentuan nilai bobot jenis kriteria minimum harus lebih kecil dari nilai bobot kriteria maksimum. Koefisien signifikansi diberikan pada atribut yang lebih penting dengan cara melakukan perkalian nilai bobot dengan nilai koefisiensi tersebut. Persamaan digunakan untuk menghitung bagian ini.

$$y_i = \sum_{j=1}^g w_j x_{ij}^* - \sum_{j=g+1}^n w_j x_{ij}^*$$

Gambar 2. 4 Atribut atau kriteria

Keterangan:

- I : 1,2,3, ..., g adalah kriteria atau atribut *maximized*.
 j : g+1, g+2, g+3, ..., n adalah kriteria atau atribut *minimized*
 w_j : nilai bobot alternatif.
 j * y_i : nilai penilaian yang sudah dinormalisasi dari alternatif j terhadap semua atribut.

4. Pemeringkatan atau perangkingan. Total nilai maksimal (atribut *benefit*) dalam sebuah *matriks* keputusan dapat mengakibatkan nilai y_i bernilai *positif* atau *negatif*. Pada tahap ini dilakukan pemeringkatan terhadap nilai y_i , dimana nilai y_i tertinggi menjadi penanda alternatif terbaik, sedangkan Alternatif dengan nilai y_i terendah adalah alternatif terburuk.

2.2.3 UKT (Uang Kuliah Tunggal)

Menurut jurnal [11] Uang Kuliah Tunggal (UKT) merupakan sistem yang diterapkan dalam pembiayaan perkuliahan yang harus ditanggung oleh mahasiswa Perguruan Tinggi Negeri yang berada di bawah Kementerian Riset, Teknologi, dan Pendidikan Tinggi (Kemendiknas). Dalam hal ini, pembiayaan perkuliahan tidak seperti pada saat-saat sebelumnya dimana biaya perkuliahan mahasiswa adalah terdiri dari beberapa macam komponen yang terpisah, misalnya SPP, Uang Praktikum, Iuran Orang Tua Mahasiswa (IOM), Uang Ujian Skripsi, Uang Wisuda, Sumbangan Peningkatan Kualitas Pendidikan (SPKP), dan lain-lain. Melalui penerapan UKT berarti bahwa hanya terdapat satu jenis pungutan biaya perkuliahan yang dikenakan kepada mahasiswa.

Menurut jurnal [12] Uang Kuliah Tunggal (UKT) merupakan sebagian biaya kuliah yang ditanggung setiap mahasiswa berdasarkan kemampuan ekonominya, sebagaimana peraturan pemerintah terhadap semua Perguruan Tinggi Negeri. Ketentuan itu tertuang dalam Peraturan Menteri Pendidikan dan Kebudayaan No. 55 Tahun 2013.

2.2.4 WEBSITE

Menurut jurnal [12] pengertian *website* adalah “keseluruhan halamanhalaman web yang terdapat dari sebuah domain yang mengandung informasi”.

Menurut *website* [13] disebut juga site, situs, situs web atau portal. Merupakan kumpulan halaman *website* yang berhubungan antara satu dengan lainnya, halaman pertama sebuah *website* adalah home page, sedangkan halaman demi halamannya secara mandiri disebut web page, dengan kata lain *website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet diseluruh dunia. *Website* adalah situs yang dapat diakses dan dilihat oleh para pengguna Internet. Pengguna Internet semakin hari semakin bertambah banyak, sehingga hal ini adalah potensi pasar yang berkembang terus.

2.2.5 XAMPP

Menurut jurnal [14] *Xampp* merupakan sebuah perangkat lunak gratis sehingga bebas digunakan. *Xampp* berfungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari *Apache HTTP Server*, *MySQL database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.

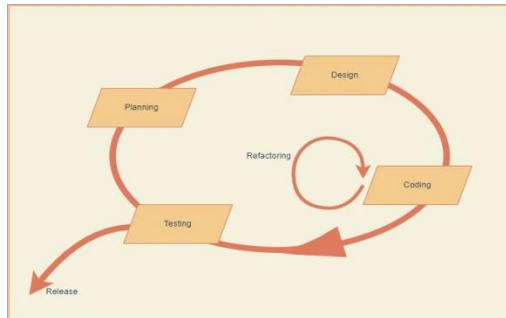
2.2.6 REKAYASA PERANGKAT LUNAK

Menurut jurnal [15] *Rekayasa perangkat lunak (software engineering)* merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

2.2.7 Metode XP

Dalam pengembangan sistem ini penulis akan menggunakan metode pengembangan *Extreme Programming (XP)*. Pembangunan sistem secara keseluruhan dilakukan melalui beberapa tahapan/langkah. Metode *Extreme Programming (XP)* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan-perubahan *requirement* yang sangat cepat[5].

Menurut jurnal [16] metode *Extreme Programming (XP)* adalah salah satu metodologi dalam rekayasa perangkat lunak dan juga merupakan bagian dari metodologi pengembangan perangkat lunak agile. Secara umum *Extreme Programming (XP)* dapat sebagai sebuah pendekatan pengembangan perangkat lunak yang mencoba meningkatkan efisiensi dan fleksibilitas dari sebuah proyek pengembangan perangkat lunak dengan mengkombinasikan berbagai ide simple/ sederhana tanpa mengurangi kualitas *software* yang akan dibangun.



Gambar 2. 5 Tahapan Metode Extreme Programming

Gambar 2.5 menunjukkan aktivitas perencanaan dimulai dengan membentuk alur pengguna dengan tahapan :

1. *Planning*

Pada tahap ini penulis akan mengumpulkan kebutuhan informasi dan data sebanyak-banyaknya yang diperlukan dalam pembuatan aplikasi.

2. *Design*

Pada tahap ini akan mengidentifikasi kebutuhan data ke dalam perancangan sistem, untuk memberikan Gambaran perancangan sistem kepada pengguna/user.

3. *Coding*

Pada tahap ini akan membuat aplikasi dari rancangan sistem yang diajukan pada tahap *Design*, kemudian akan dilakukan tahapan selanjutnya yaitu pengujian, akan tetapi dikarenakan disini membuat aplikasi menggunakan metode *Extreme Programming* maka setiap menu yang dibuat tidak langsung pada keseluruhan akan tetapi setiap menu yang dibuat akan langsung diujikan

4. *Testing*

Pada tahap ini akan dilakukan pengujian, dimana sesuai dengan konsep pada metode *Extreme Programming*, setiap pengujian dilakukan setiap pembuatan menu. sebagaimana yang dicontohkan ketika pertama kali membuat rancangan menu *login* maka pada saat selesai akan langsung diujikan, apabila validasi menu *login* bermasalah maka akan dilakukan perbaikan pada saat hari itu juga, akan tetapi apabila validasi *login* berhasil maka akan dilanjutkan untuk pembuatan menu selanjutnya.

2.2.8 ERD

A. Pengertian ERD

ERD (Entity Relationship Diagram) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (Entity) dan hubungan (Relationship), yang ada pada Entity berikutnya[17].

Simbol-simbol dalam ERD (Entity Relationship Diagram) adalah sebagai berikut:

- a. Entitas: suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.
- b. Atribut: ciri umum semua atau sebagian besar instansi pada entitas tertentu.
- c. Relasi: hubungan alamiah yang terjadi antara satu atau lebih entitas.
- d. Link: garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

B. Kardinalitas Relasi




- a. Satu ke satu (One to One)
Setiap elemen dari Entitas A berhubungan paling banyak dengan elemen pada Entitas B. Demikian juga sebaliknya setiap elemen B berhubungan paling banyak satu elemen pada Entitas A.
- b. Satu ke banyak (One to Many)
Setiap elemen dari Entitas A berhubungan dengan maksimal banyak elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan paling banyak satu elemen di Entitas A.
- c. Banyak ke satu (Many to One)
Setiap elemen dari Entitas A berhubungan paling banyak dengan satu elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan maksimal banyak elemen di entitas A.
- d. Banyak ke banyak (Many to Many)
Setiap elemen dari Entitas A berhubungan maksimal banyak elemen pada Entitas B demikian sebaliknya.

2.2.9 Flowchart

Menurut jurnal [18] *Flowchart* adalah cara penulisan algoritma dengan menggunakan notasi grafis. *Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung. Dengan menggunakan *flowchart* akan memudahkan untuk melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah. *Flowchart* dapat membantu analisis dan programmer untuk memecahkan masalah kedalam segmensegmen yang lebih kecil dan membantu dalam menganalisis alternatif-alternatif lain dalam pengoperasian.







a. Flow Direction Symbols

Tabel 2.1 Flow Direction Symbols

Simbol	Keterangan
	Simbol arus / <i>flow</i> yaitu menyatakan jalannya arus suatu proses.
	Simbol <i>communication link</i> , yaitu menyatakan transmisi data dari satu lokasi ke lokasi lain.
	Simbol <i>connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.




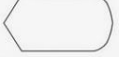
b. *Processing Symbols*

Tabel 2. 2 Processing Symbols

Simbol	Keterangan
	Simbol <i>process</i> , yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh <i>computer</i> .
	Simbol manual, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh <i>computer</i> .
	Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan dengan jawaban ya /tidak.
	Simbol predefined process, yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	Simbol terminal, yaitu menyatakan permulaan atau akhir suatu program.
	Simbol keying operation, menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.

c. *Input-output Symbols*





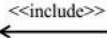
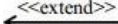
Tabel 2. 3 Input-output Symbols

Simbol	Keterangan
	Simbol <i>input/output</i> , menyatakan proses <i>input</i> atau <i>output</i> tanpatergantung jenis peralatannya.
	Simbol <i>punched card</i> , menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
	Simbol document, mencetak keluaran dalam bentuk dokumen (melalui printer).
	Simbol display, mencetak keluaran dalam layar monitor.

2.2.10 Use Case

Menurut jurnal [14] *CVX 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. Secara kasar *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut ini adalah simbol-simbol yang ada pada diagram *use case* :



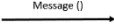



Tabel 2. 4 Use Case Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Merupakan peran orang, sistem yang lain, atau alat ketika berhubungan dengan <i>Use Case</i> .
	<i>Use Case</i>	Abstraksi dari penghubung antara Actor dengan <i>Use Case</i> .
	<i>Association</i>	Abstraksi dari penghubung antara aktor dengan <i>Use Case</i> .
	Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>Use Case</i> .
	<i>Include</i>	Menunjukkan bahwa suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya.
	<i>Extend</i>	Menunjukkan bahwa suatu <i>Use Case</i> merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi terpenuhi.

2.2.11 Sequence Diagram

Menurut jurnal [19] *Sequence Diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu :








Tabel 2. 5 *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>LifeLine</i>	<i>Objek entity</i> , antarmuka yang saling berinteraksi.
	<i>Actor</i>	Menggambarkan user atau pengguna.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktivitas yang terjadi.
	<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
	<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

2.2.12 Clas Diagram

Menurut jurnal [20] Class Diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

Tabel 2. 6 Class Diagram

Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.