

PROPOSAL TUGAS AKHIR

SISTEM PENDUKUNG KEPUTUSAN PENENTUAN CALON PENERIMA BEASISWA BIDIKMISI DAN PPA MENGGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW) BERBASIS *WEB*



Disusun oleh:
Yuni Rohmatin A1316127

**JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI TANAH LAUT
PELAIHARI
2019**

HALAMAN PERSETUJUAN

Sistem Pendukung Keputusan
Penentuan Calon Penerima Beasiswa Bidikmisi Dan PPA Menggunakan Metode
Simple Additive Weighting (SAW) Berbasis Web

Diusulkan oleh:

YUNI ROHMATIN
(A1316127)

Telah disetujui
Pada Tanggal: 26 Maret 2019

Pembimbing 1,

Pembimbing 2,

Wiwik Kusrini, S.Kom., M.Cs
NIK. 120102070

Agustian Noor, M.Kom
NIK. 160201153

DAFTAR ISI

HALAMAN JUDUL	I
HALAMAN PERSETUJUAN.....	II
DAFTAR ISI.....	III
DAFTAR GAMBAR	V
DAFTAR TABEL.....	VI
1. LATAR BELAKANG	1
2. RUMUSAN MASALAH.....	2
3. BATASAN MASALAH	2
4. TUJUAN PENELITIAN	3
5. MANFAAT PENELITIAN.....	3
6. TINJAUAN PUSTAKA.....	3
6.1 Sistem Pendukung Keputusan	3
6.2 Beasiswa Bidikmisi.....	6
6.3 Beasiswa PPA	7
6.4 <i>Simple Additive Weighting</i> (SAW)	8
6.5 <i>Web</i>	10
6.6 Basis Data	10
6.6.1 <i>Entity Relationship Diagram</i> (ERD)	11
6.6.2 MySQL	12
6.7 <i>Conceptual Data Model</i> (CDM).....	13
6.8 <i>Physical Data Model</i> (PDM).....	15
6.9 <i>Unified Modeling Language</i> (UML).....	15
6.9.1 <i>Use Case Diagram</i>	15
6.9.2 <i>Activity Diagram</i>	17
6.9.3 <i>Sequence Diagram</i>	19
6.9.4 <i>Class Diagram</i>	22
6.10 Bahasa Pemrograman	23
6.10.1 <i>HyperText Markup Language</i> (HTML).....	23
6.10.2 <i>Cascading Style Sheets</i> (CSS)	24
6.10.3 <i>Hypertext Preprocessor</i> (PHP)	25
6.10.4 <i>JQuery</i>	25

6.10.5 AJAX	26
6.11 <i>Framework CodeIgniter</i>	26
6.12 <i>Software Development Life Cycle (SDLC)</i>	27
6.13 <i>Blackbox Testing</i> (Pengujian Kotak Hitam)	29
7. METODE PENELITIAN	30
7.1 Pengumpulan Data	31
7.2 Rancangan Penelitian	32
7.3 Rancangan Pengujian	33
8. JADWAL PENELITIAN	33
DAFTAR PUSTAKA	34

DAFTAR GAMBAR

Gambar 1. Arsitektur Sistem Pendukung Keputusan	5
Gambar 2. Formula Normalisasi Matriks.....	9
Gambar 3. Formula Perangkingan	9
Gambar 4. Contoh Diagram <i>Use Case</i>	17
Gambar 5. Contoh Diagram <i>Activity</i>	19
Gambar 6. Contoh Diagram <i>Sequence</i>	21
Gambar 7. Struktur Dokumen HTML	24
Gambar 8. Contoh <i>Source Code</i> HTML	24
Gambar 9. Contoh Kode CSS	24
Gambar 10. Contoh Kode PHP	25
Gambar 11. Contoh Kode <i>JQuery</i>	26
Gambar 12. Contoh Kode AJAX	26
Gambar 13. Ilustrasi Metode <i>Waterfall</i>	28
Gambar 14. Alur Penelitian.....	31

DAFTAR TABEL

Tabel 1. Daftar Simbol ERD (<i>Entity Relationship Diagram</i>)	11
Tabel 2. Simbol-simbol <i>Conceptual Data Model</i> (CDM)	13
Tabel 3. Aturan Mengubah ERD menjadi CDM	13
Tabel 4. Simbol-simbol <i>Physical Data Model</i> (PDM).....	15
Tabel 5. Simbol-simbol <i>Use Case Diagram</i>	16
Tabel 6. Simbol-simbol <i>Activity Diagram</i>	18
Tabel 7. Simbol-simbol <i>Sequence Diagram</i>	20
Tabel 8. Simbol-simbol Diagram Kelas	22
Tabel 9. Jadwal Penelitian.....	33

SISTEM PENDUKUNG KEPUTUSAN
PENENTUAN CALON PENERIMA BEASISWA BIDIKMISI DAN PPA
MENGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)
BERBASIS *WEB*

1. LATAR BELAKANG

Program beasiswa Bidikmisi dan PPA merupakan program kerja yang ada pada setiap universitas maupun perguruan tinggi. Tujuan dari program tersebut adalah meringankan beban mahasiswa dalam menempuh masa studi kuliah khususnya dalam masalah biaya. Selain itu juga untuk memberikan apresiasi terhadap mahasiswa yang mempunyai prestasi. Penentuan mahasiswa yang menerima beasiswa dilakukan secara selektif sesuai dengan jenis beasiswa yang diadakan.

Peraturan tentang beasiswa dituangkan dalam Undang-undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional, Bab V pasal 12 (1.c), menyebutkan bahwa setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan beasiswa bagi yang berprestasi yang orang tuanya kurang mampu membiayai pendidikannya. Pasal 12 (1.d), menyebutkan bahwa setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan biaya pendidikan bagi mereka yang orang tuanya kurang mampu membiayai pendidikannya.

Perguruan tinggi Politeknik Negeri Tanah Laut menyediakan program beasiswa, yaitu Beasiswa peningkatan Prestasi Akademik (PPA) dan Beasiswa Bidikmisi. Setiap program beasiswa memiliki kriteria atau faktor bobot penilaian yang berbeda-beda. Pada umumnya berdasarkan peraturan Direktorat Jendral Pendidikan Tinggi (DIKTI) pada situs www.dikti.go.id perolehan beasiswa memiliki kriteria Indeks Prestasi Kumulatif (IPK) paling rendah 3.0, jenjang S1/Diploma IV paling rendah duduk pada semester II dan paling tinggi duduk pada semester VIII, sedangkan untuk jenjang Diploma III paling rendah duduk pada semester II dan paling tinggi duduk pada semester VI, penghasilan orang tua, tanggungan orang tua dan jumlah saudara kandung. Kemudian apabila calon penerima melebihi kuota yang telah ditetapkan, maka perguruan tinggi dapat menentukan mahasiswa penerima beasiswa sesuai dengan urutan prioritas sebagai berikut: mahasiswa yang mempunyai IPK paling tinggi, mahasiswa yang mempunyai SKS paling banyak (jumlah semester paling sedikit), mahasiswa yang

memiliki prestasi dikegiatan ko/ekstra kurikuler (olahraga, teknologi, seni/budaya tingkat internasional/dunia/Regional/Asia/Asean dan Nasional) dan mahasiswa yang orang tuanya paling tidak mampu.

Bagian kemahasiswaan setiap ajaran baru penerimaan beasiswa harus menyeleksi mahasiswa-mahasiswa yang layak mendapatkan beasiswa. Proses penyeleksian tersebut membutuhkan ketelitian dan waktu karena setiap berkas yang dikumpulkan oleh mahasiswa akan dibandingkan dengan kriteria beasiswa satu persatu. Sehingga dibutuhkan suatu sistem yang dapat membantu membuat keputusan calon penerima beasiswa dengan cepat dan tepat untuk meringankan kerja bagian mahasiswa dalam penentuan calon penerima beasiswa.

Sistem yang akan dibangun untuk penentuan calon penerima beasiswa menggunakan metode SAW (*Simple Additive Weighting*). Metode tersebut menentukan nilai bobot untuk setiap atribut yang kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif. Penilaian diharapkan akan lebih tepat dengan adanya metode perankingan tersebut, karena didasarkan pada kriteria dan bobot yang sudah ditentukan sehingga akan mendapatkan hasil yang lebih akurat terhadap siapa saja yang berhak menerima beasiswa.

2. RUMUSAN MASALAH

Berdasarkan latar belakang permasalahan yang telah diuraikan maka rumusan masalahnya adalah “Bagaimana Membangun Sebuah Sistem Pendukung Keputusan Penentuan Calon Penerima Beasiswa Bidikmisi dan PPA Menggunakan Metode *Simple Additive Weighting* (SAW) Berbasis *Web*?”.

3. BATASAN MASALAH

Batasan masalah dalam penelitian ini sebagai berikut:

- a. Studi kasus dalam penelitian ini di Perguruan Tinggi Politeknik Negeri Tanah Laut.
- b. Sistem Pendukung Keputusan yang dibangun menggunakan metode *Simple Additive Weighting* (SAW) dan berbasis *web*.
- c. *Database* yang digunakan yaitu MySQL.

- d. Sistem dibangun menggunakan *framework CodeIgniter* teknik HMVC dengan bahasa pemrograman yang digunakan adalah HTML, PHP, CSS, *JavaScript*.

4. TUJUAN PENELITIAN

Tujuan dari penelitian ini berdasarkan kasus yang diuraikan pada latar belakang adalah sebagai berikut:

- a. Membangun Sistem Pendukung Keputusan Penentuan Calon Penerima Beasiswa Bidikmisi dan PPA Menggunakan Metode *Simple Additive Weighting* (SAW) berbasis *Web*.
- b. Memberikan solusi bagi mahasiswa dalam pengambilan keputusan penentuan calon penerima beasiswa dengan cepat dan tepat.

5. MANFAAT PENELITIAN

Manfaat yang diperoleh dari penelitian ini yaitu sebagai berikut:

- a. Membantu meringankan kerja bagian mahasiswa dalam menentukan calon penerima beasiswa Bidikmisi dan PPA dengan cepat dan tepat sehingga lebih efisien dari segi waktu.
- b. Penulis dapat mengembangkan Sistem Pendukung Keputusan Menggunakan Metode SAW.

6. TINJAUAN PUSTAKA

6.1 Sistem Pendukung Keputusan

Pengambilan keputusan dapat diartikan sebagai suatu kegiatan memilih alternatif terbaik diantara beberapa alternatif yang ada (Diana, 2018). Teori pengambilan keputusan organisasi dikembangkan di *Carnegie institute of Technology* pada tahun 1950. Teori dalam pengambilan keputusan antara lain:

1) Teori Rasional Komprehensif.

Pada teori ini alternatif-alternatif pilihan dievaluasi dengan seksama, pengambilan keputusan mengambil keputusan dengan berdasarkan pada pedoman yang jelas berdasarkan pada tujuan atau sasaran yang telah ditetapkan dan telah diurutkan tingkat prioritasnya, masalah-masalah yang akan dipecahkan diselesaikan berdasarkan urutan prioritas, pengambilan keputusan akan memilih alternatif terbaik yang bersesuaian dengan tujuan atau sasaran.

2) Teori *Incremental*

3) Teori Pengamatan Terpadu

Teori ini merupakan penggabungan teori rasional komprehensif dan *incremental*, sehingga teori ini memungkinkan pembuat keputusan untuk menggunakan kedua teori ini dalam proses pengambilan keputusan.

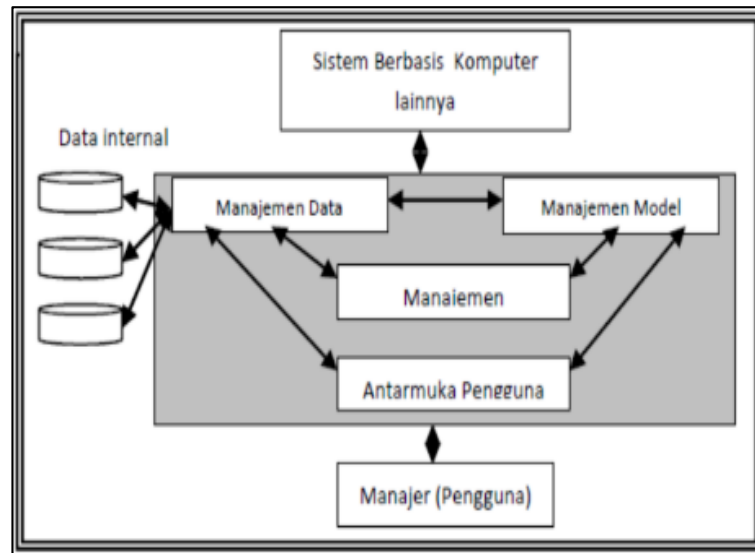
Selanjutnya *Massachusetts Institute of Technology*, pada tahun 1960-an, mengimplementasikan sistem pendukung keputusan dalam bentuk sistem komputer interaktif. Sistem berbasis komputer yang mendukung pengambilan keputusan dalam bidang produksi, promosi, penetapan harga, pemasaran dan beberapa fungsi logistik mulai dikembangkan pada tahun 1980-an. Perkembangan selanjutnya yaitu pada tahun 1990-an dimana terjadi pergeseran dalam Sistem Pendukung Keputusan dimana dikembangkan sistem yang lebih kompleks dengan menggabungkan teknologi basis data yang canggih dan teknologi komputer yang menerapkan *client/server*.

Professor dari MIT yaitu G. Anthony Gorry dan Michael S.Scott Morton memperkenalkan Sistem Pendukung Keputusan dimana mereka mengembangkan kerangka pemikiran tentang pemanfaatan aplikasi komputer pada proses pengambilan keputusan bagi level manajemen. Sistem Pendukung Keputusan yaitu berkaitan erat dengan sistem informasi atau model analisis yang dirancang untuk membantu para pengambil keputusan dan para profesional agar mendapatkan informasi yang akurat (Diana, 2018). Tujuan implementasi sistem pendukung keputusan antara lain:

- a) Sistem pendukung keputusan dapat memproses data dengan cepat dan dalam jumlah yang banyak sehingga lebih efisien dari segi waktu untuk para pengambil keputusan.
- b) Sistem pendukung keputusan diharapkan dapat membantu manajer pengambil keputusan dengan dukungan data informasi yang akurat menghasilkan keputusan yang lebih akurat dan berkualitas.
- c) Menghasilkan keputusan yang efektif dan efisien dari segi biaya dan sumber daya.

- d) Meningkatkan tingkat pengendalian guna meningkatkan kemampuan untuk mendeteksi adanya kesalahan-kesalahan pada sistem sehingga dapat dilakukan antisipasi.
- e) Menghasilkan keputusan yang berkualitas karena keputusan yang diambil didasarkan pada data yang lengkap dan akurat.

Arsitektur sistem pendukung keputusan dapat digambarkan sebagai berikut:



Gambar 1. Arsitektur Sistem Pendukung Keputusan (Sumber: Sari, 2018)

Komponen utama pada sistem pendukung keputusan adalah:

1. Manajemen Data

Pada manajemen data terdapat 2 hal penting yaitu sumber data pada sistem basis data semua data bisnis yang dimiliki oleh perusahaan atau organisasi, data dapat berupa data transaksi sehari-hari dan data dasar yang dimiliki perusahaan atau organisasi tersebut.

2. Manajemen Model

Model manajemen meliputi model finansial, statistika, manajemen *sains*, yang dapat diimplementasikan dan meningkatkan kemampuan analisis pada sistem pendukung keputusan. Beberapa model yang bias diimplementasikan pada sistem pendukung keputusan yaitu, model kuantitatif, model analog dan model *ionic*.

3. Antarmuka Pengguna

Antarmuka pengguna atau disebut juga manajemen dialog atau komunikasi atau sub sistem dialog berguna sebagai tempat pengguna berkomunikasi dan

memberi perintah ke dalam sistem pendukung keputusan. Komponen pendukung sistem dapat berupa data internal dan sistem lain berbasis komputer.

4. Pengguna

Sistem pendukung keputusan didesain untuk dapat digunakan oleh pengguna, manajer atau pengambil keputusan. Ada 2 jenis pengguna sistem yaitu:

- a. *Manager*, pengguna *manager* mengharapkan antarmuka yang mudah, tidak berhubungan dengan data tapi langsung melihat data yang telah diolah dalam bentuk informasi, tidak perlu melihat data secara detail.
- b. *Staff* analis merupakan perantara antara *manager* dan sistem pendukung keputusan.

6.2 Beasiswa Bidikmisi

Beasiswa Bidikmisi adalah bantuan biaya pendidikan dari pemerintah bagi lulusan Sekolah Menengah Atas (SMA) atau sederajat yang memiliki potensi akademik baik tetapi memiliki keterbatasan ekonomi (Kemenristekdikti, 2019). Bidikmisi bertujuan untuk meningkatkan akses dan kesempatan belajar di perguruan tinggi, meningkatkan prestasi mahasiswa, menjamin keberlangsungan studi mahasiswa dengan tepat waktu, dan melahirkan lulusan yang mandiri, produktif serta memiliki kepedulian sosial sehingga mampu berperan dalam upaya pemutusan mata rantai kemiskinan dan pemberdayaan masyarakat. Penerima bidikmisi adalah siswa SMA atau sederajat yang akan lulus pada tahun berjalan atau lulus 1 (satu) tahun sebelumnya, memiliki potensi akademik baik tetapi memiliki keterbatasan ekonomi yang didukung bukti dokumen yang sah dan lulus seleksi penerimaan mahasiswa baru pada perguruan tinggi.

Peraturan perundang-undangan yang dijadikan landasan dalam pemberian bantuan biaya pendidikan adalah:

1. Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional, Bab V pasal 12 (1.c), menyebutkan bahwa setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan beasiswa bagi yang berprestasi yang orang tuanya tidak mampu membiayai pendidikannya. Pasal 12 (1.d), menyebutkan bahwa setiap peserta didik pada setiap satuan

pendidikan berhak mendapatkan biaya pendidikan bagi mereka yang orang tuanya tidak mampu membiayai pendidikannya.

2. Undang-Undang Republik Indonesia Nomor 12 Tahun 2012 tentang Pendidikan Tinggi, Pasal 76 ayat (1), menyebutkan bahwa Pemerintah, Pemerintah Daerah, dan atau Perguruan Tinggi berkewajiban memenuhi hak Mahasiswa yang tidak mampu secara ekonomi untuk dapat menyelesaikan studinya sesuai dengan peraturan akademik. Pasal 76 ayat (2), menyebutkan bahwa pemenuhan hak Mahasiswa sebagaimana dimaksud pada ayat (1) dilakukan dengan cara memberikan: (a) beasiswa kepada Mahasiswa berprestasi, (b) bantuan atau membebaskan biaya Pendidikan.
3. Peraturan Pemerintah Republik Indonesia Nomor 48 tahun 2008 tentang Pendanaan Pendidikan, Bagian Kelima, Pasal 27 ayat (1), menyebutkan bahwa Pemerintah dan Pemerintah Daerah sesuai kewenangannya memberi bantuan biaya pendidikan atau beasiswa kepada peserta didik yang orang tua atau walinya tidak mampu membiayai pendidikannya. Pasal 27 ayat (2), menyebutkan bahwa Pemerintah dan Pemerintah Daerah sesuai dengan kewenangannya dapat memberi beasiswa kepada peserta didik yang berprestasi.
4. Peraturan Pemerintah Republik Indonesia Nomor 66 Tahun 2010 tentang Perubahan Atas Peraturan Pemerintah Nomor 17 Tahun 2010 tentang Pengelolaan dan Penyelenggaraan Pendidikan, Pasal 53A yang menegaskan bahwa satuan pendidikan tinggi yang diselenggarakan oleh Pemerintah atau Pemerintah Daerah sesuai dengan kewenangan masing-masing wajib menyediakan beasiswa bagi peserta didik berkewarganegaraan Indonesia yang berprestasi dan wajib mengalokasikan tempat bagi calon peserta didik berkewarganegaraan Indonesia, yang memiliki potensi akademik baik dan tidak mampu secara ekonomi, paling sedikit 20% (dua puluh persen) dari jumlah keseluruhan peserta didik baru.

6.3 Beasiswa PPA

Sejak tahun 2012 pemerintah telah meluncurkan Beasiswa Peningkatan Prestasi Akademik (PPA) DAN Bantuan Belajar Mahasiswa (BBM) yang kemudian istilahnya disesuaikan menjadi Beasiswa Peningkatan Prestasi Akademik

(Beasiswa-PPA) dan Bantuan Biaya Pendidikan Peningkatan Prestasi Akademik (BPP-PPA) (Kemenristekdikti, 2018). Pada tahun 2018 skema beasiswa PPA hanya akan difokuskan untuk aspek Peningkatan Prestasi Akademik, sementara Bantuan Biaya Pendidikan sepenuhnya akan dibiayai melalui Bantuan Biaya Pendidikan Bidikmisi. Bagi setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan biaya pendidikan bagi mereka yang orang tuanya tidak mampu membiayai pendidikannya dan berhak mendapatkan beasiswa bagi mereka yang berprestasi.

Undang-undang Nomor 12 Tahun 2012 tentang Pendidikan Tinggi Pasal 76 bahwa: (1) Pemerintah, Pemerintah Daerah, dan atau Perguruan Tinggi berkewajiban memenuhi hak mahasiswa yang kurang mampu secara ekonomi untuk dapat menyelesaikan studinya sesuai dengan peraturan akademik. (2) Pemenuhan hak Mahasiswa sebagaimana dimaksud pada ayat (1) dilakukan dengan cara memberikan: a). beasiswa kepada Mahasiswa berprestasi; b). bantuan atau membebaskan biaya Pendidikan; dan atau c). pinjaman dana tanpa bunga yang wajib dilunasi setelah lulus dan atau memperoleh pekerjaan. (3) Perguruan Tinggi atau penyelenggara Perguruan Tinggi menerima pembayaran yang ikut ditanggung oleh mahasiswa untuk membiayai studinya sesuai dengan kemampuan mahasiswa, orangtua mahasiswa atau pihak yang membiayainya.

Berdasarkan Undang-Undang tersebut tentang Pendidikan Tinggi, Kementerian Riset, Teknologi dan Pendidikan Tinggi melalui Direktorat Jendral Pembelajaran dan Kemahasiswaan akan memberikan Beasiswa Peningkatan Prestasi Akademik (PPA).

6.4 Simple Additive Weighting (SAW)

Metode *Simple Additive Weighting* adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu yaitu dengan cara menentukan nilai bobot untuk setiap atribut kemudian dilanjutkan dengan proses perangkingan untuk menyeleksi alternatif yang ada (Chandra & Ocnota, 2018).

Konsep dasar metode *Simple Additive Weighting* adalah mencari penjumlahan berbobot dari rating kinerja pada setiap alternatif pada semua atribut (Nofriansyah, 2014). Metode *Simple Additive Weighting* digunakan untuk pengambilan keputusan

yang memiliki banyak atribut. Selain itu metode tersebut membutuhkan proses normalisasi matriks keputusan ke suatu skala kemudian hasil yang didapat dibandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut:

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\max X_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min X_{ij}}{X_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost) } X_{ij} \end{cases}$$

Gambar 2. Formula Normalisasi Matriks

Keterangan:

- r_{ij} = Rating kinerja ternormalisasi dari alternatif A, pada atribut C_j :
 $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.
- $\max X_{ij}$ = Nilai terbesar dari setiap kriteria i .
- $\min X_{ij}$ = Nilai terkecil dari setiap kriteria i .
- X_{ij} = Nilai atribut yang dimiliki dari setiap kriteria.
- Benefit* = Jika nilai terbesar adalah terbaik.
- Cost* = Jika nilai terkecil adalah terbaik.

Formula untuk perangkingan dengan metode *Simple Additive Weighting* sebagai berikut:

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Gambar 3. Formula Perangkingan

Keterangan:

- V_i = Rangking untuk setiap alternatif.
- w_j = Nilai bobot rangking (dari setiap kriteria).
- r_{ij} = Nilai rating kinerja ternormalisasi.

Keuntungan dari metode *Simple Additive Weighting* dibandingkan dengan metode sistem keputusan yang lain terletak pada kemampuannya dalam melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot tingkat kepentingan yang dibutuhkan. Selain itu metode SAW juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada. Kemudian alternatif terbaik

tersebut dilakukan proses perangkingan yang jumlah nilai bobot dari semua kriteria dijumlahkan setelah menentukan nilai bobot dari setiap kriteria.

6.5 Web

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen *multimedia* (teks, gambar, suara, animasi, video) didalamnya yang menggunakan *protocol* HTTP (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser* (Arief, 2011). Tim Berners-Lee pada tahun 1989 menemukan metode baru dalam menghubungkan dokumen satu dengan lainnya di *internet*. Selanjutnya metode tersebut dikembangkan hingga menjadi sebuah *web* pada tahun 1993 *web browser* yang berbasis grafis ditemukan oleh Marc Andreessen dengan nama *web browser*-nya yaitu *Mosaic*. Pada tahun 1994, Marc Andreessen mengembangkan *Mosaic* menjadi sebuah *web browser* pertama mereka yaitu *Internet Explorer*.

Web dibagi menjadi 2 jenis berdasarkan aspek *content* atau isi yaitu *web* statis dan *web* dinamis. *Web* statis adalah *web* yang isinya/*content* tidak berubah-ubah atau tidak dapat diubah secara cepat dan mudah (Arief, 2011). Teknologi yang digunakan untuk *web* statis yaitu jenis *client side scripting* seperti HTML dan CSS dimana perubahan isi data pada halaman *web* dapat dirubah pada *file* HTML dan CSS. Perubahan tersebut dapat dilakukan oleh seorang *programmer web* atau yang mempunyai pengetahuan tentang *client side scripting*.

Web dinamis adalah jenis *web* yang *content*/isinya dapat berubah-ubah setiap saat dimana untuk melakukan perubahannya *user* cukup mengubahnya langsung pada halaman *web* tanpa perlu membuka *file* mentahnya (Arief, 2011). Fitur yang ada pada *web* dinamis dibuat semudah untuk mengelola isi/*content* yang ada pada *web* tersebut karena *user* yang menggunakan *web* tersebut kemungkinan bukanlah *user* yang menguasai detail teknis bahasa pemrograman dan *database*. Teknologi yang digunakan yaitu *client side scripting* (HTML, *JavaScript*, CSS) dan *server side scripting* (PHP, program basis data seperti MySQL untuk menyimpan data).

6.6 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia

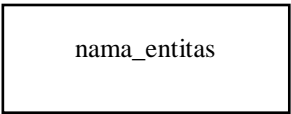
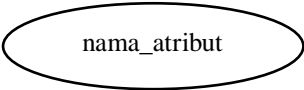
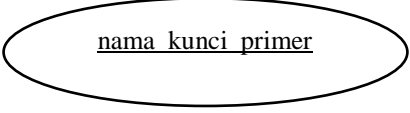
saat dibutuhkan (Sukanto & Shalahuddin, 2016). Sehingga dapat disimpulkan bahwa basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data dalam sistem informasi meliputi:

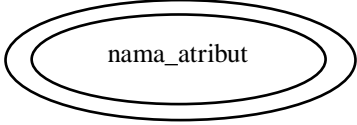
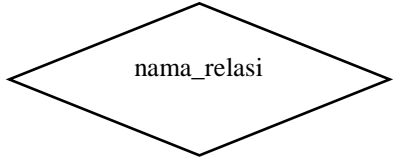
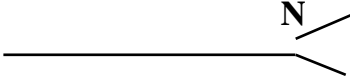
1. Memasukkan, menyimpan dan mengambil data.
2. Membuat laporan berdasarkan data yang telah disimpan.

6.6.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu rancangan atau bentuk hubungan suatu kegiatan didalam sistem yang berkaitan langsung dan mempunyai fungsi didalam proses tersebut (Puspitasari, 2016). Model data E – R pertama kali diperkenalkan oleh Chen (1976) pada artikelnya yang mendiskusikan konstruksi utama dari model E – R, hubungan antar entitas (*relationship*) serta atribut-atribut yang bersesuaian dengan tiap entitas. Model yang diperkenalkan oleh Chen kemudian diperluas dan dikembangkan oleh Teorrey, Yang, Fry (1986), serta Story (1991) saat ini model E – R masih berkembang namun tidak ada notasi baku untuk pemodelan E – R. Selain itu ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan yang lain yang sering disebut dengan entitas. Simbol-simbol yang digunakan pada ERD dengan notasi Chen sebagai berikut :

Tabel 1. Daftar Simbol ERD (*Entity Relationship Diagram*)

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel 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 tabel.
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

Simbol	Deskripsi
Atribut multivalai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas dimana dikedua ujungnya memiliki <i>multiplicity</i> memungkinkan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dengan entitas B.

Sumber : Rosa A.S. & M. Shalahuddin (2016)

6.6.2 MySQL

MySQL adalah salah satu DBMS (*Database Management System*) yang mempunyai fungsi mengolah *database* menggunakan bahasa SQL (*Structured Query Language*) (Anhar, 2010). *Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur *file* yang menyusun sebuah *database* adalah *Data Record* dan *Field*. Data adalah satu satuan informasi yang akan diolah. Sedangkan *record* adalah data yang isinya merupakan satu kesatuan seperti *username* dan *password*.

SQL (*Structured Query Language*) mempunyai 3 jenis perintah yaitu:

1. *Data Definition Language* (DDL)

Data Definition Language (DDL) merupakan perintah SQL yang digunakan untuk melakukan definisi awal atau perintah dasar suatu basis data dan tabel pada konsep RDBMS yang meliputi perintah *create*, *alter* dan *drop* (Priyadi, 2014).

2. *Data Manipulation Language* (DML)

Data Manipulation Language (DML) merupakan perintah SQL yang digunakan untuk melakukan pengolahan *record* atau manipulasi *record* pada

tabel dalam suatu basis data yang meliputi perintah *insert*, *select*, *update* dan *delete* (Priyadi, 2014).

3. *Data Control Language* (DCL)

Data Control Language (DCL) merupakan perintah SQL yang digunakan untuk melakukan pengaturan hak akses suatu objek data para pengguna dalam basis data yang meliputi perintah *grant* dan *revoke* (Priyadi, 2014).

6.7 *Conceptual Data Model* (CDM)

Conceptual Data Model (CDM) atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data (Sukamto & Shalahuddin, 2016). CDM dibuat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran dari ERD dimana untuk mengkonversi ERD menjadi CDM terdapat aturan-aturan yang harus diikuti. Simbol-simbol yang ada pada CDM sebagai berikut:

Tabel 2. Simbol-simbol *Conceptual Data Model* (CDM)


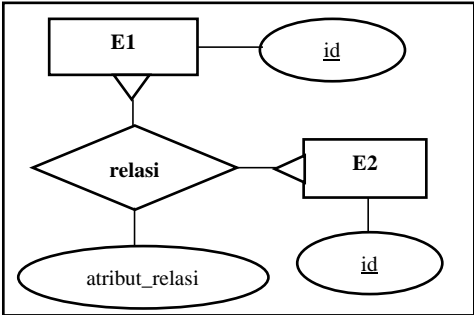
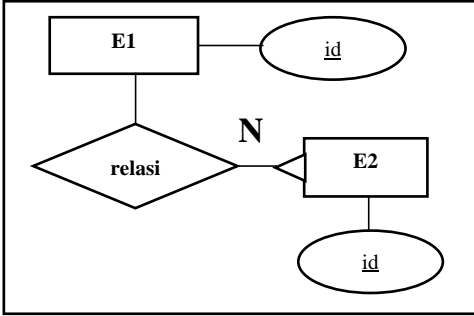
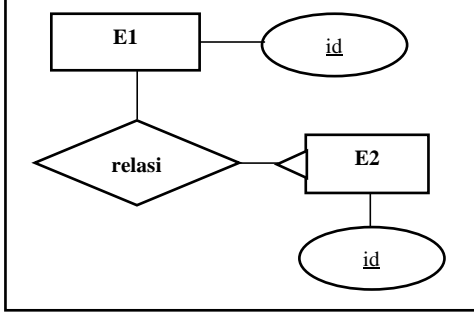
Simbol	Deskripsi
Entitas / tabel <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 150px;"> nama_tabel </div>	Entitas atau tabel yang menyimpan data dalam basis data.
Relasi <div style="display: flex; align-items: center; margin: 10px auto; width: 150px;"> <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin-right: 10px;"> 1 ..* nama_relasi 1 ..* </div> </div>	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i> .

Sumber : Rosa A.S. & M. Shalahuddin (2016)

Aturan untuk mengubah ERD menjadi CDM secara umum adalah sebagai berikut:

Tabel 3. Aturan Mengubah ERD menjadi CDM

ERD	CDM												
entitas <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 100px; text-align: center;"> petugas </div>	<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 150px;"> <table> <tr> <th colspan="2">Petugas</th></tr> <tr> <td style="text-align: center;">PK</td><td style="text-align: center;"><u>Username</u></td></tr> <tr> <td></td><td>Password</td></tr> <tr> <td></td><td>Nama</td></tr> <tr> <td></td><td>No_petugas</td></tr> <tr> <td></td><td>Hak_akses</td></tr> </table> </div> <p>Menjadi sebuah tabel tersendiri.</p>	Petugas		PK	<u>Username</u>		Password		Nama		No_petugas		Hak_akses
Petugas													
PK	<u>Username</u>												
	Password												
	Nama												
	No_petugas												
	Hak_akses												

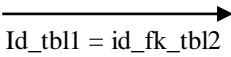
ERD	CDM								
 <p>Atribut <i>multivalue</i></p>	<table border="1"> <thead> <tr> <th colspan="2">Pengarang</th></tr> </thead> <tbody> <tr> <td>PK</td><td><u>Id_pustaka</u></td></tr> <tr> <td>PK</td><td><u>pengarang</u></td></tr> <tr> <td></td><td></td></tr> </tbody> </table> <p>Menjadi sebuah tabel tersendiri dengan kunci primer (<i>primary key</i>) adalah kunci primer pada entitas dan memiliki atribut dengan nama seperti pada atribut entitas.</p>	Pengarang		PK	<u>Id_pustaka</u>	PK	<u>pengarang</u>		
Pengarang									
PK	<u>Id_pustaka</u>								
PK	<u>pengarang</u>								
 <p>Relasi dengan kardinalitas <i>many to many</i></p>	<table border="1"> <thead> <tr> <th colspan="2">Relasi</th></tr> </thead> <tbody> <tr> <td>PK</td><td><u>Id_E1</u></td></tr> <tr> <td>PK</td><td><u>Id_E2</u></td></tr> <tr> <td></td><td>Atribut_relasi</td></tr> </tbody> </table> <p>Menjadi sebuah tabel tersendiri dengan kunci primer adalah atribut yang menjadi kunci primer dikedua entitas yang direlasikannya.</p>	Relasi		PK	<u>Id_E1</u>	PK	<u>Id_E2</u>		Atribut_relasi
Relasi									
PK	<u>Id_E1</u>								
PK	<u>Id_E2</u>								
	Atribut_relasi								
 <p>Relasi dengan kardinalitas <i>one to many</i></p>	<table border="1"> <thead> <tr> <th colspan="2">E2</th></tr> </thead> <tbody> <tr> <td>PK</td><td><u>Id_E1</u></td></tr> <tr> <td>PK</td><td><u>Id_E2</u></td></tr> <tr> <td></td><td></td></tr> </tbody> </table> <p>Kunci primer entitas yang memiliki hubungan <i>one</i> akan dijadikan kunci primer di entitas yang dimiliki hubungan <i>many</i> dengan kata lain, relasi tidak menjadi tabel sendiri.</p>	E2		PK	<u>Id_E1</u>	PK	<u>Id_E2</u>		
E2									
PK	<u>Id_E1</u>								
PK	<u>Id_E2</u>								
 <p>Relasi dengan kardinalitas <i>one to one</i></p>	<p>Kunci primer salah satu entitas akan dijadikan kunci asing (<i>foreign key</i>) pada tabel yang lain dan kunci asing itu dijadikan kunci primer juga, dengan kata lain, relasi tidak menjadi tabel sendiri.</p> <table border="1"> <thead> <tr> <th colspan="2">E2</th></tr> </thead> <tbody> <tr> <td>PK</td><td><u>Id_E1</u></td></tr> <tr> <td>PK</td><td><u>Id_E2</u></td></tr> <tr> <td></td><td></td></tr> </tbody> </table>	E2		PK	<u>Id_E1</u>	PK	<u>Id_E2</u>		
E2									
PK	<u>Id_E1</u>								
PK	<u>Id_E2</u>								

Sumber : Rosa A.S. & M. Shalahuddin (2016)

6.8 Physical Data Model (PDM)

Physical Data Model (PDM) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antar data (Sukamto & Shalahuddin, 2016). Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik beserta tipe datanya. PDM merupakan konsep yang menerangkan detail dari bagaimana data disimpan didalam basis data. PDM merupakan bentuk fisik perancangan basis data yang sudah siap diimplementasikan kedalam DBMS sehingga nama tabel merupakan nama asli tabel yang akan diimplementasikan kedalam DBMS. Simbol-simbol yang ada pada PDM adalah sebagai berikut:

Tabel 4. Simbol-simbol *Physical Data Model* (PDM)

Simbol	Deskripsi
Tabel nama_tabel	Tabel yang menyimpan data dalam basis data.
Relasi  Id_tbl1 = id_fk_tbl2	Relasi antar tabel yang terdiri dari persamaan antara <i>primary key</i> (kunci primer) tabel yang diacu dengan kunci yang menjadi referensi acuan ditabel lain.

Sumber : Rosa A.S. & M. Shalahuddin (2016)

6.9 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah standar Bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Sukamto & Shalahuddin, 2016).

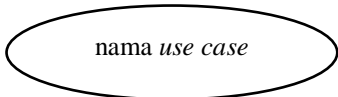
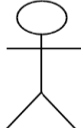

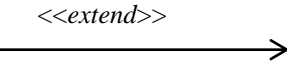
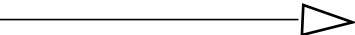
6.9.1 Use Case Diagram

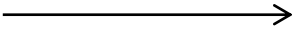
Use case merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat (Sukamto & Shalahuddin, 2016). *Use case diagram* merupakan deskripsi lengkap tentang interaksi yang terjadi antara para *actor* dengan sistem/perangkat lunak yang sedang dibuat atau dikembangkan (Nugroho, 2009). Diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Dua hal utama pada *use case* yaitu pendefinisian apa yang disebut *actor* dan *use case*.

- a. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, sehingga *actor* belum tentu merupakan orang meskipun simbol *actor* adalah gambar orang. Setiap objek yang berinteraksi dengan sistem/perangkat lunak (misalnya, orang, suatu perangkat keras, sistem lain, dan sebagainya) merupakan *actor*.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*. *Use case* mendeskripsikan secara lengkap tentang bagaimana sistem/perangkat lunak berperilaku untuk para *actor*-nya.

Simbol-simbol yang ada pada diagram *use case* sebagai berikut:

Tabel 5. Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i>; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p>Aktor / <i>actor</i></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 <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i>.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antar <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i></p>

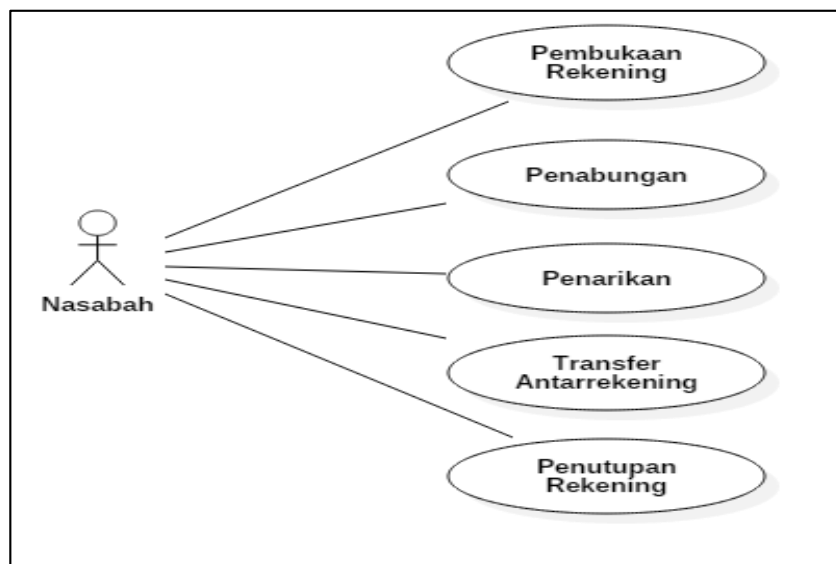
Simbol	Deskripsi
	dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / <i>include</i> / <i>uses</i> <pre><<include>></pre> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber : Rosa A.S. & M. Shalahuddin (2016)

Sebagai contoh untuk diagram *use case* dengan kasus ATM, *actor* dalam kasus tersebut yaitu nasabah kemudian apa saja yang nasabah lakukan saat berinteraksi dengan aplikasi perbankan. Perilaku-perilaku (*behaviour*) *actor* nasabah seperti berikut:

1. Nasabah membuka rekeningnya.
2. Nasabah memeriksa saldo.
3. Nasabah menyimpan uang pada rekeningnya.
4. Nasabah menarik uang dari rekeningnya.
5. Nasabah melakukan *transfer* uang antar rekening.
6. Nasabah menutup rekeningnya.

Sehingga diagram *use case* dari kasus ATM adalah sebagai berikut:



Gambar 4. Contoh Diagram *Use Case* (Sumber: Nugroho, 2009)

6.9.2 Activity Diagram



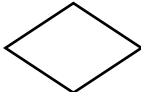


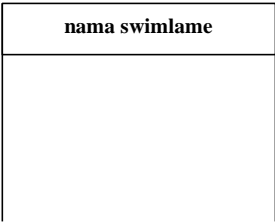
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak

(Sukanto & Shalahuddin, 2016). Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity* diagram digunakan untuk mendefinisikan hal-hal berikut :

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

Simbol-simbol yang ada pada *activity diagram* sebagai berikut:

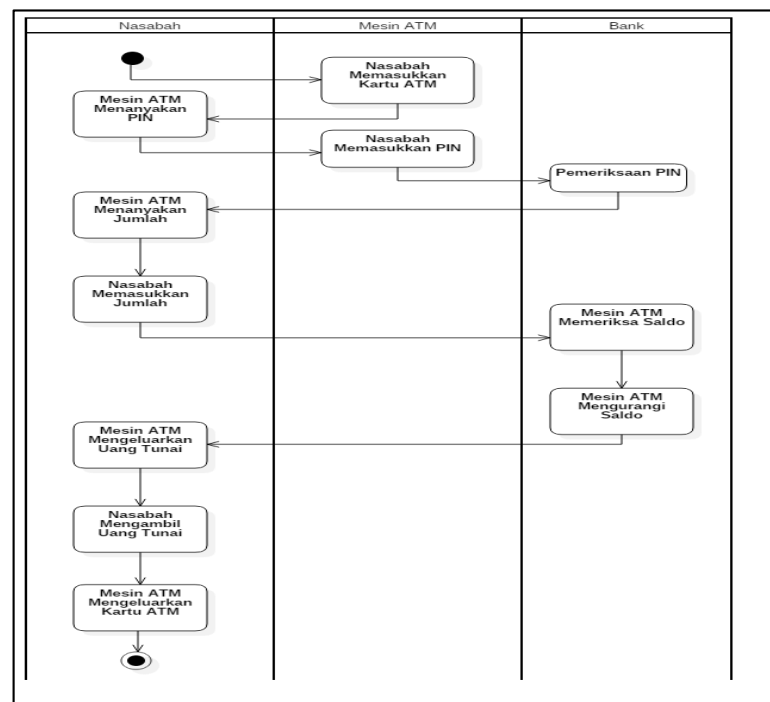
Tabel 6. Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
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

Simbol	Deskripsi
atau	
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; transform: rotate(-90deg); transform-origin: left top;">nama swiname</div> <div style="border: 1px solid black; width: 100px; height: 50px; margin-left: 10px;"></div> </div>	

Sumber : Rosa A.S. & M. Shalahuddin (2016)

Pada pembuatan skenario untuk diagram *activity* dari kasus ATM, sebagai contoh mengambil *use case* penarikan uang. Penarikan uang pada mesin ATM antara nasabah, mesin ATM dan bank terdapat interaksi proses interaksi tersebut digambarkan pada diagram *activity* penarikan uang. Diagram *activity* penarikan uang sebagai berikut:

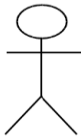

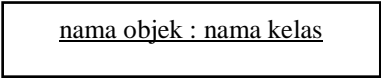

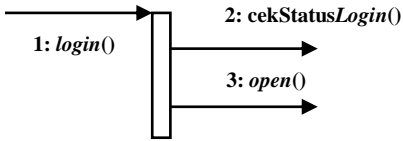
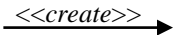
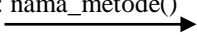
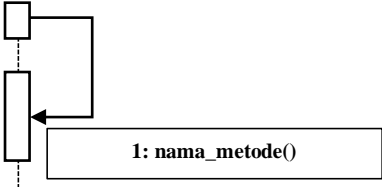


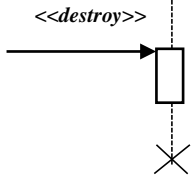
Gambar 5. Contoh Diagram *Activity* (Sumber: Nugroho, 2009)

6.9.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Sukamto & Shalahuddin, 2016). Simbol-simbol yang ada pada *sequence diagram* sebagai berikut:

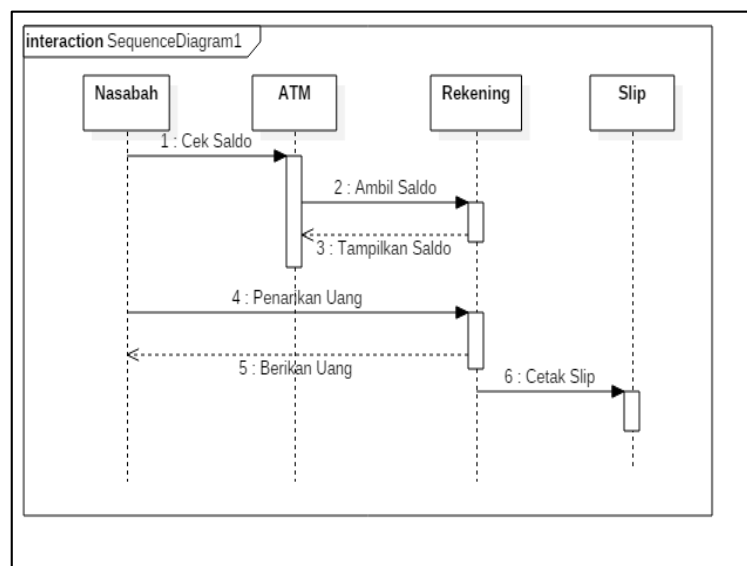
Tabel 7. Simbol-simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i>.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Meyatakan 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 didalamnya, misalnya</p>  <p>maka <i>cekStatusLogin()</i> dan <i>open()</i> dilakukan didalam metode <i>login()</i> <i>Actor</i> tidak memiliki waktu aktif.</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> <p>1: nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p> 

Simbol	Deskripsi
	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.
Pesan tipe <i>send</i> 1: masukan →	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe <i>return</i> 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 <i>destroy</i> → <i><<destroy>></i> 	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 A.S. & M. Shalahuddin (2016)

Sebagai contoh pada kasus ATM nasabah melakukan cek saldo dan melakukan penarikan uang dari ATM. Nasabah dapat melakukan cek saldo dari objek ATM, Nasabah dapat melakukan penarikan uang dan berikan uang dari objek rekening. Proses tersebut dapat digambarkan dalam bentuk diagram *sequence* sebagai berikut:



Gambar 6. Contoh Diagram *Sequence* (Sumber: Nugroho, 2009)

6.9.4 Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Sukamto & Shalahuddin, 2016). Pada diagram kelas memiliki atribut dan metode atau operasi, atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Tujuan dari pembuatan diagram kelas agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan yang ada pada diagram kelas sehingga antar dokumentasi perancangan dan perangkat lunak sinkron. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

a. Kelas *main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

b. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

c. Kelas yang diambil dari pendefinisian *use case* (*controller*)

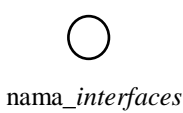

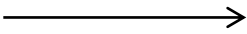
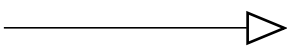
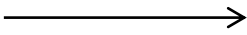
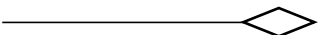
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

d. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut *multivalue* pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan aksesnya dapat dipertanggung jawabkan atau tetap ada didalam perancangan kelas. Simbol-simbol pada diagram kelas sebagai berikut:

Tabel 8. Simbol-simbol Diagram Kelas

Simbol	Deskripsi			
Kelas <table><tr><td>Nama_kelas</td></tr><tr><td>+atribut</td></tr><tr><td>+operasi()</td></tr></table>	Nama_kelas	+atribut	+operasi()	Kelas pada struktur sistem.
Nama_kelas				
+atribut				
+operasi()				

Simbol	Deskripsi
<p>Antarmuka / <i>interfaces</i></p> 	Sama dengan konsep <i>interfaces</i> dalam pemrograman berorientasi objek.
<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan / <i>dependency</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Agregasi / <i>agregation</i></p> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber : Rosa A.S. & M. Shalahuddin (2016)

6.10 Bahasa Pemrograman

Bahasa pemrograman adalah bahasa yang dapat dipahami oleh komputer yang dituliskan dalam bentuk kode-kode. Pada setiap bahasa pemrograman memiliki fungsi yang berbeda-beda seperti bahasa pemrograman untuk aplikasi *desktop*, *mobile*, *web*, dan untuk membuat *game*. Pada pembuatan aplikasi berbasis *web* ada banyak bahasa pemrograman yang dapat digunakan seperti HTML, CSS, PHP, *Javascript*, *JQuery* dan lainnya.

6.10.1 *HyperText Markup Language* (HTML)

HTML adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi seperti gambar, teks, video, dan suara pada penjelajah *web internet*, yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan yang terintegrasi (Sulistiono, 2018).

HyperText Markup Language adalah bahasa pemrograman yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web* (Sibero, 2013). Tugas utama dari bahasa HTML dalam membangun *web* yaitu menentukan *layout*

website, memformat teks dasar seperti pengaturan dan format *font*, membuat *list*, membuat tabel, menyisipkan gambar, video dan audio, membuat *link*, dan membuat formulir. Struktur dokumen HTML sebagai berikut:

```
<html>
  <head> <title> </title>
  </head>
<body></body>
</html>
```

Gambar 7. Struktur Dokumen HTML (Sumber: Sibero, 2013)

Contoh dari *source code* HTML sebagai berikut:

```
<html>
  <head>
    <title> Judul Dokumen </title>
  </head>
  <body>
    <p> Membuat HTML Sederhana </p>
  </body></html>
```

Gambar 8. Contoh *Source Code* HTML (Sumber: Sibero, 2013)

6.10.2 *Cascading Style Sheets* (CSS)

CSS merupakan aturan mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam digunakan untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML (Sulistiono, 2018).

Cascading Style Sheets memiliki arti Gaya Menata Halaman Bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak dan telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya (Sibero, 2013). Contoh kode CSS sebagai berikut:

```
<style type="text/css">
  body{ background-color: #EDED;
    color: #999;
  }
  .center {
    Position: fixed;
    Top: 50%;
    Left: 50%;
    Margin-top: -50px;
    Margin-left: -100px;
  }
</style>
```

Gambar 9. Contoh Kode CSS (Sumber: Sibero, 2013)

6.10.3 Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah Bahasa pemrograman yang digunakan untuk membuat *website* atau situs dinamis dan menangani rangkaian bahasa pemrograman antara *client side scripting* dan *server side scripting* (Sulistiono, 2018).

PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan (Sibero, 2013). PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses pada sisi *server*. Pengelolaan data mulai dari tambah, ubah, hapus dan tampil ke halaman *web* diatur oleh bahasa pemrograman PHP. Berikut adalah contoh penulisan kode PHP:

```
<?php
echo "<html>";
echo "<head>";
echo "<title> Contoh PHP </title>";
echo "</head>";
echo "<body>";
echo "<p>Dibawah ini adalah tulisan dari PHP</p>";
echo "Teks dari PHP";
echo "</body>";
echo "</html>";
?>
```

Gambar 10. Contoh Kode PHP (Sumber: Sibero, 2013)

6.10.4 JQuery

JQuery adalah sebuah *framework* berbasiskan *javascript* atau *javascript library*, yaitu kumpulan kode atau fungsi *javascript* siap pakai sehingga mempermudah dan mempercepat kita dalam membuat kode *javascript* (Sulistiono, 2018).

JQuery adalah salah satu *javascript framework* yang dikembangkan oleh John Resig pada tahun 2006 di BarCamp NYC (Sibero, 2013). *JQuery* memiliki ciri khas pada penggunaan perintahnya, *prefix* untuk *JQuery* dengan tanda \$ kemudian dilanjutkan dengan fungsi atau perintah. Berikut adalah contoh kode dari *JQuery*:

```

<head>
  <script src="jquery-1.5.1.min.js" type="text/javascript"></script>
</head>
<body>
  <input type = "text"/>
  <input type = "checkbox"/>
  <button></button>
  <script type = "text/javascript">
    $(":button").html("ini adalah button");
  </script>
</body>

```

Gambar 11. Contoh Kode *jQuery* (Sumber: Sibero, 2013)

6.10.5 AJAX

AJAX adalah mekanisme komunikasi antara *javascript* yang berada disisi *client* dengan Bahasa disisi *server* seperti PHP dan lainnya (Sibero, 2013). Cara kerja dari AJAX sendiri yaitu menjalankan suatu alamat perintah pada *server* dan menerima data yang dikembalikan oleh *server*. Berikut adalah contoh kode AJAX:

```

<script type = "text/javascript">
$.ajax(){
  url: "contoh.html",
  dataType: "html",
  beforeSend: function(){
    alert ('Sebelum perintah dijalankan');},
  error: function (data){
    alert ('ditemukan kesalahan' +
error.msg);},
  success: function (data){
    $("konten").html (data);},
  complete: function (){
    alert ('Proses selesai')}}
}
</script>

```

Gambar 12. Contoh Kode AJAX(Sumber: Sibero, 2013)

6.11 Framework *CodeIgniter*

Web Application Framework (WAF), atau sering disingkat *web framework* adalah kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan menjadi suatu kerangka kerja (*framework*) berguna untuk memudahkan dan mempercepat proses pengembangan aplikasi *web* (Raharjo, 2018).

Framework web banyak bermunculan yang dirancang untuk bahasa-bahasa pemrograman PHP, *Python*, *Ruby*, *Perl*, *C++*, *Java* dan lainnya. Jenis *framework* untuk bahasa pemrograman PHP yaitu *CodeIgniter*, *Yii*, *Slim Framework*, *Zend Framework*, *Laravel*, *Symfony*, *CakePHP*, *Phalcon*, *Kohana*, *FuelPHP* dan lain-

lain. *CodeIgniter* adalah sebuah aplikasi *open source* yang berupa kerangka kerja atau *framework* untuk membangun *website* menggunakan bahasa pemrograman PHP (Sulistiono, 2018). Tujuannya memungkinkan pengembangan proyek yang lebih cepat daripada penulisan kode dasar atau kode terstruktur, dengan menyediakan banyak *library* yang biasanya digunakan dalam pengerjaannya. *CodeIgniter* ditulis atau dibuat oleh Ellis Lab dan dirilis pertama kali pada 28 Februari 2006.

Pada tanggal 9 Juli 2013, Ellis Lab mengumumkan bahwa mereka mencari pemilik baru untuk *CodeIgniter*, dengan alasan kurangnya sumber daya untuk memberi kerangka perhatian yang mereka rasa pantas. Pada tanggal 6 Oktober 2014, Ellis Lab mengumumkan bahwa *CodeIgniter* akan melanjutkan pengembangan dibawah pengelolaan *British Columbia Institute of Technology*, dengan versi stabil terakhir adalah versi 3.1.6 dirilis pada tanggal 25 September 2017. *Framework CodeIgniter* mempunyai arsitektur MVC (*Model, View, Controller*) dan juga HMVC (*Hierarkis Model, View, Controller*).

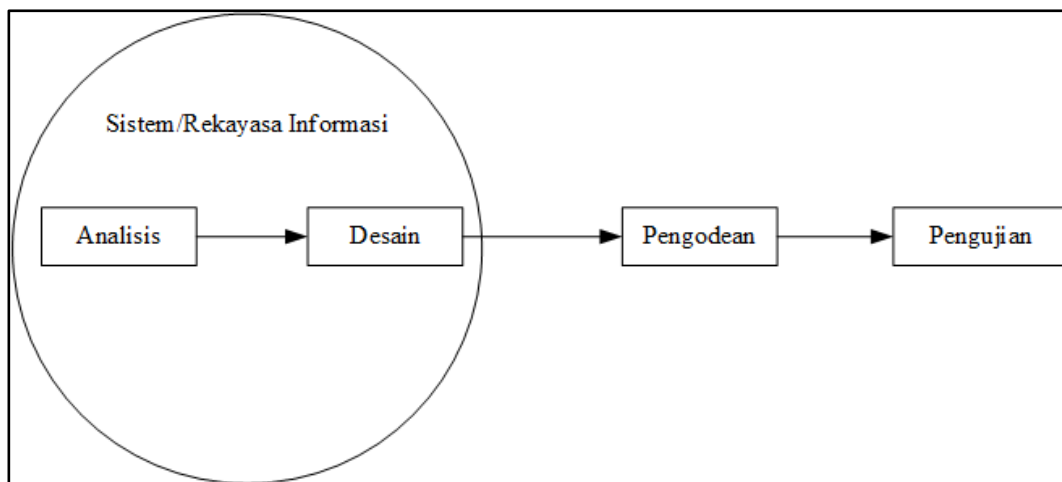
Arsitektur MVC adalah pemisahan tampilan pengguna (*view*), kendali (*control*) dan model informasi sehingga pada sebuah sistem antara komponen presentasi sistem dengan komponen logika bisnisnya dapat lebih mudah dipahami dan dipelihara (Nugroho, 2009).

Sedangkan untuk HMVC merupakan turunan dari arsitektur MVC yaitu membagi suatu sistem menjadi bagian-bagian MVC yang terpisah dan kemudian masing-masing MVC bertanggung jawab pada bagian masing-masing (Nugroho, 2009). Penggunaan arsitektur HMVC pada pengembangan sebuah sistem memberikan keuntungan dimana standarisasi struktur kode lebih mudah dipahami oleh para pengembang. Selain itu, penyederhanaan pemeliharaan kode sehingga setiap komponen dapat dikembangkan dan diperbaharui secara terpisah.

6.12 Software Development Life Cycle (SDLC)

Software Development Life Cycle adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) (Sukamto & Shalahuddin, 2016).

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya. Model SDLC yang digunakan penulis dalam perancangan dan pembangunan Sistem Pendukung Keputusan Penentuan Calon Penerima Beasiswa Bidikmisi dan PPA adalah model *waterfall*. Model SDLC air terjun (*waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). Berikut adalah ilustrasi model *waterfall*:



Gambar 13. Ilustrasi Metode *Waterfall* (Sumber : Rosa A.S. & M. Shalahuddin 2016)

Keterangan :

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses *multi* langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.

3. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasilnya program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Tahap pengujian dilakukan

untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Sebuah perangkat lunak yang sudah sampai ke *user* kemungkinan terjadi kesalahan yang tidak terdeteksi pada saat pengujian perangkat lunak. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

6.13 Blackbox Testing (Pengujian Kotak Hitam)

Pengujian kotak hitam yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program (Sukanto & Shalahuddin, 2016). Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalnya untuk kasus login maka kasus uji yang dibuat adalah :

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

Klasifikasi *black box testing* mencakup beberapa pengujian, yaitu:

1. Pengujian Fungsional (*functional testing*)
2. Pengujian Tegangan (*stress testing*)
3. Pengujian Beban (*load testing*)
4. Pengujian Khusus (*ad-hoc testing*)
5. Pengujian Penyelidikan (*exploratory testing*)
6. Pengujian Usabilitas (*usability testing*)
7. Pengujian Asap (*smoke testing*)
8. Pengujian Pemulihan (*recovery testing*)
9. Pengujian Volume (*volume testing*)
10. Pengujian Domain (*domain testing*)

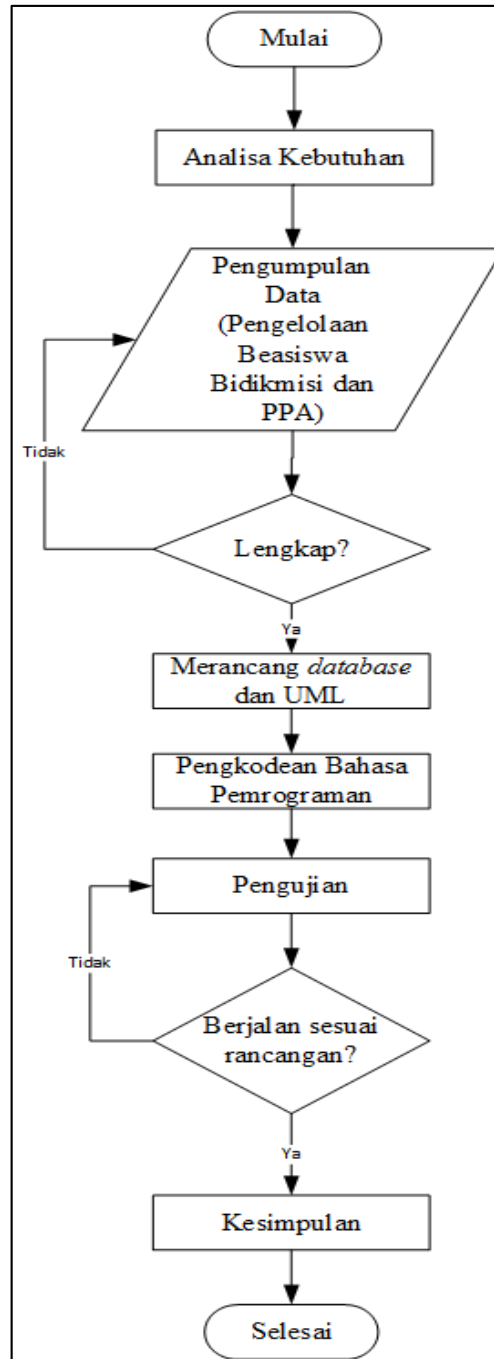
11. Pengujian Skenario (*scenario testing*)
12. Pengujian Regresi (*regression testing*)
13. Penerimaan Pengguna (*user acceptance*)
14. Pengujian Alfa (*alpha testing*)
15. Pengujian Beta (*beta testing*)

Pengujian fungsional dilakukan pada perangkat lunak untuk persyaratan fungsional apakah aplikasi berjalan sesuai dengan yang diharapkan (Simarmata, 2010). Tahap pengujian jenis ini meliputi seberapa baik sistem melaksanakan fungsinya, termasuk perintah-perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar dan integrasi. Pengujian ini dapat dilakukan pada akhir dari siklus pengembangan maupun pada awal pengembangan terhadap masing-masing komponen dan proses yang ada pada sistem.

7. METODE PENELITIAN

Tahapan-tahapan yang dilakukan pada penelitian ini dapat dilihat pada gambar 14. sebagai berikut:

- a. Melakukan analisa kebutuhan untuk membangun Sistem Pendukung Keputusan Penentuan Calon Penerima Beasiswa Bidikmisi dan PPA.
- b. Melakukan pengumpulan data pengelolaan beasiswa Bidikmisi dan PPA, dimana tempat studi kasusnya yaitu Politeknik Negeri Tanah Laut.
- c. Kemudian apabila data lengkap maka tahapan selanjutnya yaitu merancang *database* dan UML. Apabila data belum lengkap maka mencari data tersebut, sehingga dapat melakukan tahap selanjutnya.
- d. Setelah perancangan *database* dan UML, maka tahap selanjutnya yaitu proses pengkodean program berbasis *web*. Melakukan pengkodean harus sesuai dengan rancangan *database* dan UML.
- e. Tahap selanjutnya yaitu melakukan pengujian terhadap sistem. Apabila belum sesuai maka kembali ke tahap pengkodean program untuk memperbaiki sesuai dengan hasil luaran kesalahan telah dilakukan dalam pengujian.
- f. Tahapan terakhir yaitu kesimpulan sesuai dengan perencanaan yang telah dibuat.



Gambar 14. Alur Penelitian

7.1 Pengumpulan Data

Tahapan pengumpulan data dilakukan dengan dua metode yaitu studi pustaka dan wawancara.

1. Studi Pustaka

Studi pustaka yang dilakukan oleh penulis yaitu mempelajari literatur-literatur yang berhubungan dengan topik penelitian yaitu tentang sistem pendukung

keputusan penentuan calon penerima beasiswa yang diperoleh dari beberapa sumber antara lain jurnal ilmiah, laporan hasil penelitian dan buku.

2. Wawancara

Wawancara yang dilakukan oleh penulis yaitu melakukan tanya jawab dengan pengelola beasiswa pada bagian kemahasiswaan yang ada di Perguruan Tinggi Politeknik Negeri Tanah Laut.

7.2 Rancangan Penelitian

Rancangan penelitian untuk membangun sistem tersebut adalah sebagai berikut:

1. Desain Sistem

Tahap desain sistem yaitu tahap penentuan proses-proses bisnis yang dilakukan oleh pengelola beasiswa dan para calon penerima beasiswa dengan data-data persyaratan pengajuan beasiswa Bidikmisi dan PPA. Kemudian melakukan identifikasi hubungan pengelola beasiswa dan calon penerima beasiswa terhadap data persyaratan pengajuan beasiswa Bidikmisi dan PPA.

2. Desain Arsitektur

Setelah tahap desain sistem tersebut kemudian dilanjutkan tahap desain arsitektur proses data antara pengelola beasiswa dan para calon penerima beasiswa. Proses komunikasi tersebut meliputi proses pengiriman dan penerimaan data, proses penambahan dan pengurangan kriteria penilaian serta proses penelaian untuk menentukan calon penerima beasiswa.

3. Desain *Database*

Tahap selanjutnya yaitu tahap desain *database* untuk penyimpanan data yang dikelola oleh sistem yang meliputi data persyaratan pengajuan beasiswa, data kriteria penilaian dan data hasil penilaian. Desain *database* menggambarkan desain sistem dan arsitektur yang akan dibangun. Desain *database* yang dirancang pada penelitian ini yaitu *Entity Relationship Diagram* (ERD).

4. Desain *User Interfaces*

Desain *user interfaces* merupakan rancangan tampilan sistem yang akan digunakan oleh pengguna yang memberikan kenyamanan pengguna dalam menggunakan sistem. Desain *user interfaces* pada penelitian ini menggunakan aplikasi *JustInMind*.

7.3 Rancangan Pengujian

Tahap pengujian dilakukan setelah sistem selesai dibangun untuk dilakukan tahap pengujian aplikasi dan analisis. Pada penelitian ini pengujian sistem menggunakan metode *blackbox* dimana pengujian yang dilakukan adalah pengujian fungsional. Pengujian fungsional dilakukan untuk mengetahui fungsional dari sistem apakah berjalan sesuai dengan yang diharapkan atau tidak.

8. JADWAL PENELITIAN

Jadwal penelitian dijabarkan pada tabel 9. sebagai berikut:

Tabel 9. Jadwal Penelitian

No.	Kegiatan	Target Pencapaian	Bulan									
			Feb		Mar		Apr		Mei		Jun	
			1- 2	3- 4	1- 2	3- 4	1- 2	3- 4	1- 2	3- 4	1- 2	3- 4
1.	Penyusunan Proposal	Proposal siap disidangkan										
2.	Pengumpulan Data	Terkumpulnya data penentuan calon penerima beasiswa bidikmisi dan PPA										
3.	Analisis dan Perancangan	Desain proses bisnis, desain <i>database</i> , desain UI										
4.	Implementasi	Sistem Pendukung Keputusan Penentuan Calon Penerima Beasiswa Bidikmisi dan PPA										
5.	Pengujian dan Analisa	Pengujian sistem dengan nilai <i>error</i> peramalan terkecil										
6.	Penyusunan Laporan	Laporan Siap disidangkan dan revisi										

DAFTAR PUSTAKA

- Anhar, 2010. *Panduan Menguasai PHP & MySQL Secara Otodidak*. 1 ed. Jakarta: Penerbit PT. TransMedia.
- Arief, M. R., 2011. *Pemrograman Web Dinamis Menggunakan PHP dan MySQL*. I ed. Yogyakarta: Penerbit Andi.
- Chandra, B. & Ocnota, G. G., 2018. *Studi Kasus Sistem Penunjang Keputusan Metode SAW dan TOPSIS*. Malang: CV. Seribu Bintang.
- Diana, 2018. *Metode dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta: Penerbit Deepublish.
- Kemenristekdikti, 2014. *Pedoman Penyelenggaraan Bantuan Biaya Pendidikan Bidikmisi Tahun 2014*. Jakarta: Kemenristekdikti.
- Kemenristekdikti, 2018. *Panduan Beasiswa Peningkatan Prestasi Akademik (PPA) Tahun 2018*. Jakarta: Kemenristekdikti.
- Kemenristekdikti, 2019. *Panduan Pendaftaran Beasiswa Bidikmisi 2019*. Jakarta: Ditjen Belmawa Kemenristekdikti.
- Nofriansyah, D., 2014. *Konsep Data Mining Vs Sistem Pendukung Keputusan*. I ed. Yogyakarta: Penerbit Deepublish.
- Nugroho, A., 2009. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Yogyakarta: Penerbit Andi.
- Peraturan Pemerintah Republik Indonesia Nomor 48 Tahun 2008, 4 Juli 2008. *Pendanaan Pendidikan*. Jakarta, s.n.
- Peraturan Pemerintah Republik Indonesia Nomor 66 Tahun 2010, 28 September 2010. *Perubahan Atas Peraturan Pemerintah Nomor 17 Tahun 2010 Tentang Pengelolaan dan Penyelenggaraan Pendidikan*. Jakarta, s.n.
- Priyadi, Y., 2014. *Kolaborasi SQL dan ERD dalam Implementasi Database*. Yogyakarta: Penerbit Andi.
- Puspitasari, D., 2016. Sistem Informasi Perpustakaan Sekolah. *Jurnal Pilar Nusa Mandiri*, Volume XII, p. 229.
- Raharjo, B., 2018. *Belajar Otodidak Framework CodeIgniter*. I ed. Bandung: Penerbit Informatika.
- Sari, F., 2018. *Metode Dalam Pengambilan Keputusan*. Yogyakarta: Penerbit Deepublish.
- Sibero, A. F., 2013. *Web Programming Power Pack*. Pertama ed. Jakarta: Penerbit MediaKom.
- Simarmata, J., 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Penerbit Andi.
- Sukanto, R. A. & Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. 4 ed. Bandung: Penerbit Informatika.

Sulistiono, H., 2018. *Coding Mudah dengan CodeIgniter, JQUERY, Bootstrap, dan Datatable*. Jakarta: PT Elex Media Komputindo.

Undang-Undang Republik Indonesia Nomor 12 Tahun 2012, 10 Agustus 2012. *Pendidikan Tinggi*. Jakarta, s.n.

Undang-Undang Republik Indonesia Nomor 20 Tahun 2003, 8 Juli 2003. *Sistem Pendidikan Nasional*. Jakarta, s.n.