

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penelitian terkait ini, penulis mengkaji penelitian terdahulu mengenai sistem aplikasi yang dilakukan oleh penelitian-penelitian sebelumnya yang dapat menjadi dasar dari penelitian yang dilakukan. Beberapa penelitian mengenai sistem pencarian rumah kost diantaranya adalah sebagai berikut:

(Rezalendra, 2019) melakukan penelitian dengan judul “Aplikasi Persebaran Rumah Kost Berbasis *Android* (Studi Kasus: Kota Malang)”. Dalam penelitian tersebut mereka merancang aplikasi ‘Pencarian Rumah Kost’ yang berbasis *Android* menyediakan informasi tentang lokasi rumah kost dan fasilitas kost. Informasi yang tersaji di aplikasi mulai nama rumah kost, nama pemilik kost, alamat rumah kost, jarak radius rumah kost dengan pengguna, jenis kost, dan fasilitas kost.

(Abidin, 2012) melakukan penelitian dengan judul ”Sistem Informasi Rumah Kost *Online* Berbasis *Web* dan *Messaging*”. Aplikasi yang dibangun merupakan aplikasi berbasis *web* dan *messaging*. Pembuatan aplikasi pencarian informasi rumah kost terdekat ini bertujuan untuk membantu mahasiswa perantau untuk mengetahui lokasi rumah kost yang ada di kota tempat studi mereka. Sistem yang digunakan pada aplikasi informasi pencarian lokasi rumah kost ini di titik beratkan pada pengambilan koordinat GPS pengguna. Kemudian untuk mendapatkan peta dan rute, sistem menggunakan *google map API* dengan memberikan parameter koordinat bumi. Setelah mengirim parameter tersebut ke *google maps*, maka *google maps server* akan membalas berupa peta statik. Dalam pencarian rute, sistem mengirimkan dua koordinat bumi sebagai alamat awal dan alamat tujuan, kemudian *google maps* akan membalas berupa data rute yang akan ditampilkan pada pengguna. Dalam pencarian lokasi rumah kost, sistem terhubung ke *server database* dengan memasukkan titik koordinat *latitude* dan *longitude* dari semua lokasi. Setelah itu *server* akan membalas berupa data lokasi-lokasi rumah kost yang selanjutnya akan di *parsing* oleh sistem dan ditampilkan pada pengguna. Tujuan dari *parsing* data rumah kost agar

penguraian data rumah kost yang dipilih menjadi lebih khusus dalam memudahkan pengguna untuk menentukan rumah kost di *google maps*.

Dari kedua aplikasi penelitian diatas terdapat perbedaan pada penelitian (Rezalendra, 2019) yang berbasis *Android* dan penelitian yang dilakukan oleh (Abidin, 2012) berbasis *web* dan *messaging*. Pengambilan data rumah kost untuk kedua penelitian diambil menggunakan layanan data *google maps API*. Selain berbeda basis dari penelitian yang dilakukan, juga terdapat perbedaan dalam memasarkan rumah kost dan update data. Pada penelitian yang dilakukan Rezalendra, Silvester, & Adkha. (2019) hanya mengandalkan data yang disediakan di *google maps* sehingga data terbatas dan masih banyak rumah kost yang masih belum terdaftar. Penelitian yang dilakukan oleh (Abidin, 2012) mempunyai fitur *rules* untuk pemilik rumah kost dan *rules* pencarian rumah kost sehingga manajemen pada aplikasi teratur dengan baik. Pada *rules* manajemen aplikasi juga memiliki fitur pemasaran sehingga pemilik rumah kost bisa mendaftarkan rumah kostnya agar bisa diiklankan. Kekurangan dari kedua penelitian ini untuk manajemen sistem aplikasi dalam pemasaran masih kurang efektif, penelitian yang dilakukan oleh (Rezalendra, 2019) data rumah kost yang masih terbatas hanya mengambil data dari *google maps* saja yang mana banyak pemilik rumah kost yang belum mendaftarkan rumah kost nya pada *google maps*. Sedangkan kekurangan dari penelitian yang dilakukan oleh Abidin, Prasetyaningrum, & Karlita. (2012) juga dalam manajemen pemasaran rumah kost yang memiliki masa aktif yang terbatas dalam update data dan pemasaran sehingga harus mengeluarkan biaya untuk aktivasi ulang.

(Hayati, 2019) melakukan penelitian dengan judul “Penerapan *Metode Simple Additive Weighting* dalam Pemilihan Rumah Kost”. Penelitian ini memberikan solusi mengenai permasalahan yang terjadi dalam menentukan pilihan rumah kost yang sesuai dengan kriteria. Untuk memecahkan masalah memilih rumah kost yang ideal untuk pengguna yang mementingkan berbagai kriteria dapat ditentukan dengan mudah sistem pendukung keputusan yaitu dengan metode *simple additive weighting*. Masyarakat dapat mencari kost dan menentukan pilihan kost yang ada di kota Medan. Penelitian ini menghasilkan sebuah proses pencarian rumah kost menjadi lebih efisien dan mempercepat

proses penyelesaian perhitungan pemilihan rumah kost di kota Medan, dan yang dapat diimplementasikan sesuai dengan kebutuhan masyarakat, terutama dalam menentukan keputusan.

(Asniadi, 2019) melakukan penelitian dengan judul “Sistem Pendukung Keputusan Pemilihan Kost di Sekitar Kampus Unmuh Jember Menggunakan Metode *Simple Additive Weighting* (SAW)”. Penelitian ini bertujuan mengimplementasikan metode *simple additive weighting* (SAW) untuk menentukan pilihan tempat kost di sekitar kampus Universitas Muhammadiyah Jember. Penelitian yang dilakukan (Asniadi, 2019) menjadi acuan penelitian dalam memberikan solusi dalam menentukan pilihan rumah kost yang sesuai dengan kriteria.

Penelitian yang dilakukan oleh (Hayati, 2019), serta penelitian (Asniadi, 2019) ini akan menjadi acuan penulis dalam menentukan kriteria penting untuk mencari rumah kost. Oleh karena itu penelitian terkait yang sudah dipaparkan diatas, maka pada penelitian ini penulis akan membangun sebuah aplikasi pencarian rumah kost di Kota Pontianak menggunakan metode *simple additive weighting* (SAW) berbasis *progressive web app*. Perbedaan penelitian ini dengan penelitian-penelitian sebelumnya adalah aplikasi yang dibuat penulis menggunakan *progressive web app* sebagai basis aplikasi, yang bisa di muat dalam bentuk *website* maupun *mobile*. Selain menggunakan PWA aplikasi ini berbeda dari cara pengambilan data, yaitu melalui data data diambil dari *google maps* yang akurat dan *update*, data diambil menggunakan layanan *google my business*, sehingga manajemen data dan pemasaran rumah kost selalu di *maintenance* sehingga aplikasi dan data rumah kost bisa dipakai terus menerus. Selain itu terdapat fitur *rating* dan ulasan, serta memiliki fitur filter kriteria terkait rumah kost yang tidak terdapat pada penelitian sebelumnya, dan tidak mengenakan tarif dalam mengiklankan rumah kost.

(Sugianto, 2016), melakukan penelitian dengan judul “ Sistem Pendukung Keputusan Pemelihan Tempat Kost Khusus Mahasiswa dengan Metode AHP dan TOPSIS Berbasis *Web* (Studi Kasus: Kota Pontianak)”. Penelitian ini bertujuan untuk membantu mahasiswa dalam memilih tempat kost yang tepat sesuai dengan kriteria yang diinginkan oleh mahasiswa tersebut.

Penelitian yang dilakukan oleh Herik Sugianto. (2016), juga menjadi acuan penelitian dalam menggunakan metode sistem pendukung keputusan (SPK) serta mengambil referensi kriteria yang akan digunakan dalam menentukan hasil menentukan keputusan pada pemilihan rumah kost.

Berikut adalah hal-hal yang membedakan setiap kajian terkait yang dijelaskan pada tabel 2.1 berikut ini.

Tabel 2.1 Perbandingan kajian terkait

No	Penulis	Judul	Keterangan
1.	(Rezalendra, 2019) Institut Teknologi Nasional Malang	Aplikasi Persebaran Rumah Kos Berbasis Android (Studi Kasus: Kota Malang)	1. Berbasis <i>Web</i> 2. Aplikasi ‘Pencarian Rumah Kost’ yang berbasis Android menyediakan informasi tentang lokasi rumah kost dan fasilitas kost. 3. Informasi yang tersaji di aplikasi mulai nama rumah kost, nama pemilik kost, alamat rumah kost, jarak radius rumah kost dengan pengguna, jenis kost, dan fasilitas kost.
2.	(Abidin, 2012) Politeknik Elektronika Negeri Surabaya	Sistem Informasi Rumah Kost <i>Online</i> Berbasis <i>Web</i> dan Messaging	1. Berbasis <i>Web</i> 2. Sistem yang digunakan pada aplikasi informasi pencarian lokasi rumah kost ini di

No	Penulis	Judul	Keterangan
			<p>titik beratkan pada pengambilan koordinat GPS pengguna.</p> <p>3. Sistem menggunakan <i>google map</i> API dengan memberikan parameter koordinat bumi.</p> <p>4. Dalam pencarian lokasi rumah kost, sistem terhubung ke <i>server database</i> dengan memasukkan titik koordinat latitude dan longitude dari semua lokasi.</p>
3.	(Hayati, 2019) Universitas Potensi Utama	Penerapan Metode <i>Simple Additive Weighting</i> dalam Pemilihan Rumah Kost	<p>1. Berbasis <i>Web</i></p> <p>2. Penelitian ini memberikan solusi mengenai permasalahan yang terjadi dalam menentukan pilihan rumah kost yang sesuai dengan kriteria.</p>

No	Penulis	Judul	Keterangan
			3. Pengguna yang mementingkan berbagai kriteria dapat ditentukan dengan mudah sistem pendukung keputusan yaitu dengan metode <i>simple additive weighting</i> .
4.	(Asniadi, 2019) Universitas Muhammadiyah Jember	Sistem Pendukung Keputusan Pemilihan Kost di Sekitar Kampus Unmuh Jember Menggunakan Metode Simple Additive Weighting (SAW)	1. Aplikasi berbasis <i>web</i> . 2. Model yang digunakan dalam sistem pendukung keputusan ini adalah metode SAW, metode <i>Simple Additive Weighting (SAW)</i> adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut.
5.	(Sugianto, 2016) Universitas Tanjungpura Pontianak	Sistem Pendukung Keputusan Pemilihan Tempat Kost Khusus Mahasiswa	1. Aplikasi berbasis <i>web</i> . 2. Metode yang digunakan dalam sistem ini yaitu

No	Penulis	Judul	Keterangan
		dengan Metode AHP dan TOPSIS Berbasis Web (Studi Kasus: Kota Pontianak)	metode AHP dan TOPSIS. 3. Aplikasi ini bertujuan untuk membantu mahasiswa dalam memilih tempat kost yang tepat sesuai dengan kriteria yang diinginkan oleh mahasiswa tersebut. 4. Menggunakan <i>google maps</i> API untuk menghitung jarak dari kost ke kampus.

Tabel 2.2 Penelitian yang dilakukan

No	Penulis	Judul	Keterangan
1.	Arif Pirmantara Universitas Tanjungpura Pontianak	Aplikasi Pencarian Rumah Kost Pada Kota Pontianak Dengan Menggunakan Metode <i>Simple Additive Weighting</i> (Saw) Yang Berbasis <i>Progressive Web</i>	1. Aplikasi Berbasis <i>Progressive web app</i> . 2. Menggunakan metode <i>simple additive weighting</i> (SAW). 3. Menggunakan GPS sebagai penentuan titik lokasi rumah kost. 4. Aplikasi menampilkan titik-titik lokasi rumah

No	Penulis	Judul	Keterangan
		<p><i>App</i> Studi Kasus: Kota Pontianak</p>	<p>kost area kota pontianak.</p> <p>5. Membantu pemilik rumah kost akan ditambahkan fitur pendaftaran <i>online</i> dengan menggunakan layanan dari <i>google</i> yaitu <i>google my business</i>, pembaruan informasi rumah kost secara <i>update</i>.</p> <p>6. Mahasiswa dan perantau melukan pencarian dengan mengisi kriteria rumah kost yang diinginkan pada aplikasi.</p> <p>7. Mahasiswa dan perantau mendapatkan informasi lengkap seperti lokasi, jarak, harga, fasilitas, <i>contact</i> pemilik rumah kost, serta kriteria kebutuhan dari hasil keputusan pemilihan tempat kost.</p>

2.2 Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System*. Sistem tersebut adalah suatu sistem yang berbasis komputer yang

ditunjukkan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Dewanto, 2015). Istilah SPK mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan. Untuk memberikan pengertian yang lebih maka ada beberapa definisi mengenai SPK oleh beberapa ahli. Menurut Turban, Sistem Pendukung Keputusan (SPK) merupakan sistem informasi yang berbasis komputer yang fleksibel, interaktif dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi untuk masalah manajemen spesifik yang tidak terstruktur. Sistem Pendukung Keputusan menggunakan data, memberikan antarmuka pengguna yang mudah dan dapat menggabungkan pemikiran pengambilan keputusan (Turban, 2011). Menurut Little, Sistem Pendukung Keputusan adalah suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data dan model (Little, 2004). Menurut Kusriani, Sistem Pendukung Keputusan merupakan suatu sistem informasi yang menyediakan informasi, pemodelan dan pemanipulasian data (Kusriani, 2017). Menurut Hermawan, Sistem Pendukung Keputusan didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manajer maupun sekelompok manajer dalam memecahkan masalah semi terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu (Hermawan, 2005).

Sistem Pendukung Keputusan digunakan untuk mendeskripsikan sistem yang didesain untuk membantu manajer memecahkan masalah tertentu (McLeod, 2008). Dari beberapa ahli di atas, dapat disimpulkan bahwa sistem pendukung keputusan merupakan sistem informasi yang mendukung manajemen level menengah dalam mengambil keputusan semiterstruktur dengan menggunakan pemodelan analitis dan data yang ada.

2.3 Metode *Simple Additive Weighting* (SAW)

Metode *Simple Additive Weighting* (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967) (MacCrimmon, 1968). Metode SAW membutuhkan proses

normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode SAW dapat digunakan dalam menghadapi situasi *Multiple Attribute Decision Making* (MADM).

MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Metode SAW ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya.

Kelebihan dari metode SAW adalah yang pertama menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif, penilaian akan lebih tepat karena didasarkan pada nilai kriteria dari bobot preferensi yang sudah ditentukan, dan adanya perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai benefit dan *cost*). Kekurangan dari metode SAW itu sendiri adalah digunakan pada pembobotan lokal yang mana nilai dari pembobotan itu diperoleh dari setiap alternatif yang digunakan, dan menentukan satu persatu nilai tertinggi dari hasil normalisasi dari setiap alternatif.

Langkah-langkah dalam menggunakan metode SAW yang pertama adalah menentukan kriteria – kriteria yang akan dijadikan acuan dalam mengambil keputusan yaitu C_i , menentukan rating kecocokan setiap alternatif pada setiap kriteria, membuat matrik keputusan berdasarkan kriteria (C_i), kemudian melakukan normalisasi matrik berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matrik ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi. Formula untuk melakukan normalisasi tersebut adalah persamaan (2.1):

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad (2.1)$$

Dari rumus diatas dapat dijabarkan sebagai berikut persamaan (2.2):

$$R_{ij} = (X_{ij} / \max\{X_{ij}\}) \text{ Jika } j \text{ adalah atribut keuntungan (benefit)} \quad (2.2)$$

$$R_{ij} = (\min\{X_{ij}\} / X_{ij}) \text{ Jika } j \text{ adalah atribut biaya (cost)}$$

Dimana:

- a. R_{ij} = *rating* kinerja ternormalisasi
- b. Max_{ij} = nilai maksimum dari setiap baris dan kolom
- c. Min_{ij} = nilai minimum dari setiap baris dan kolom
- d. X_{ij} = baris dan kolom dari matrik

Dengan r_{ij} adalah *rating* kinerja ternormalisasi dari alternative A_i pada atribut C_j ; $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$. Nilai preferensi untuk setiap *alternative* (V_i) diberikan sebagai:

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad (2.3)$$

persamaan (2.4) dari rumus nilai preferensi sebagai berikut:

$$V_i = (W_{j1} \times R_{ij1}) + (W_{j2} \times R_{ij2}) \quad (2.4)$$

Dimana :

- a. V_i = nilai preferensi
- b. W_j = bobot *ranking*
- c. R_{ij} = nilai minimum dari setiap baris dan kolom

Dengan r_{ij} adalah *rating* kinerja ternormalisasi dari *alternative* A_i pada atribut C_j ; $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.

Perbedaan dalam menentukan keputusan dari metode SAW, AHP dan TOPSIS terdapat pada langkah-langkah dalam menggunakan metode, yang mana pada langkah-langkah metode AHP sebagai berikut (Kursini, 2017):

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan.

2. Menentukan prioritas elemen

- a. Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.
- b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk mempresentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.

3. Sintesis

Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini yaitu:

- a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
- b. Membagi setiap nilai kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
- c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

4. Mengukur konsistensi

Hal ini penting dilakukan untuk mengetahui seberapa baik konsistensi yang ada berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah mengukur konsistensi adalah:

- a. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.
- b. Jumlahkan setiap baris.
- c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
- d. Jumlahkan hasil bagi di atas dengan banyaknya elemen yang ada, hasilnya disebut maks.

5. menghitung *consistency* indeks

6. Menghitung consistency ratio

Berikut adalah langkah-langkah menggunakan metode *Technique for Order Preference by Similarity to an Ideal Solution* (TOPSIS) (Kursini, 2017):

1. Menentukan matriks keputusan yang ternormalisasi, dimana menggambarkan alternatif (X) dan kriteria (n) ke dalam sebuah matriks
2. Membuat matriks R yaitu matriks keputusan ternormalisasi setiap normalisasi dari nilai rij
3. Membuat pembobotan pada matriks yang telah dinormalisasi.
4. Menentukan nilai solusi ideal positif dan solusi ideal negatif.
5. Kemudian menghitung *Separation Measure*. *Separation Measure* merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif.
6. Menghitung nilai preferensi untuk setiap alternatif.

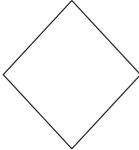
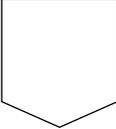
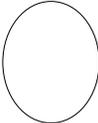
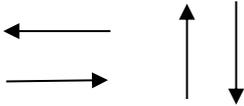
Perbedaan dalam menggunakan metode SPK berpengaruh juga terhadap hasil keputusan. Perbedaan dari metode SAW, AHP dan TOPSIS adalah bagaimana cara menormalisasikan matriks keputusan dan rumus yang digunakan dalam menentukan alternatif, serta hasil yang diperoleh dari setiap alternatif juga berbeda.

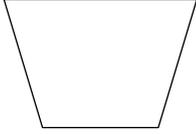
2.4 Flowchart

Menurut Bambang (Hartono, 2011) “*flowchart* merupakan penggambaran dari grafik dari langkah-langkah atau bagian yang memperlihatkan urutan-urutan prosedur dari suatu program dan hubungan antara proses beserta bentuknya”. Gambar *flowchart* dinyatakan dengan simbol, dengan demikian setiap simbol menggambarkan proses tertentu, sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* menolong analisis dari programmer untuk memecahkan masalah kedalam segmen-segmen lebih kecil dan dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Simbol – simbol yang digunakan dalam *flowchart* adalah sebagai berikut :

Tabel 2.3 Simbol-Simbol Dalam *Flowchart*

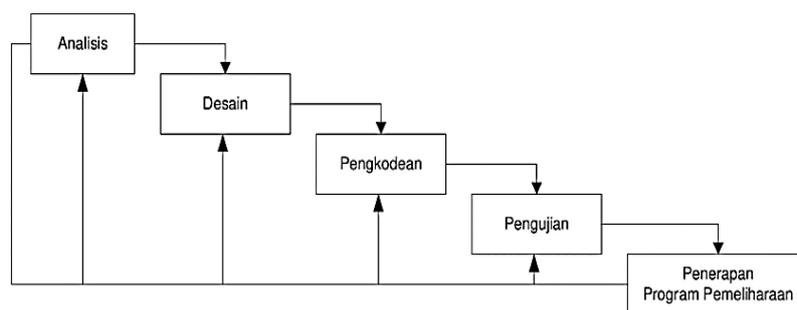
No	Simbol	Keterangan
1		Terminator Simbol yang menyatakan awal atau akhir program

No	Simbol	Keterangan
2		<p><i>Input/output</i> Simbol yang menyatakan proses input atau output tanpa tergantung peralatan</p>
3		<p><i>Decision</i> Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak</p>
4		<p><i>Off-page Reference</i> Simbol untuk keluar – masuk atau penyambungan proses dalam lembar kerja yang berbeda</p>
5		<p><i>On-Page Reference</i> Simbol untuk keluar – masuk atau penyambungan proses dalam lembar kerja yang sama</p>
6		<p><i>Flow</i> Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line</p>
7		<p><i>Process</i> Simbol yang menyatakan suatu proses yang dilakukan komputer</p>
8		<p><i>Display</i> Simbol yang menyatakan peralatan output yang digunakan</p>

No	Simbol	Keterangan
9		Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer
10		Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak
11		Predefine Process Simbol untuk pelaksanaan suatu bagian (<i>sub-program</i>) atau <i>procedure</i>

2.5 Model Waterfall

Terdapat beberapa metodologi *Systems Development Life Cycle* (SDLC) yang biasa digunakan dalam membangun sebuah sistem, salah satunya adalah model *waterfall*. *Waterfall* merupakan model yang bersifat sistematis dan termasuk dalam model klasik, nama lainnya adalah *Linear Sequential Model* (Pressman, 2001) Tahapan-tahapan model *waterfall* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model *Waterfall* (Pressman, 2001)

Penjelasan tahapan-tahapan *waterfall* tersebut yaitu:

1. *Analysis* (Analisis)

Fase ini merupakan proses analisa terhadap sistem yang sedang berjalan dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem, sehingga kebutuhan yang diperlukan untuk sistem baru akan didapatkan.

2. *Design* (Perancangan)

Perancangan merupakan proses penentuan cara kerja sistem dalam hal perancangan antarmuka, *database*, dan perancangan alur program. Perancangan diperlukan untuk menggambarkan sistem baru dengan tujuan memenuhi kebutuhan pengguna.

3. *Implementation* (Implementasi)

Tahapan implementasi yaitu tahap rancangan sistem yang dibentuk menjadi suatu kode program untuk pembuatan sistem.

4. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur atau tidak dan memastikan sistem terhindar dari error yang terjadi. Testing juga dilakukan untuk memastikan kevalidan dalam proses input sehingga dapat menghasilkan output yang sesuai.

5. *Maintenance* (Pemeliharaan)

Fase ini yaitu pemeliharaan dan pengembangan sistem yang berguna untuk melihat kemampuannya, mengecek jika masih ada ditemukan error atau ada penambahan fitur-fitur yang belum ada pada sistem tersebut. Pengembangan diperlukan ketika adanya perubahan dari pengguna seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

2.6 Aplikasi

Aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia (Pramana, 2012). Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu (Yuhefizar, 2012). Aplikasi adalah program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu (Supriyanto, 2005),.

Berdasarkan dua pendapat ahli tersebut, maka dapat disimpulkan bahwa aplikasi merupakan sebuah program perangkat lunak yang dibuat dengan tujuan memenuhi kebutuhan dari aktivitas yang dilakukan oleh pengguna serta mempermudah pekerjaan dari pengguna tersebut. Pada penelitian ini Aplikasi

akan menjadi alat bantu kepada mahasiswa dan perantau yang dibangun dengan *Progressive Web App (PWA)*.

2.7 Rumah

Menurut UU Nomor 1 Tahun 2011, bahwa setiap orang berhak hidup sejahtera lahir dan batin, bertempat tinggal, dan mendapatkan lingkungan hidup yang baik dan sehat.

Rumah sebagai tempat tinggal tetap maupun sementara misalnya rumah kost, harus memenuhi syarat-syarat kehidupan yang layak, dipandang dari berbagai segi kehidupan masyarakat. Dengan rumah akan perlindungan, beristirahat dan bersuka ria bersama keluarga.

2.8 Rumah Kost

Menurut Kamus Besar Bahasa Indonesia, kost merupakan jasa yang menawarkan sebuah kamar atau tempat tinggal untuk ditinggali dengan sejumlah pembayaran tertentu untuk setiap periode tertentu (umumnya pembayaran per bulan atau per tahun). Selain itu tempat kost lebih akrab digunakan sebagai domisili, karena kebanyakan tempat kost disewa dalam jangka waktu yang cukup lama dari pada hotel atau penginapan yang menggunakan hitungan hari.

Untuk menentukan pilihan rumah kost yang sesuai kebutuhan pasti mempunyai kriteria, kriteria tersebut di peroleh melalui kuesioner survei kepada mahasiswa dan perantau yang mencari rumah kost serta mengacu penelitian yang dilakukan oleh (Hayati, 2019), (Asniadi, 2019), dan (Sugianto, 2016). Berikut uraian kriteria yang peroleh:

2.8.1 Lokasi

Lokasi merupakan salah satu faktor dari situasional yang ikut berpengaruh pada keputusan dalam menentukan tempat. Lokasi adalah ilmu yang menyelidiki tata ruang (*spatial order*) kegiatan ekonomi, atau ilmu yang menyelidiki alokasi geografis dari sumber-sumber yang potensial, serta hubungannya dengan atau pengaruhnya terhadap keberadaan berbagai macam usaha/kegiatan lain baik ekonomi maupun sosial. Faktor penentuan lokasi ialah lingkungan masyarakat, kedekatan dengan pasar, fasilitas dan biaya transportasi (Tarigan, 2006). Dalam menentukan lokasi rumah kost, mahasiswa dan perantau

lebih mengutamakan faktor lokasi yang dekat dengan kampus ataupun tempat bekerja, sehingga lokasi merupakan kriteria dalam mencari rumah kost.

2.8.2 Jarak

Jarak merupakan salah satu konsep dasar dalam ilmu geografis. Jarak mempunyai peranan penting di kehidupan sosial, ekonomi, dan politik. Jarak berkaitan erat dengan lokasi, jarak juga menjadi sebagai acuan untuk menempuh suatu perjalanan atau perpidahan dari satu tempat ketempat yang lain, baik yang berkaitan dengan waktu perjalanan yang diperlukan, maupun dengan satuan transportasi. Faktor jarak juga akan berpengaruh terhadap faktor lainnya, jarak dari rumah kost ketempat tujuan akan mempengaruhi waktu dan akan menambah biaya perjalanan jika jarak jauh dari tempat tujuan, oleh karena itu jarak merupakan salah satu kriteria yang dilakukan dalam mencari rumah kost.

2.8.3 Harga

Harga adalah jumlah uang (ditambah beberapa barang kalau mungkin) yang dibutuhkan untuk mendapatkan sejumlah kombinasi dari barang beserta pelayanannya. Harga adalah satuan moneter atau ukuran lainnya termasuk barang dan jasa lainnya yang ditukarkan agar memperoleh hak kepemilikan atau pengguna suatu barang dan jasa (Tjiptono, 2005). Faktor harga sewa rumah kost juga merupakan kriteria penting yang bisa mempengaruhi dalam memilih rumah kost yang sesuai dengan kondisi ekonomi mahasiswa atau perantau di Kota Pontianak.

2.8.4 Fasilitas

Menurut Prof. Dr. Hj. Zakiah Drajat fasilitas merupakan semua hal yang dapat mempermudah upaya serta memperlancar kerja dalam rangka mencapai suatu tujuan tertentu. Pendapat ini menekankan pada adanya tujuan yang akan dicapai dengan memanfaatkan fasilitas. Fasilitas yang ada pada rumah kost juga, akan menjadi faktor penunjang agar penghuni rumah kost dapat tinggal dengan nyaman sehingga penghuni dapat lebih produktif dan efisien dalam mencapai suatu tujuan.

2.8.5 Jenis Kost

Jenis kost pada umumnya memiliki bentuk bangunan rumah memanjang dan bertingkat yang terdiri dari banyak kamar. Berbeda dengan rumah kontrakan yang bentuk bangunannya hanya rumah biasa seperti rumah tinggal yang hanya terdiri dari beberapa kamar saja dan biasanya kamar tidak lebih banyak dari kost-kost-an. Persamaan keduanya terdapat pada sistem pembayaran yang menggunakan sistem kontrak kesepakatan antara pemilik dan calon penghuni.

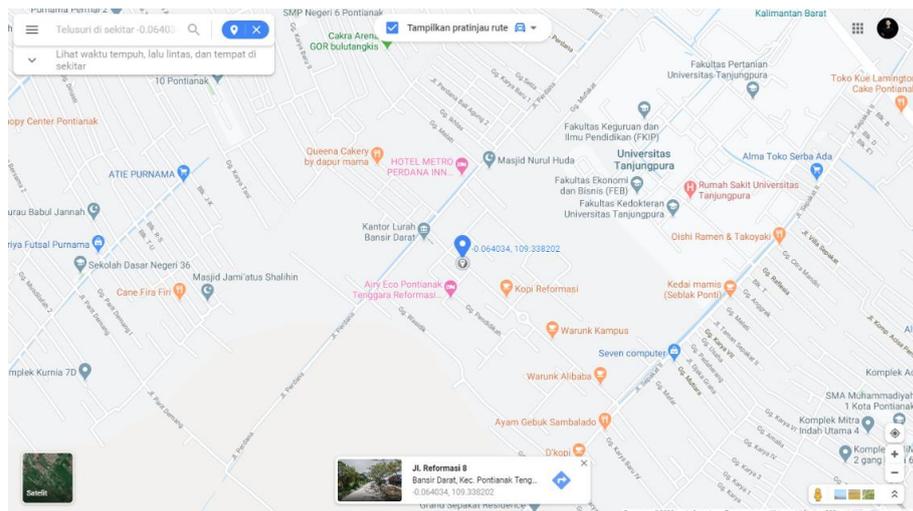
Jenis rumah kost juga menjadi kriteria yang paling sering dipertanyakan oleh calon penghuni rumah kost, di antaranya kategori rumah kost berdasarkan jenis kelamin, suku, dan agama. Jenis kost tentunya pasti memiliki peraturan tersendiri agar penghuni merasa nyaman tinggal di rumah kost.

2.9 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau juga dikenal sebagai *Geographic Information System* (GIS) pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan *geografis*. Empat puluh tahun kemudian GIS berkembang tidak hanya bertujuan untuk menyelesaikan permasalahan geografi saja, tetapi sudah merambah ke berbagai bidang, seperti analisis penyakit epidemi (demam berdarah) dan analisis kejahatan (kerusuhan), termasuk analisis kepariwisataan. Kemampuan dasar dari SIG adalah mengintegrasikan berbagai operasi basis data seperti query, menganalisisnya serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan SIG dengan sistem informasi lain (Prahasta, 2014).

2.10 Google Maps

Google Maps adalah sebuah jasa peta global virtual gratis dan *online* yang disediakan oleh perusahaan *Google*. *Google Maps* yang dapat ditemukan di alamat <http://maps.google.com>. *Google Maps* menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia. *Google Maps* juga menawarkan pencarian suatu tempat dan rute perjalanan. Dapat dilihat pada gambar 2.2.

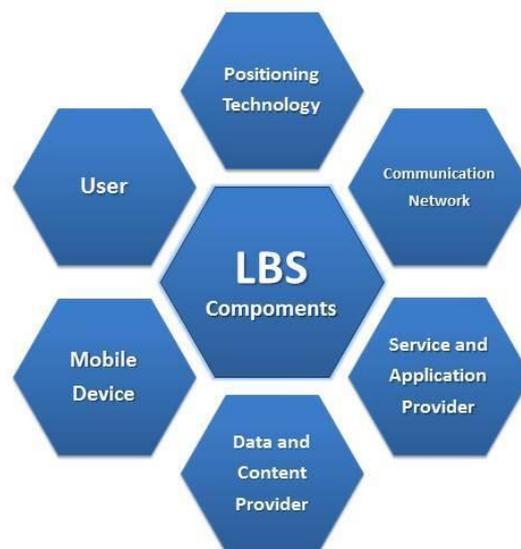


Gambar 2.2 Tampilan *Google Maps*

Google Maps API adalah sebuah layanan (*service*) yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi enterprise di dalam *website* nya, *Google Maps API* adalah suatu *library* yang berbentuk *Javascript*. (Kindarto, 2008). Pada penelitian ini *Google Map API* akan menjadi *service* seperti *interface* bagi pengguna berupa *Map* yang akan dipakai.

2.11 Location Based Service

Location Based Service (LBS) atau Layanan Berbasis Lokasi merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak (Steiniger, Neun, & Edwardes, 2006). Dalam layanan Berbasis Lokasi terdapat lima komponen penting seperti terlihat pada Gambar 2.3.



Gambar 2.3 Komponen Dasar *Location Based Service* (LBS)

Setiap komponen mempunyai fungsi.

1. *Mobile Devices*, merupakan suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan *text*.
2. *Communication Network*, komponen ini mengirim data pengguna dan informasi yang diminta dari *Mobile* terminal ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupa jaringan *seluler* (GSM, CDMA), *Wireless Local Area Network* (WLAN), atau *Wireless Wide Area Network* (WWAN).
3. *Positioning Component*, digunakan untuk memproses suatu layanan maka posisi pengguna harus diketahui.
4. *Service dan Application Provider*, penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.
5. *Data dan Content Provider*, penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, *data* dapat diminta dari *data dan content provider*.

Pada penelitian ini memanfaatkan Layanan Berbasis Lokasi atau yang kita kenal dengan *LBS* merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan

dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak seperti *smartphone* yang akan dipakai pada penelitian ini.

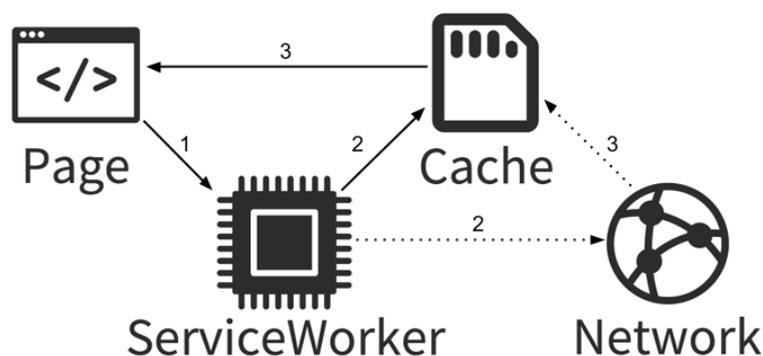
2.12 Progressive Web App (PWA)

Progressive Web App (PWA) adalah sebuah *website* yang dibangun menggunakan teknologi *web* modern, namun dapat berlaku seperti sebuah *mobile app*. Pada tahun 2015, *Google Engineer* Alex Russel dan Frances Berriman memberi istilah *Progressive Web App* pada konsep *web app* yang dapat memberikan *user experience* dalam keandalan (*reliability*), kecepatan (*speed*) dan keterlibatan pengguna (*user engagement*). Pada penelitian ini *Progressive Web App* (PWA) merupakan teknologi *modern* yang dapat diakses seperti aplikasi *mobile* (Santoso, 2019).

2.12.1 Service Worker

Service Worker merupakan salah satu jenis *web worker*, *javascript* yang berjalan di belakang layar (*background*) tanpa mempengaruhi kinerja halaman *web*. *Service Worker* pada dasarnya adalah *file javascript* yang berjalan *client side* secara terpisah dari rangkaian *browser* utama yang berfungsi untuk mencegah permintaan jaringan, melakukan *cache* atau mengambil sumber daya dari *cache* dan mengirimkan pesan.

Secara teknis, *Service Worker* menyediakan *script “network proxy”* di *web browser* untuk mengelola permintaan *web* (*HTTP request*) secara terprogram. *Service Worker* menggunakan mekanisme *cache* secara efisien dan memungkinkan perilaku *error-free* selama periode *offline*. Mekanisme kerja dari *Service Worker* dapat dilihat pada gambar 2.4.



Gambar 2.4 Mekanisme kerja *Service Worker*

Pada gambar diatas dijelaskan bahwa mekanisme kerja dari *service worker* yaitu menerima *request* dari halaman *web* kemudian mengirimkan ke *server*. Selanjutnya *service worker* bertugas menerima *response* dari *server* untuk diteruskan kembali ke halaman *web*, sistem *service worker* akan membantu aplikasi ini agar bisa di akses dalam keadaan *offline* sekalipun.

2.12.2 Kelebihan dan Kekurangan PWA

Kemunculan teknologi PWA membawa pengaruh positif pada pengembangan aplikasi *web*. Namun aplikasi ini masih terbilang baru dan masih perlu pengembangan lebih lanjut. Berikut ini akan dibahas kelebihan dan kekurangan (atau keterbatasan saat ini) dari *Progressive Web App*.

1. Kelebihan PWA:

- Respon Seperti *Mobile App*
- *Layout Responsive*
- Tersedia di *Mode Offline*
- *Add to Home Screen (A2HS)*
- *Push Notification*

2. Keterbatasan PWA:

- Perlu *HTTPS*
- Dukungan *Web Browser Modern*
- Dukungan Pengguna

2.12.3 Karakteristik PWA

1. *Progressive*:

bekerja sama di semua *user*, tanpa melihat *web browser* yang dipakai, karena telah memiliki peningkatan progresif pada prinsipnya

2. *Responsive*:

Cocok di segala bentuk *device*, (*desktop, mobile, tablet* lainnya)

3. *Connectivity independent*:

Ditingkatkan dengan *service worker* untuk bekerja secara *online* atau pada jaringan internet kualitas rendah

4. *App-like*:
Terasa seperti *app*, karena model *App Shell* akan memisahkan fungsionalitas aplikasi dari kontennya
5. *Fresh*:
Selalu *update* berkat adanya proses *update service worker*
6. *Safe*:
Dilayani oleh HTTPS yang mencegah pengintaian (*snooping*) dan memastikan konten tidak rusak
7. *Discoverable*:
Teridentifikasi sebagai “*application*” berkat *manifest* W3C dan *registrasi service worker*, sehingga memungkinkan *search engine* untuk mengenalinya
8. *Re-engageable*:
Memudahkan keterlibatan *user* dengan *fitur push notification*.
9. *Installable*:
Memungkinkan *user* untuk menambahkan *apps* yang sering digunakan di layar *home screen* tanpa harus ke *app store*.
10. *Linkable*:
Share aplikasi dengan mudah melalui *Uniform Resource Locator (URL)*, tanpa repot-repot menginstalnya.

2.13 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. (Nugroho, 2005).

Unified Modeling Language (UML) adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi

tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat (A. S. Rosa, 2013).

Di Dalam UML ada 13 buah diagram yang dikelompokkan kedalam tiga kategori yaitu:

1. *Structure Diagrams*, yaitu kumpulan diagram-diagram yang menggambarkan struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams*, yaitu kumpulan diagram-diagram yang menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi di dalam sistem.
3. *Interaction Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan interaksi sistem dengan sistem lain ataupun interaksi antar subsistem dalam sebuah sistem

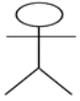
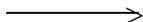
Secara garis besar, beberapa diagram utama sudah dapat menggambarkan keseluruhan sistem. Diagram tersebut antara lain *use case diagram*, *class diagram*, *sequence diagram*, dan *activity diagram*.

2.13.1 Use Case Diagram

Use Case Diagram bersifat statis, yang memperlihatkan himpunan *Use Case* dan aktor-aktor (suatu jenis khusus dari kelas) dan menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sistem (Widodo, 2011:10).

Deskripsi simbol-simbol yang digunakan pada *Use Case Diagram* dapat dilihat pada Tabel 2.4 (Nugroho, 2005).

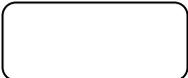
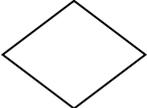
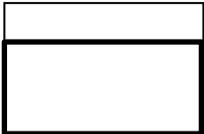
Tabel 2.4 Deskripsi Notasi pada *Use Case Diagram*

No	Notasi	Nama	Deskripsi
1		Use Case	Menggambarkan fungsionalitas yang dimiliki sistem.
2		Actor	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang dikembangkan.
3		Association	Lintasan komunikasi antara actor dengan use case.
4	<<extend>> 	Extended (Ekstensi)	Penambahan perilaku ke suatu <i>use case</i> dasar.
5	<<include>> 	Include (Menggunakan)	Penambahan perilaku ke suatu <i>use case</i> dasar yang secara <i>explicit</i> mendeskripsikan penambahan tersebut.
6		Generalisasi / generalization	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya

2.13.2 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity diagram* sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan *parallel behavior* atau menggambarkan interaksi antara beberapa *use case*. Berikut merupakan simbol notasi *Activity Diagram* pada Tabel 2.5 (A. S. Rosa, 2013).

Tabel 2.5 Deskripsi Notasi pada *Activity Diagram*.

No	Notasi	Nama	Deskripsi
1		Status Awal (<i>initial node</i>)	Status awal aktivitas, sebuah diagram aktivitas memiliki status awal.
2		Status Akhir (<i>final node</i>)	Status akhir yang dilakukan sistem.
3		Aktivitas (<i>activity</i>)	Aktivitas yang dilakukan oleh sistem, biasanya diawali oleh kata kerja.
4		Percabangan (<i>decision</i>)	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
5		Penggabungan (<i>join</i>)	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.13.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. (A. S. Rosa, 2013). *Class Diagram* mendeskripsikan jenis – jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. (Fowler, 2005). *Class Diagram* digunakan untuk menggambarkan hubungan kelas – kelas antara

satu dengan yang lain seta memiliki atribut dan operasi yang terdapat dalam sistem yang akan dibuat. Atribut merupakan variabel – variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu kelas. Atribut dan metode dapat memiliki salah satu sifat sebagai berikut:

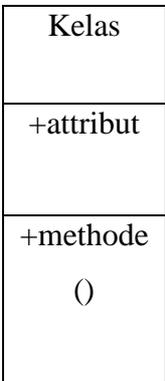
1. *Private* (-), hanya dapat digunakan oleh *class* yang memilikinya
2. *Public* (+), dapat digunakan oleh *class* lain.
3. *Protected* (#), hanya dapat dipanggil oleh *class* yang bersangkutan jumlah suatu anak yang mewarisinya.

Nilai kardinalitas dan *multiplicity* sebuah *class* menunjukkan jumlah suatu objek yang dapat berhubungan dengan objek lain. Berikut nilai kardinalitas atau *multiplicity* pada Tabel 2.5 dan notasi *class diagram* pada Tabel 2.6 (A. S. Rosa, 2013).

Tabel 2.6 Jenis-jenis *Multiplicity*

No	Indikator	Keterangan
1	0 .. 1	Nol atau satu
2	1	Hanya satu
3	0 .. *	Nol atau lebih
4	1 .. *	Satu atau lebih

Tabel 2.7 Deskripsi Notasi pada *Class Diagram*

No	Notasi	Nama	Deskripsi
1		<i>Class</i>	Menggambarkan konsep dasar pemodelan sistem.
2		Asosiasi	Sebuah garis solid antara dua class,

		(Association)	ditarik dari class sumber ke class target lebih spesifik, digunakan dalam struktur pewarisan.
3		Ketergantungan (Dependency)	Relasi antara dua elemen jika perubahan definisi sebuah elemen (supplier atau sumber) dapat menyebabkan perubahan pada elemen lainnya (Client atau target).

2.14 Teknologi Pendukung

2.14.1 Web

World Wide Web (WWW) atau biasa disebut dengan web merupakan salah satu sumber daya Internet yang berkembang pesat. Pertama kali aplikasi web dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language*) dan protokol yang digunakan dinamakan HTTP (*HyperText Transfer Protocol*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML yang sekarang ini terdapat banyak skrip seperti: PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah *applet* (*java*). Jadi aplikasi *web* atau aplikasi berbasis *web* (*Web-based application*) adalah aplikasi untuk menyampaikan informasi kepada pengguna yang menggunakan layanan *Internet* berbasis *web*. Dalam aplikasi tersebut, terjadi pertukaran antara *klien* (komputer yang meminta informasi) dengan *server* (komputer yang memasok atau menanggapi informasi). *Web* memberikan informasi secara *online* melalui *internet* langsung. *Klien* melakukan permintaan informasi dengan menggunakan *browser* (contoh *browser*: *Internet Explorer*, *Opera*, *Mozilla*, dan sebagainya). *Server* menerima informasi dan melayani permintaan dari *client*. Hal ini biasa disebut dengan *web server* (contoh *web server*: *Apache*, *IIS*, *Xitami*, dan sebagainya). Setelah itu, *web server* akan berkomunikasi dengan *middleware* (contoh *middleware*: *ASP*, *JSP*, *PHP*, dan sebagainya) untuk bisa berhubungan dengan basis data atau *database* (contoh *database*: *access*, *oracle*, *sql*, dan sebagainya). Setelah berinteraksi dengan *database*, *server* yang telah mendapatkan

informasi akan memberikan tanggapan terhadap *klien* yang meminta informasi tadi Kadir (2005).

2.14.2 XAMPP

XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang digunakan sebagai alat pembantu pengembangan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa paket perangkat lunak berbeda dalam satu paket. Di dalam Paket XAMPP terdapat tiga paket penting yaitu *Apache* sebagai *web server*, PHP sebagai bahasa pemrograman dan MySQL sebagai *database*. *Apache* adalah *server web (web server)* yang dapat dijalankan di banyak sistem operasi.

Apache merupakan perangkat lunak open source yang dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

2.14.3 MYSQL

MySQL adalah *software* atau program aplikasi *database*, yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi teks dan juga angka. Semua informasi data yang dipakai di dalam satu program aplikasi disimpan dalam satu *software database*, yaitu MySQL.

Dengan menggunakan *database* MySQL, maka data yang tersimpan di dalam *database* dapat diakses secara bersama menggunakan beberapa komputer/laptop yang berbeda, konsep ini sering disebut dengan sistem *multi user*. *Database* MySQL juga dapat diinstal pada sebuah komputer pusat (*server*) yang tersimpan di dalam ruang *server*, kemudian datanya diakses melalui komputer/laptop yang terinstal dengan program *client* seperti program kasir toko atau program akademik siswa yang letaknya berbeda tempat, dan konsep tersebut disebut dengan sistem *client/server* (Nugroho, 2014).

2.14.4 Javascript

Javascript adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer (IE)*, *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan *tag SCRIPT* (Sunyoto, 2007). Beberapa hal tentang *Javascript*:

1. *Javascript* didesain untuk menambah interaktif suatu *web*.
2. *Javascript* merupakan sebuah bahasa *scripting*.
3. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
4. *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*).
5. *Javascript* biasanya disisipkan (*embedded*) dalam halaman HTML.
6. *Javascript* adalah bahasa *interpreter* (yang berarti *script* dieksekusi tanpa proses kompilasi).

2.14.5 HTML

HTML merupakan halaman yang berada pada suatu situs internet atau *web*. HTML merupakan metode yang menautkan (*link*) satu dokumen ke dokumen lain melalui teks. Menurut Deris Setiawan, HTML merupakan framework internet, hampir semua situs web yang ada menggunakan HTML untuk menampilkan teks, grafik, suara, dan animasinya (Kuswayatno, 2006). HTML adalah suatu bahasa yang dikenali oleh web browser untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*) (Oktavian 2010).

HTML adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan HTML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di *CERN* pada tahun 1989 (*CERN* adalah lembaga penelitian fisika energi tinggi di Jenewa).

2.14.6 *Laravel*

Pengertian *framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan. Salah satu *framework* yang banyak digunakan oleh programmer adalah *framework laravel*. *Laravel* adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model – view – controller*. *Laravel* berada di bawah lisensi MIT *License* dengan menggunakan *Github* sebagai tempat berbagi *code* menjalankannya (Naista, 2017). Dalam penggunaannya *laravel* memiliki beberapa kekurangan salah satunya yaitu ukuran file yang cukup besar. Di dalam *laravel* terdapat *file* yang sifatnya *default* seperti *vendor*. *File* tersebut tidak boleh dihapus sembarangan sehingga ukuran *website* yang dibuat berukuran cukup besar. Selain itu, dibutuhkan koneksi *internet* untuk instalasi dan mengunduh *library laravel*, dan PHP minimal versi 5.4 untuk menjalankannya (Naista, 2017).

2.14.7 PHP

PHP merupakan bahasa scripting *server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* lah yang akan menerjemahkan *script program*, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan.

PHP adalah bahasa yang sederhana namun kuat dirancang untuk membuat konten HTML. Sejak dimulai pada tahun 1994, PHP telah mampu menyaingi bahasa *web* karena keunggulan popularitas bahasa dan kemudahan penggunaan. PHP pertama kali dibuat oleh Rasmus Lerdorf. PHP sering digunakan untuk membangun *web* dinamis dimana proses keseluruhan berjalan pada *web server* dan menampilkan hasilnya pada *web browser* (Prasetyo, 2004).

PHP adalah bahasa pemrograman untuk membuat situs *web* dinamis dan interaktif. PHP berjalan di *server web* dan melayani pengunjung dengan halaman *web* sesuai permintaan (Doyle & Matt, 2009).

2.15 Pengujian perangkat lunak

Pengujian perangkat lunak merupakan suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah sesuai dengan yang diharapkan atau belum. Menurut (Roger S. Pressman, 2002), pengujian adalah proses eksekusi suatu program untuk menemukan kesalahan sebelum digunakan oleh pengguna akhir (*end-user*).

2.15.1 Pengujian *Black Box*

Black Box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman, 2010).

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung untuk menemukan hal-hal berikut (Mustaqbal, M. S., Firdaus, R. F. & Rahmadi, H., 2015).

1. Fungsi yang tidak benar atau tidak ada
2. Kesalahan antarmuka (*interface errors*)
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan performansi (*performance errors*)
5. Kesalahan inisialisasi dan terminasi

Adapun teknik pengujian secara *black box* menurut (Sukamto, Rosa Ariani, 2009), adalah sebagai berikut:

1. *Boundary value Analysis (BVA) / Limit Testing* adalah banyak kesalahan terjadi pada kesalahan masukan. BVA mengijinkan untuk menguji seleksi kasus uji yang menguji batasan nilai input. BVA merupakan komplemen dari *equivalence partitioning*. Lebih pada memilih elemen-elemen didalam kelas ekivalen pada bagian sisi batas dari kelas.
2. *Comparison Testing* adalah pengujian pada beberapa aplikasi *reliability* dari sebuah perangkat lunak. Redundansi perangkat keras dan perangkat lunak

mungkin digunakan untuk meminimalisir kesalahan (*error*). Untuk redundansi perangkat lunak, gunakan tim yang terpisah untuk mengembangkan setiap versi perangkat lunak yang independen. Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama. Jalankan semua versi dengan paralel dan perbandingan keluaran secara *real-time*.

3. *Sample Testing* adalah pengujian yang melibatkan beberapa nilai yang terpilih dari sebuah kelas ekuivalen. Mengintegrasikan nilai pada kasus uji dan nilai-nilai yang terpilih mungkin dipilih dengan urutan tertentu atau interval tertentu
4. *Robustness Testing* adalah pengujian dengan data input dipilih diluar spesifikasi yang telah didefinisikan. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid
5. *Behavior Testing* adalah pengujian dengan hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur *data stack*.
6. *Requirement Testing* adalah spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (*input, output, fungsi, performansi*) diidentifikasi pada tahap spesifikasi kebutuhan dan desain. *requirement testing* melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program.
7. *Performance Testing* adalah mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi dan lain-lain. Untuk mencari tahu beban kerja atau kondisi konfigurasi program dan dapat digunakan untuk menguji batasan lingkup program.
8. Uji ketahanan (*Endurance Testing*) adalah melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai spesifikasi kebutuhan.
9. *Equivalence partitioning* adalah membagi input menjadi kelas-kelas data yang dapat digunakan untuk meregenerasi kasus uji dengan tujuan untuk menemukan kelas-kelas kesalahan. Selain itu, *equivalence partitioning* berdasarkan pada kesamaan kelas-kelas kondisi *input*. Sebuah kelas yang ekuivalen merepresentasikan kumpulan status/kondisi yang valid atau tidak valid. Sebuah kondisi input dapat berupa nilai numerik yang spesifik, rentan

nilai, kumpulan nilai yang berkaitan, atau kondisi *boolean*.

10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*) adalah teknik yang merupakan suplemen dari *equivalence testing* dengan menyediakan cara untuk memilih kombinasi data input dan melibatkan kondisi *input* (*Cause*) dan kondisi *output* (*Effect*) untuk mencegah pendefinisian kasus uji yang terlalu banyak.

2.15.2 Pengujian *User Acceptance Testing* (UAT)

Menurut (Perry, William E., 2006), *User Acceptance Testing* merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah *staff* / karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya.

Menurut (Lewis, 2002), setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem *software* memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

Menurut (Black, 2005), *acceptance testing* biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu. Pada pengembangan *software* dan *hardware* komersial, *acceptance test* biasanya disebut juga "*alpha tests*" (yang dilakukan oleh pengguna *in-house*) dan "*beta tests*" (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). *Alpha* dan *beta test* biasanya juga menunjukkan bahwa produk sudah siap untuk dijual atau dipasarkan. *Acceptance testing* mencakup data, *environment* dan skenario yang sama atau hampir sama pada saat *live* yang biasanya berfokus pada skenario penggunaan produk tertentu.

Dari definisi di atas, *user acceptance testing* adalah pengujian yang dilakukan oleh pengguna dari sistem tersebut untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna. (Binus, 2017),.