



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Tugas Akhir ini dilakukan tidak lepas dari hasil penelitian – penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Adapun hasil – hasil penelitian yang dijadikan perbandingan tidak terlepas dari topik penelitian Tugas Akhir yaitu mengenai aplikasi Sistem Pendukung Keputusan mengenai dosen pembimbing dan *Analytical Hierarchy Process*.

Tim pengembang yang bernama Weda Adistianaya Dewa dan Linda Suvi Rahmawati program studi sistem informasi mahasiswa STIMK PPKIA Pradnya Paramita membuat sebuah penelitian dengan judul “Analisis dan Desain Sistem Pendukung Keputusan Penentuan Dosen Pembimbing Tugas Akhir Menggunakan Metode AHP” dengan 5 (lima) kriteria penentu diantaranya tingkat pendidikan dosen, roadmap penelitian dosen, latar belakang, bidang minat, dan jabatan fungsional. Dapat disimpulkan bahwa dari hasil penelitian, sistem pengambilan keputusan yang dibuat dapat membantu untuk memudahkan pengambilan keputusan yang terkait dengan penentuan dosen pembimbing, sehingga akan didapatkan dosen pembimbing tugas akhir yang paling layak untuk membimbing mahasiswa [1].

Penelitian dengan Metode *Analytical Hierarchy Process* (AHP) juga digunakan oleh Asrul Abdullah dan Menur Wahyu Pangestika program studi Teknik Informatika dari Universitas Muhammadiyah Pontianak dengan judul “Perancangan Sistem Pendukung Keputusan Pemilihan Dosen Pembimbing Skripsi Berdasarkan Minat Mahasiswa dengan Metode AHP (*Analytical Hierarchy Process*) di Universitas Muhammadiyah Pontianak”. Penelitian ini memiliki kriteria yang penentuannya didasarkan pada jumlah minat yang ada di program studi Teknik Informatika UM Pontianak. Terdapat tiga kriteria yang digunakan dalam penelitian ini yaitu RPL, Jaringan, dan Multimedia. Sistem yang dibuat ini dapat memberikan kemudahan bagi ketua program studi Teknik Informatika UM Pontianak dalam memberikan keputusan dosen pembimbing yang sesuai dengan minat mahasiswa. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *web-engineering* dan algoritma Rabin-Karp [3].

Penelitian lain sehubungan dengan Sistem Pendukung Keputusan dalam menentukan dosen pembimbing menggunakan metode AHP dengan judul “Rancangan Sistem Informasi Pengajuan Kuliah Kerja Praktek dan Penentuan. Dosen Pembimbing Dengan Metode AHP” pernah dilakukan oleh Nadiatul Khairiyah dan Agnes Novita dari Institusi Perbanas Fakultas Teknologi Informasi membuat sebuah sistem yang membantu memecahkan masalah dimana dalam proses pemilihan dosen pembimbing kuliah kerja praktek sebelumnya kaprodi hanya bergantung pada data – data yang ada secara manual sehingga mengakibatkan hasil yang tidak akurat apakah dosen yang diajukan berada di bidang ilmu yang sesuai dengan judul laporan kuliah kerja praktek mahasiswa. Kriteria – kriteria yang digunakan dalam sistem pendukung keputusan ini berdasarkan bidang ilmu dosen diantaranya analisis, pemrograman, *database*, dan *networking*. Sistem pendukung keputusan yang dibangun ini membantu mahasiswa dalam melaksanakan kegiatan KKP dan membantu kaprodi dalam mengambil keputusan untuk menentukan dosen pembimbing yang sesuai dengan judul laporan KKP mahasiswa [4].

Penelitian lain mengenai pemilihan SPK dosen pembimbing dengan menggunakan metode selain AHP dibuat oleh Tantri Hari Mukti, Syaad Patmantara, dan Aji Prasetya Wibawa dari Universitas Negeri Malang. SPK yang dibuat menggunakan metode TOPSIS ini dibangun untuk membantu Kelompok Bidang Keahlian (KBK) dan Koordinator Program Studi (Korprodi) dalam memberikan rekomendasi dosen pembimbing sesuai dengan kriteria – kriteria calon dosen pembimbing yang sesuai dengan kondisi dan judul yang diajukan mahasiswa untuk melaksanakan tugas akhir. Kriteria – kriteria yang digunakan dalam sistem ini diantaranya adalah kompetensi dan bidang keahlian dosen, tingkat pendidikan dosen, pangkat dan fungsionalitas dosen, jumlah total mahasiswa yang dibimbing oleh seorang dosen, jadwal bimbingan dosen, durasi dalam setiap kali bimbingan, serta layanan konsultasi. Metode penelitian yang digunakan pada sistem ini adalah metode *Waterfall* berdasarkan referensi dari Sommerville. Sistem ini telah membuktikan bahwa hasil keputusan pemilihan dosen pembimbing telah membantu KBK dan Korprodi dalam menyesuaikan dosen pembimbing dengan mahasiswa dan mendapatkan hasil perhitungan sistem yang sesuai dengan hasil perhitungan secara manual [5].

Penelitian lain yang sehubungan dengan Sistem Pendukung Keputusan menggunakan metode AHP untuk penentuan alokasi dana desa yang dibuat oleh Vasma Vitriani Sianipar, Anjar Wanto, dan M Safii. Jumlah kriteria yang digunakan berjumlah 4 (empat) buah. Penelitian ini menghasilkan sebuah website untuk melakukan pemilihan dengan perhitungan menggunakan metode AHP yang hasilnya telah dibuktikan bahwa perhitungan secara manual dan yang telah dikomputerisasi memiliki hasil yang sama [6].

Penelitian yang sehubungan dengan Sistem Pendukung Keputusan menggunakan metode AHP juga dilakukan oleh D. G. Edi Putra, M. Ramaddan Julianti, dan Siti Maesaroh. Para peneliti ini menggabungkan metode AHP dan TOPSIS sebagai perhitungan untuk mencari petugas AIS terbaik tiap tahun. Peneliti ini juga menggunakan metode SDLC Waterfall sebagai metode pengembangan sistem. Terdapat 10 (sepuluh) alternatif dan 5 (lima) kriteria yang digunakan dalam sistem ini. Penelitian ini memberikan kesimpulan bahwa dengan membuat sebuah website, perhitungan yang tadinya dilakukan secara manual dapat diselesaikan dengan lebih cepat [7].

Tabel 2. 1 Penelitian Terdahulu

| Komponen | Peneliti | | | | |
|--|---|--|---|--|--|
| Judul Penelitian | Wedi Aditrianyara Deva & Linda Sery Rahmawati | Analisis dan Desain Sistem Keputuasan Penentuan Dosen Pembimbing Tugas Akhir Menggunakan Metode AHP | Perancangan Pendukung Pemilihan Dosen Pembimbing Berbasis Sistem Keputuasan Penentuan Dosen Pembimbing Dengan Metode AHP (Analytical Hierarchy Process) di Universitas Muhammadiyah Pontianak | Nadialat Khairiyah & Agnes Novita | Rancangan Sistem Informasi Pengujian Kuliiah Kerja Praktek dan Penentuan Dosen Pembimbing Dengan Metode AHP |
| Metode Penelitian | Identifikasi dan Analisis Kebutuhan, Perancangan sistem untuk pemetaan dosen pembimbing, Perubahan program berdasarkan rancangan sistem yang telah dibuat. | Analisa masalah, Analisa kebutuhan, Mendesain sistem yang akan dibangun, Membuat perangkat lunak, Menguji perangkat lunak, Menganalisa hasil keputuasan yang diperoleh dari perangkat lunak, Membuat laporan dan menyimpulkan hasil penelitian | System Development Life Cycle (SDLC) dengan tahapan Analisis (Perencanaan), Analisis (Analisa), Design (Desain) dan Implementasi (Implementasi) | Waterfall dengan tahap diantaranya adalah Analisis dan definisi permasalahan, Perancangan sistem dan perangkat lunak, Implementasi dan pengujian unit, Integrasi dan pengujian sistem, dan perawatan serta | Pengumpulan data dengan melakukan observasi, interviu dan studi pustaka. Dengan menggunakan SDLC sebagai metode pengembangan sistem. Selain itu, penduan in juga menggunakan AHP dengan TOPSIS. |
| Tahun Penelitian | 2018 | 2018 | 2021 | 2018 | 2020 |
| Basis | Sistem pengambilan keputuasan yang dibuat dapat membantu untuk memudahkan pengambilan keputuasan yang terkait dengan penentuan dosen pembimbing. sehingga akan didapatkan dosen pembimbing tugas akhir yang paling banyak untuk membimbing mahasiswa. | Menentukan kemabahan bagi lenua program studi Teknik Informatika UM Pontianak dalam menentukan keputuasan dosen pembimbing yang sesuai dengan minat mahasiswa. | Membantu mahasiswa dalam melakukan kegiatan KKP dan membantu keputuasan untuk menentukan dosen pembimbing yang sesuai dengan jumlah laporan KKP mahasiswa. | Hasil keputuasan pemilihan dosen pembimbing telah membantu NKR dan Kaprodi dalam menyelesaikan dosen pembimbing mahasiswa dan memperhatikan hasil penelitian sistem yang sesuai dengan hasil penelitian secara manual. | Metode AHP ternyata dapat digunakan dan diterapkan dengan baik dalam SPK untuk kasus penentuan alodasi Desa Nagori, Siboga. Hasil akhir yang diperoleh menggunakan penelitian manual atau komputer menunjukkan hasil yang memuaskan. |
| D.G. Edi Putra, M. Romadhen Juhanti, Siti Masrah | Decision Support System for the INADMA AIS Officer of the Year Award using AHP-TOPSIS Method | Decision Support System for Demanation of Village Fund Allocation Using AHP Method | Decision Support System for Demanation of Village Fund Allocation Using AHP Method | Decision Support System for Demanation of Village Fund Allocation Using AHP Method | Pengumpulan data dengan melakukan observasi, interviu dan studi pustaka. Dengan menggunakan SDLC sebagai metode pengembangan sistem. Selain itu, penduan in juga menggunakan AHP dengan TOPSIS. |
| D.G. Edi Putra, M. Romadhen Juhanti, Siti Masrah | Decision Support System for the INADMA AIS Officer of the Year Award using AHP-TOPSIS Method | Decision Support System for Demanation of Village Fund Allocation Using AHP Method | Decision Support System for Demanation of Village Fund Allocation Using AHP Method | Decision Support System for Demanation of Village Fund Allocation Using AHP Method | Pengumpulan data dengan melakukan observasi, interviu dan studi pustaka. Dengan menggunakan SDLC sebagai metode pengembangan sistem. Selain itu, penduan in juga menggunakan AHP dengan TOPSIS. |

Berdasarkan penelitian sebelumnya mengenai pemilihan dosen pembimbing, terdapat perbedaan dengan aplikasi yang akan dikembangkan oleh peneliti pada penelitian ini, yaitu, sistem – sistem yang dikembangkan sebelumnya ditujukan untuk membantu program studi dalam memilih dosen pembimbing yang sesuai dengan tema yang diangkat mahasiswa yang akan melaksanakan tugas akhir / skripsi, beberapa sistem dari penelitian sebelumnya juga menggunakan gabungan model perhitungan dalam SPK. Sedangkan, aplikasi yang akan di peneliti kembangkan merupakan aplikasi sistem pendukung keputusan berbasis website dengan menggunakan metode perhitungan AHP yang ditujukan untuk membantu mahasiswa teknik informatika dalam mendukung pilihannya dalam memilih dosen pembimbing. Dengan menggunakan metode pengembangan sistem *Waterfall* menurut Pressman.

2.2 Landasan Teori

2.2.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan dapat diartikan sebagai sistem yang mampu memecahkan masalah baik pada masalah yang terstruktur maupun tak terstruktur serta dapat membantu dalam pengambilan keputusan ketika seseorang atau suatu organisasi telah mengetahui pasti bagaimana keputusan seharusnya diambil.

Menurut Turban (2001) Sistem Pendukung Keputusan didefinisikan sebagai suatu informasi yang dapat menghasilkan berbagai alternatif keputusan dengan menggunakan data dan model untuk membantu manajemen dalam menangani berbagai permasalahan terstruktur maupun tidak terstruktur [8].

Kusrini (2007) juga mendefinisikan bahwa Sistem Pendukung Keputusan merupakan sistem yang digunakan untuk membantu pengambilan keputusan dimana telah menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem informasi yang interaktif ini sangat membantu dalam pengambilan keputusan dalam suatu kondisi yang semi terstruktur maupun tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat [9].

Terdapat proses yang terdiri dari tiga fase menurut Simon dalam (Luzaenah, 2008) model yang menggambarkan proses pengambilan keputusan, yaitu sebagai berikut [10] :

1. *Intelligence*

Merupakan tahap yang memetakan tingkat problematika dan mampu mengenali permasalahan yang terjadi melalui proses penelusuran dan pendeteksian dimana data masukan yang diperoleh akan diuji serta diproses untuk mendukung proses identifikasi masalah.

2. *Design*

Design atau rancangan merupakan tahap yang dimulai dengan pengembangan dan pencarian solusi alternatif untuk tindakan sangat mungkin untuk dilakukan. Tahap ini meliputi proses verifikasi dan validasi guna mengetahui masalah, menurunkan, dan menguji kelayakan solusi.

3. *Choice*

Tahap ini berfungsi sebagai proses untuk memilih berbagai solusi dari alternatif yang dapat dipilih. Hasil dari pemilihan tersebut nantinya akan diimplementasikan ke dalam proses pengambilan keputusan.

Menurut Turban & Aronson (2005) sistem pendukung keputusan memiliki beberapa unsur karakteristik yang merupakan syarat utama bagi tercapainya tujuan yang mendasai pengembangan suatu sistem [10].

2.2.2 Pengertian Metode *Analytical Hierarchy Process* (AHP)

Menurut Casym dan Oktiara (2020) *Analytical Hierarchy Process* atau AHP merupakan teori umum struktur hierarki tingkat berganda yang dapat menghasilkan skala rasio dari perbandingan berpasangan yang berbentuk diskrit maupun kontinu. Dan dapat memberikan manfaat dari mengambil keputusan untuk memilih alternatif terbaik berdasarkan kriteria tertentu [11].

Taylor (2014) berpendapat bahwa AHP dapat memilih alternatif terbaik dari beberapa kriteria berdasarkan pada sejauh mana tiap – tiap alternatif memenuhi kriteria pengambil keputusan melalui pengembangan satu nilai numerik untuk memeringkatkan setiap alternatif keputusan [12].

Terdapat beberapa prinsip dasar dari AHP, yaitu sebagai berikut [13] :

1. Permasalahan yang akan diselesaikan dipecah menjadi unsur – unsur dengan menyusun hierarki.

- Membentuk tabel perbandingan berpasangan (*pairwise comparison*) untuk melakukan penilaian kriteria dan alternatif.

2.2.2.1 Penerapan Metode *Analytical Hierarchy Process* (AHP)

Pada dasarnya, penerapan metode *Analytical Hierarchy Process* (AHP) meliputi :

- Membuat Hierarki

Sistem yang kompleks akan dipecah menjadi elemen – elemen pendukung dan disusun secara hierarki.



Gambar 2. 1 Struktur Hirarki AHP

- Penilaian Kriteria dan Alternatif

Penilaian ini dilakukan dengan cara perbandingan berpasangan. Untuk berbagai persoalan yang ada, skala 1 sampai 9 menurut skala Saaty [14] adalah skala terbaik dalam mengekspresikan pendapat.

Tabel 2. 2 Skala Perbandingan Berpasangan

| Intensitas Kepentingan | Keterangan |
|------------------------|--|
| 1 | Kedua elemen sama pentingnya |
| 3 | Elemen yang satu sedikit lebih penting daripada yang lainnya |
| 5 | Elemen yang satu lebih penting daripada yang lainnya |
| 7 | Satu elemen jelas lebih mutlak penting daripada elemen lainnya |
| 9 | Satu elemen mutlak penting daripada elemen lainnya |
| 2,4,6,8 | Nilai – nilai antara dua nilai pertimbangan – pertimbangan yang berdekatan |
| Kebalikan | Jika aktivitas x mendapat satu angka dibandingkan dengan aktivitas y, maka y memiliki nilai kebalikannya dibandingkan dengan x |

3. Penentuan Prioritas
Perbandingan berpasangan perlu dilakukan untuk setiap kriteria dan alternatif yang ada. Untuk menghasilkan bobot dan prioritas, nilai perbandingan relatif dari seluruh alternatif dan kriteria dapat disesuaikan dengan penilaian yang telah ditentukan.
4. Konsistensi Logis
Konsistensi logis memiliki dua makna. Pertama, objek – objek yang serupa dapat dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu.

2.2.2.2 Prosedur Penyelesaian Metode *Analytical Hierarchy Process* (AHP)

Dalam perhitungannya, menurut Irawan, dkk (2019) metode *Analytical Hierarchy Process* memiliki beberapa prosedur penyelesaian, antara lain sebagai berikut [15] :

1. Mendefinisikan masalah serta menentukan solusi yang diinginkan, lalu menyusun hierarki dari masalah – masalah yang dihadapi. Menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas dengan penyusunan hierarki.
2. Menentukan prioritas elemen
 - a. Membandingkan elemen secara pasangan untuk membuat perbandingan pasangan, membandingkan elemen secara berpasangan sesuai kriteria yang ada.
 - b. Menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen lainnya untuk mengisi matriks perbandingan.
3. Memperoleh keseluruhan prioritas dengan mempertimbangkan perbandingan berpasangan. Hal – hal yang perlu dilakukan antara lain :
 - a. Menjumlahkan nilai – nilai dari setiap kolom kriteria dan alternatif pada matriks.
 - b. Membagi setiap nilai dari kolom kriteria serta alternatif dengan total kolom kriteria dan alternatif yang bersangkutan untuk memperoleh normalisasi matriks.

- c. Menjumlahkan nilai – nilai dari setiap baris kriteria serta alternatif dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata – rata.
4. Mengetahui seberapa baik konsistensi yang ada untuk menghindari keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal – hal yang perlu dilakukan antara lain :
- Kalikan setiap nilai pada kolom pertama kriteria serta alternatif dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.
 - Jumlahkan setiap baris pada kriteria dan alternatif.
 - Hasil dari penjumlahan baris setiap kriteria dan alternatif dibagi dengan elemen prioritas relatif yang bersangkutan.
 - Jumlahkan hasil bagi yang telah dilakukan terhadap perbandingan kriteria maupun alternatif dengan banyaknya elemen yang ada, hasilnya disebut λ (lamda) maks.
5. Menghitung *Consistency Index* (CI) dengan rumus :
- $$CI = \frac{(\lambda \text{ maks} - n)}{(n-1)}$$
6. Menghitung *Consistency Ratio* / Rasio Konsistensi dengan rumus :
- $$CR = \frac{CI}{IR}$$
7. Memeriksa konsistensi hierarki, jika nilainya lebih besar dari 10% maka penilaian data *judgement* harus diulang untuk diperbaiki. Namun jika rasio konsistensinya $\frac{(CI)}{(IR)}$ kurang atau sama dengan 0,1, maka hasil perhitungan dapat dinyatakan benar.

Keterangan :

- n = Banyaknya elemen
 CR = *Consistency Ratio*
 CI = *Consistency Index*
 IR = *Index Random Consistency*

2.2.3 MySQL (*My Structure Query Language*)

MySQL (*My Structure Query Language*) merupakan server yang melayani *database*. Pemrograman khusus yang disebut *query* (perintah) SQL dipelajari untuk membuat dan mengelola *database*. *Database* dibutuhkan jika ingin menginput data dari *user* menggunakan *form* HTML untuk kemudian diolah PHP agar dapat disimpan ke dalam *database* MySQL. MySQL bersifat *open source*, yang artinya memungkinkan setiap orang untuk menggunakan dan memodifikasinya [16].

2.2.4 HTML (*Hyper Text Markup Language*)

Hypertext Markup Language atau lebih dikenal dengan singkatan HTML memungkinkan seorang *user* menyusun dan membuat paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman web dan aplikasi. Ketika bekerja dengan HTML, *mark-up* halaman *website* menggunakan struktur kode yang sederhana (tag dan atribut). Seperti contoh ketika membuat sebuah paragraf pada HTML digunakan sebuah tag pembuka `<p>` dan tag penutup `</p>` dan menempatkan text diantara kedua tag tersebut [17].

2.2.5 Web Server

Web Server memiliki fungsi melayani permintaan pemanggilan alamat dari pengguna melalui *web browser*. *Web browser* mengirimkan kembali informasi yang diminta melalui HTTP yang kemudian ditampilkan ke layar monitor pengguna. HTTP atau *Hypertext Transfer Protocol* merupakan salah satu protokol aplikasi paling populer yang digunakan di internet [18].

2.2.6 PHP (*Hypertext Preprocessor*)

Hypertext Preprocessor atau PHP merupakan sebuah *script* yang berintegrasi dengan HTML dan berada pada *server* (*server-side HTML embedded scripting*). Untuk membuat halaman web menjadi lebih dinamis dan interaktif digunakan *script* PHP [15].

2.2.7 Website

Website merupakan kumpulan dari beberapa *webpage* yang terhubung dalam satu domain yang sama. Pada dasarnya website adalah

kepanjangan dari *Word Wide Web* (WWW). Informasi pada WWW ini disimpan pada *web server* untuk dapat diakses dari jaringan browser terlebih dahulu seperti Chrome atau Mozilla Firefox [15].

2.2.8 Rekayasa Perangkat Lunak

Pembangunan dengan menggunakan prinsip ataupun konsep merupakan tujuan dari rekayasa perangkat lunak untuk menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan dapat bekerja secara efisien dengan menggunakan mesin. Rekayasa perangkat lunak dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan juga harus dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi. Efisien dari segi sumber daya dan penggunaan serta kemampuan untuk dipakai sesuai dengan kebutuhan sehingga perangkat lunak yang baik adalah perangkat lunak yang dapat memenuhi kebutuhan user [19].

2.2.8.1 Metode Pengembangan Sistem

Dalam merancang dan membangun aplikasi Sistem Pendukung Keputusan Pemilihan Dosen Pembimbing Jurusan Teknik Informatika Politeknik Negeri Cilacap digunakan metode *waterfall*. Menurut (Pressman, 2012) *waterfall* merupakan model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Tahapan pengembangannya antara lain sebagai berikut [20] :

1. Analisis kebutuhan perangkat lunak

Langkah ini adalah langkah analisis terhadap kebutuhan *software* serta tahap untuk mengadakan pengumpulan data. Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain Sistem

Proses ini merupakan kelanjutan dari langkah analisis kebutuhan perangkat lunak dimana tahap ini menghasilkan data yang berhubungan dengan keinginan user agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pengkodean Sistem

Desain sistem yang telah dibuat pada tahap sebelumnya harus diterjemahkan dalam bahasa yang dapat dikenali oleh komputer.

Programmer akan menerjemahkan transaksi yang diminta oleh *user*. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Proses pengujian dilakukan untuk meminimalisir kesalahan dan memastikan keluaran sistem yang dihasilkan sesuai dengan yang diinginkan oleh *user*. Biasanya pengujian pada perangkat lunak berfokus dari segi logika dan fungsional.

2.2.8.2 Metode Pengujian Sistem

Digunakan metode pengujian *Black-Box Texting* yang merupakan pengujian berfokus pada spesifikasi fungsional dari perangkat lunak, serta dapat mendefinisikan kumpulan kondisi input dan melakukan pengetestan pada spesifikasi fungsional program [21].

Black-Box Testing berfokus untuk menemukan kesalahan dengan beberapa kategori sebagai berikut [22] :

1. Fungsi – fungsi yang rusak ataupun salah
2. Kesalahan antarmuka perangkat lunak
3. Kesalahan kinerja
4. Kesalahan inialisasi ataupun terminasi
5. Kesalahan struktur data atau akses basis data

Penulis menggunakan salah satu klasifikasi *black-box testing* yaitu pengujian usabilitas atau *Usability Testing*. Pengujian ini dikenal juga sebagai pengujian untuk keakraban pengguna (*testing for user friendliness*). Pengujian usabilitas juga merupakan proses yang bekerja dengan pengguna akhir secara langsung maupun tidak langsung guna menilai bagaimana pengguna merasakan perangkat. Proses ini akan membongkar area kesulitan pengguna dalam menggunakan aplikasi yang telah dibuat.

2.2.9 DBMS (*Database Management System*)

Database Management System atau dikenal dengan singkatan DBMS yang dalam bahasa Indonesia disebut sebagai Sistem Manajemen Basis Data digunakan untuk menyimpan, mengelola, serta menampilkan data. Oleh karena itu DBMS memegang peran yang amat penting bagi suatu organisasi / perusahaan sehingga hampir sebagian besar perusahaan

memanfaatkan DBMS dalam mengelola data yang mereka miliki. Salah satu contoh DBMS yang sangat sering digunakan diantaranya adalah *MySQL* dan *MariaDB* [1].

2.2.10 Basis Data

Sistem basis data adalah sebuah sistem terkomputerisasi yang digunakan untuk memelihara data yang sudah diolah atau informasi dan membuat informasi tersebut tersedia saat dibutuhkan. Secara garis basis data merupakan media penyimpanan data agar dapat diakses dengan lebih mudah dan cepat. Sedangkan kebutuhan data dalam sistem informasi meliputi: memasukkan, menyimpan, dan mengambil data [1].

Basis Data juga dikenal sebagai *Database*. Cara mengelola *database* adalah dengan menuliskan kode perintah dalam bentuk *query* SQL yang harus disesuaikan dengan kebutuhan. Ada dua jenis perintah SQL, yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). Berikut ini adalah definisi dari DDL dan DML:

1. DDL

DDL (*Data Definition Language*) digunakan untuk mengoperasikan skema struktur pada *database*. Beberapa perintah utama yang dapat digunakan pada DDL anatara lain adalah *create*, *rename*, *alter*, dan *drop*. Tujuan dari penggunaan perintah – perintah tersebut adalah untuk mengelola struktur data dalam *database*.

a. *Create*

Create digunakan untuk membuat sebuah *database* baru dan tabel baru. Contoh perintah yang menggunakan *create*, “CREATE TABLE mahasiswa(nim INTEGER PRIMARY KEY, nama VARCHAR(30) NOT NULL, email VARCHAR(30) NOT NULL);”

b. *Rename*

Rename digunakan jika kita ingin mengganti nama tabel. Contoh perintah yang menggunakan *rename*, “RENAME TABLE mahasiswa TO mhs;”

c. *Alter*

Merupakan perintah yang digunakan untuk mengubah tabel. Contoh perintah menggunakan *alter*, “Alter TABLE mhs ADD tanggal_lahir Date;”

d. *Drop*

Perintah ini digunakan untuk menghapus tabel maupun *database* yang telah dibuat. Contoh perintah menggunakan *drop*, “Drop Table mhs;”

2. DML

DML (*Data Manipulation Language*) adalah jenis perintah yang digunakan untuk melakukan operasi pada *database*, seperti menambah, menghapus, membuat baru, dan menampilkan data. Beberapa perintah utama pada DML mencakup *select*, *insert*, *update*, dan *delete*. Perintah – perintah ini berguna untuk memanipulasi dan mengatur isi dari *database*.

a. *Select*

Select digunakan ketika ingin menampilkan data yang telah ditambahkan. Contohnya, “SELECT * FROM mhs;”

b. *Insert*

Insert digunakan ketika ingin menambahkan data data ke dalam *database*. Contoh perintah dengan menggunakan *insert*, “INSERT INTO mhs (nim, nama, email, tanggal_lahir) values (200102008, “Salma Azizah”, “27-03-2002”);”.

c. *Update*

Update merupakan perintah yang digunakan untuk pembaruan data. Contoh kita menggunakan perintah *update* untuk memperbarui data nama, “UPDATE mhs SET nama = “Salma Azizah” WHERE nim = 200102008;”

d. *Delete*

Delete digunakan ketika ingin menghapus data yang ada. Contoh perintah, “DELETE FROM mhs WHERE nim=”200102008”;”

2.2.11 UML (*Unified Modelling Language*)

Untuk membuat sebuah aplikasi, diperlukan suatu proses pemodelan yang serupa dengan perancangan, di mana pemodelan merupakan bentuk implementasi sistem bagaimana suatu desain direpresentasikan dalam bentuk bentuk atau diagram. UML (*Unified Modelling Language*) adalah bahasa pemodelan visual standar yang dapat membantu seorang programmer memahami, menganalisis, dan memfasilitasi pembuatan program. Biasanya, sebuah program aplikasi merupakan suatu sistem yang akan digunakan dan diterapkan dalam

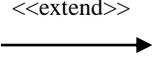
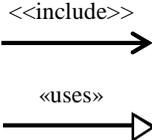
jangka waktu yang lama. Oleh karena itu, UML dianggap sebagai suatu bahasa standar untuk visualisasi, perancangan, dan dokumentasi sistem, dan disebut juga sebagai bahasa standar untuk membuat blueprint dari sebuah perangkat lunak. Diharapkan UML dapat membantu pengembangan perangkat lunak (RPL) dan memenuhi semua kebutuhan pengguna dengan cara yang efektif, lengkap, dan akurat, termasuk faktor-faktor seperti *scalability*, *robustness*, *security*, dan sebagainya [23].

2.2.11.1 Use Case Diagram

Use case diagram merupakan sebuah diagram yang menunjukkan hubungan interaksi antara aktor dengan sistem. *Use case* digunakan untuk mengetahui fungsi apa saja yang terdapat di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut [23]. Dua hal utama pada *use case* dimana aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, biasanya aktor ditandai dengan simbol orang tapi meskipun simbol dari aktor adalah gambar orang, belum tentu aktor merupakan orang. Hal utama yang kedua adalah *use case* yang merupakan fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit dan aktor.

Tabel 2. 3 *Use Case* Diagram

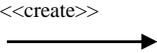
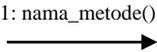
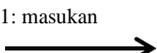
| No | Gambar Simbol | Nama | Keterangan |
|----|---|-----------------|---|
| 1 |  | <i>Use case</i> | Sistem menyediakan fungsionalitas yang memungkinkan unit – unit atau aktor untuk saling bertukar pesan. Umumnya, penggunaan kata depan di depan nama <i>use case</i> digunakan untuk menggambarkan fungsionalitas tersebut. |

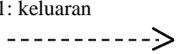
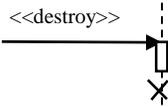
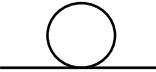
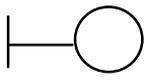
| | | | |
|---|---|--------------------------------------|--|
| 2 |  | Aktor / <i>actor</i> | Orang, proses, atau sistem lainnya dapat berinteraksi dengan sistem informasi yang akan dibuat, meskipun aktor yang digunakan dalam simbolisasi dapat berupa objek lain selain manusia. Biasanya, penggunaan kata benda di depan nama aktor digunakan untuk menjelaskan hal ini. |
| 3 |  | Asosiasi / <i>association</i> | Aktor dan <i>use case</i> terlibat dalam interaksi pada suatu <i>use case</i> , yang merupakan bentuk komunikasi antara keduanya. |
| 4 |  | Ekstensi / <i>extend</i> | <i>Use case</i> tambahan dapat dikaitkan dengan suatu <i>use case</i> , namun tetap dapat berdiri sendiri tanpa keberadaan <i>use case</i> yang ditambahkan. |
| 5 |  | Generalisasi / <i>generalization</i> | Terdapat hubungan generalisasi dan spesialisasi antara dua <i>use case</i> , dimana salah satu <i>use case</i> memiliki fungsi yang lebih umum dibandingkan dengan yang lain. |
| 6 |  | Include / <i>uses</i> | <i>Use case</i> tambahan ini dapat memiliki ketergantungan pada <i>use case</i> lainnya, baik sebagai syarat ataupun sebagai bagian dari fungsinya. |

2.2.11.2 Sequence Diagram

Sequence diagram yang menunjukkan bagaimana objek – objek yang berbeda bekerja sama dan berinteraksi di dalam suatu *class*. Diagram ini terdiri dari beberapa komponen sebagai berikut [23]:

Tabel 2. 4 *Sequence* Diagram

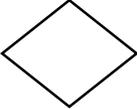
| No | Gambar Simbol | Nama | Keterangan |
|----|---|-------------------------------|--|
| 1 |  | Aktor / <i>actor</i> | Orang, proses, atau sistem lainnya dapat berinteraksi dengan sistem informasi yang akan dibuat, meskipun aktor yang digunakan dalam simbolisasi dapat berupa objek lain selain manusia. Biasanya, penggunaan kata benda di depan nama aktor digunakan untuk menjelaskan hal ini. |
| 2 |  | Garis hidup / <i>lifetime</i> | Menyatakan kehidupan suatu objek |
| 3 |  | Waku aktif | Menunjukkan objek dalam keadaan aktif dan saling berinteraksi, dengan semua tindakan yang terjadi selama periode aktif tersebut dijelaskan dalam diagram |
| 4 |  | Pesan tipe <i>create</i> | Menunjukkan objek yang membuat objek baru, dengan panah mengarah pada objek yang baru dibuat tersebut |
| 5 |  | Pesan tipe <i>call</i> | Menggambarkan saat suatu objek melakukan panggilan untuk menjalankan operasi atau metode yang ada pada objek lain atau dirinya sendiri |
| 6 |  | Pesan tipe <i>send</i> | Menunjukkan saat suatu objek mengirimkan data ke objek lainnya dan arah panah mengarah pada objek yang menerima data tersebut |

| | | | |
|----|---|---------------------------|---|
| 7 |  | Pesan tipe <i>return</i> | Menjelaskan saat suatu objek yang telah menjalankan operasi atau metode menghasilkan nilai kembalian untuk objek tertentu, dan arah panah mengarah pada objek yang menerima nilai kembalian tersebut. |
| 8 |  | Pesan tipe <i>destroy</i> | Menunjukkan saat suatu objek mengakhiri hidup objek lain dan arah panah mengarah pada objek yang diakhiri. Disarankan jika ada “ <i>create</i> ” maka juga terdapat “ <i>destroy</i> ” |
| 9 |  | <i>Entity Class</i> | Menggambarkan hubungan yang akan dilakukan |
| 10 |  | <i>Boundary Class</i> | Menggambarkan sebuah gambaran dari form |
| 11 |  | <i>Control Class</i> | Menggambarkan penghubung antar <i>boundary</i> dengan tabel |

2.2.12 ERD (Entity Relationship Diagram)

ERD atau *Entity Relationship Diagram* adalah sebuah diagram yang menggunakan notasi grafis dalam pembuatan *database* untuk menghubungkan antara data satu dengan yang lain. ERD berfungsi sebagai alat bantu dalam pembuatan *database* dan memberikan gambaran bagaimana *database* akan bekerja. Berikut ini merupakan simbol – simbol dari ERD [24]:

Tabel 2. 5 *Entity Relationship Diagram*

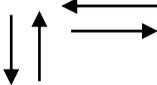
| No | Gambar Simbol | Nama | Keterangan |
|----|---|---------|--|
| 1 |  | Entitas | Entitas merupakan objek yang menjadi perhatian dalam database dan dapat berupa manusia, tempat, benda, atau kondisi yang berkaitan dengan data yang dibutuhkan |
| 2 |  | Relasi | Merupakan hubungan antara dua atau lebih entitas dan simbolnya berbentuk belah ketupat. Ada beberapa jenis relasi yang dapat dimiliki oleh ERD, seperti satu-satu (<i>one-to-one</i>), satu-ke-banyak (<i>one-to-many</i>), dan banyak-ke-banyak (<i>many-to-many</i>) |
| 3 |  | Atribut | Informasi yang terkait dengan entitas dan harus memiliki <i>primary key</i> sebagai ciri khas entitas dan atribut deskriptif. |
| 4 |  | Garis | Garis merupakan simbol yang digunakan sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut. |

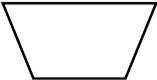
2.2.13 Flowchart

Flowchart merupakan sebuah metode penulisan algoritma yang menggunakan simbol – simbol grafis. *Flowchart* berupa gambar atau diagram yang menunjukkan urutan langkah – langkah dalam suatu program beserta hubungan antara proses – proses dan pernyataannya. Setiap simbol pada *flowchart* mewakili suatu proses tertentu, sementara garis penghubung menggambarkan hubungan antara proses tersebut.

Penggunaan *flowchart* dapat memudahkan dalam melakukan pengecekan terhadap bagian – bagian yang terlupakan dalam analisis masalah, sehingga dapat membantu programmer dalam memecahkan masalah menjadi segmen – segmen yang lebih kecil dan menganalisis alternatif – alternatif lain dalam pengoperasian. Berikut ini merupakan simbol – simbol di dalam *flowchart* [26].

Tabel 2. 6 *Flowchart*

| No | Gambar Simbol | Nama | Keterangan |
|----|---|-----------------------|---|
| 1 |  | <i>Terminal point</i> | Simbol ketika memulai atau berakhirnya suatu proses. |
| 2 |  | <i>Flow direction</i> | Simbol yang dipakai untuk menghubungkan antar simbol juga berperan dalam menandai urutan dari suatu proses. |
| 3 |  | <i>Process</i> | Mengindikasikan aktivitas yang dikerjakan oleh sistem atau perangkat komputer. |
| 4 |  | <i>Decision</i> | Simbol yang digunakan untuk memilih suatu keputusan atau proses yang sesuai dengan kondisi yang ada. |
| 5 |  | <i>Input / Output</i> | Menggambarkan suatu masukan atau keluaran yang tidak tergantung pada jenis peralatan yang digunakan. |

| | | | |
|----|---|-----------------------------|--|
| 6 |  | <i>Predefined process</i> | Mengindikasikan pelaksanaan suatu bagian dari prosedur, yang belum dijelaskan secara rinci dan akan dijelaskan secara lebih detail di tempat lain. |
| 7 |  | <i>Connector (On-page)</i> | Menghubungkan suatu simbol yang berada di halaman yang berbece secara jarak yang cukup jauh. |
| 8 |  | <i>Connector (Off-page)</i> | Menghubungkan simbol yang terletak pada halaman yang berbeda. |
| 9 |  | <i>Preparation</i> | Simbol ini menandakan persiapan untuk mengimpor data ke dalam penyimpanan (<i>storage</i>). |
| 10 |  | <i>Manual input</i> | Mengindikasikan proses input yang dilakukan secara manual melalui penggunaan <i>keyboard online</i> . |
| 11 |  | <i>Manual operation</i> | Menggambarkan kegiatan yang dilakukan tanpa menggunakan perangkat komputer. |
| 12 |  | <i>Document</i> | Menunjukkan input dalam bentuk dokumen kertas atau <i>output</i> yang perlu dicetak. |
| 13 |  | <i>Multiple document</i> | Serupa dengan simbol dokumen, tetapi dokumen yang digunakan dapat digunakan oleh lebih dari satu orang. |
| 14 |  | <i>Display</i> | Simbol yang menandakan penggunaan peralatan <i>output</i> . |
| 15 |  | <i>Delay</i> | Menunjukkan adanya keterlambatan (<i>delay</i>) dalam suatu proses. |

- HALAMAN INI SENGAJA DIKOSONGKAN -