

BAB II

LANDASAN TEORI

1.1 Kajian Pustaka

Penelitian dari Adianto, dkk (2017) dalam penelitiannya menerapkan metode *simple additive weighting (SAW)* dalam sistem pengambilan keputusan (SPK) untuk memilih rumah tinggal. *Developer* bertanggung jawab menawarkan berbagai kriteria rumah dari mulai harga, lokasi, desain maupun cara pembayaran. Proses pengembangan sistem setelah memiliki data yang cukup untuk menunjang kegiatan pengembangan sistem. Perancangan sistem penentuan pemilihan rumah dalam bentuk program dengan menggunakan bahasa pemrograman PHP. Dalam penelitiannya pengujian yang dilakukan metode *simple additive weighting (SAW)* mampu memberikan alternatif terbaik dalam penentuan perumahan, memberikan rekomendasi kepada calon pembeli sesuai data yang dimasukkan oleh admin.

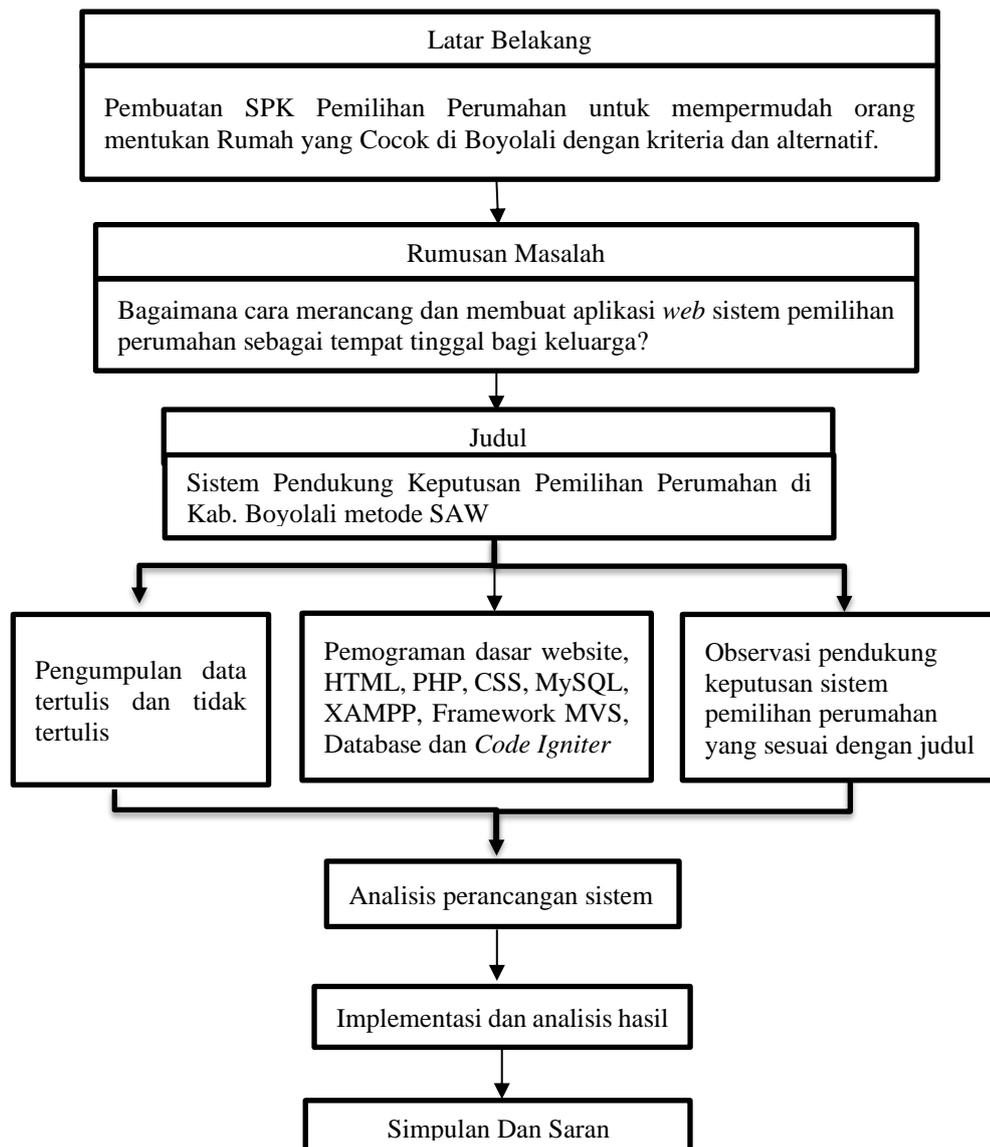
Perbedaan dengan penelitian sebelumnya adalah menawarkan dengan kriteria yang lebih banyak dicari mulai dari harga, lokasi, tipe rumah, jumlah kamar, dan fasilitas. Proses perancangan sistem pemilihan perumahan dalam bentuk program dengan menggunakan bahasa PHP.

Penelitian dari Riyanto dan Haryanti (2017) dalam penelitiannya menerapkan metode *simple additive weighting (SAW)* memberikan rekomendasi dalam pengambilan keputusan untuk pemilihan *teller pooling* terbaik. Dalam metodenya ditetapkan 5 kriteria. Kriteria-kriteria tersebut antara lain: *Tangible, Realibility, Responsiveness, Assurance, Emphaty*. Dari hasil perhitungan dengan menggunakan metode *simple additive weighting (SAW)* dapat membantu dalam menentukan *teller pooling* terbaik dengan lebih tepat berdasarkan kemampuan yang tidak jauh berbeda dengan *teller pooling* yang lainnya.

Penelitian dari Hermanto dan Nailul Izzah (2018) dalam penelitiannya menerapkan metode *Simple Additive Weighting (SAW)* memberikan rekomendasi dalam pengambilan keputusan untuk pemilihan motor. Rancangan desain sistem pendukung keputusan pemilihan produk motor menghasilkan temuan sebuah desain yang mampu mempermudah pengguna dalam menjalankan sistem pendukung keputusan pemilihan produk motor terbaik.

1.2 Kerangka Pemikiran

Alur logika penelitian perancangan dan pembangunan Sistem Pemilihan Perumahan di Boyolali pada penelitian ini ditunjukkan pada Gambar 2.1



Gambar 2.1 Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah:

1. Latar Belakang Masalah

Informasi tentang perumahan masih minim informasi untuk membandingkan suatu perumahan dengan harga yang diinginkan oleh masyarakat. Kurang efisiennya waktu dalam penjelasan sebuah rumah kepada konsumen karena hanya melalui sebuah brosur, membutuhkan biaya ekstra untuk mencetak brosur.

2. Rumusan Masalah

Rumusan masalah pada penelitian ini adalah: Bagaimana membangun sistem *website* SPK Pemilihan Perumahan di Boyolali dengan metode SAW?

3. Penguasaan Dasar

Penguasaan dasar merupakan suatu kunci terciptanya sistem pemilihan perumahan tersebut, karena pembuatan sistem ini menggunakan bahasa pemrograman Pemrograman dasar website, HTML, PHP, CSS, MySQL, XAMPP, Framework MVS, Database dan *Code Igniter*.

4. Pengumpulan Data

Tahap pengumpulan data pada penelitian ini melalui studi pustaka, wawancara, populasi dan sampel, dan kuisioner. Pengumpulan data bertujuan untuk mengetahui permasalahan dan memenuhi kebutuhan informasi tentang perumahan.

5. Analisis & perancangan sistem

Analisis dan perancangan sistem pemilihan perumahan bertujuan untuk mengetahui kesiapan data yang terkumpul dan rancangan dari aplikasi sistem pemilihan perumahan yang diharapkan dapat memberikan suatu informasi edukatif dan mudah dipahami oleh pengguna. Perancangan sistem pada penelitian ini menggunakan HTML, PHP dan CSS

6. Implementasi sistem

Mengimplementasikan sistem pemilihan perumahan menggunakan bahasa pemrograman HTML, PHP dan CSS. *Database My. sql* sebagai media penyimpanan datanya. Proses implementasi dikerjakan setelah menganalisis dan perancangan sistem.

7. Pengujian sistem

Pengujian sistem merupakan tahap setelah sistem berhasil dibuat. Pengujian sistem bertujuan untuk mengetahui kelebihan, kelemahan serta mengetahui kelayakan suatu sistem untuk digunakan oleh masyarakat. Pengujian sistem pada penelitian ini menggunakan metode *Webqual* sebagai teknik pengukuran kualitas *website* berdasarkan persepsi pengguna.

8. Dokumentasi

Pada tahap dokumentasi ini menjelaskan dan memberikan tampilan hasil dari pembuatan aplikasi sistem pendukung keputusan pemilihan perumahan.

1.3 Teori-teori Pendukung

1.3.1 Sistem

Menurut Kadir (2014), sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai tujuan. Tujuan utama sistem yang umum ada tiga macam yaitu:

- a. Untuk mendukung fungsi kepengurusan manajemen
- b. Untuk mendukung pengambilan keputusan manajemen
- c. Untuk mendukung kegiatan operasi perusahaan

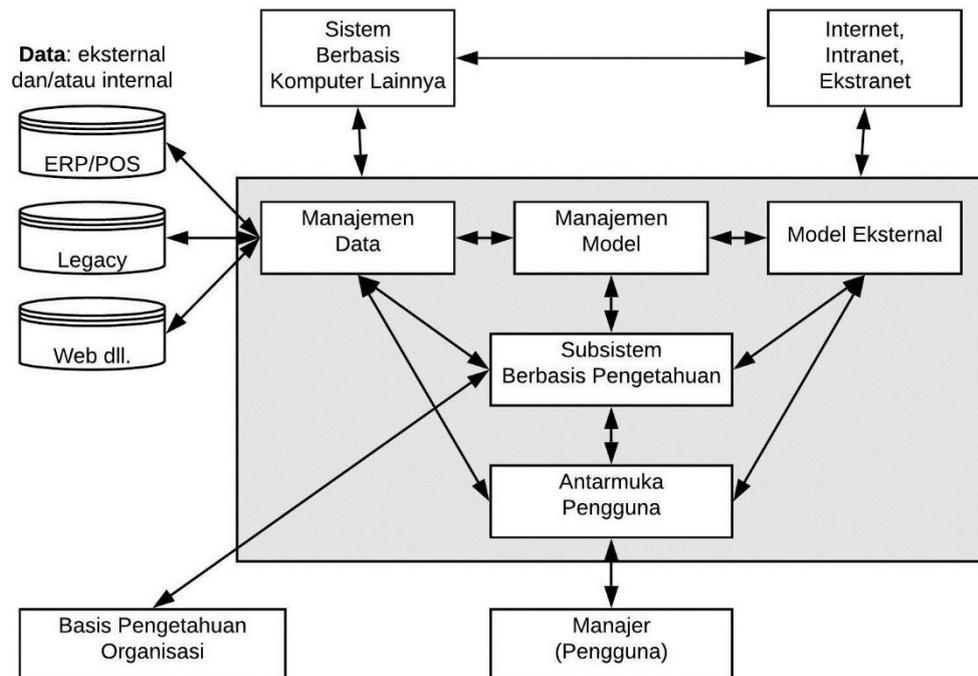
1.3.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support Systems* (DSS) merupakan bagian dari sistem informasi berbasis komputer (termasuk berbasis pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan.

Menurut Turban, dkk (2011), 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.

1.3.3 Komponen SPK

Aplikasi SPK dapat disusun dari subsistem manajemen data, subsistem manajemen model, subsistem antarmuka pengguna, dan subsistem manajemen berbasis pengetahuan (Turban dkk. 2010). Skema hubungan antar komponen dalam SPK ditunjukkan pada Gambar 2.2.



Gambar 2.2 Skema Hubungan Antar Komponen dalam SPK

1.3.4 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)

Metode *Simple Additive Weighting* (SAW) membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak 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 *Simple Additive Weighting* (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.

Langkah Penyelesaian Simple Additive Weighting (SAW) sebagai berikut :

1. Menentukan Alternatif (A_i).
2. Menentukan kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_j .
3. Memberikan nilai rating kecocokan setiap alternatif pada setiap kriteria.
4. Menentukan bobot preferensi atau tingkat kepentingan (W) setiap kriteria.
 $W = (W_1, W_2, \dots, W_j) \dots\dots\dots (2.1)$
5. Membuat tabel rating kecocokan dari setiap alternatif pada setiap kriteria.
6. Membuat matrik keputusan X yang dibentuk dari tabel rating kecocokan dari setiap alternatif pada setiap kriteria. Nilai x setiap alternatif (A_i) pada setiap kriteria (C_j) yang sudah ditentukan, dimana, $i=1,2,\dots,m$ dan $j=1,2,\dots,n$.

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1j} \\ x_{21} & x_{22} & \ddots & x_{2j} \\ \vdots & \vdots & \vdots & \vdots \\ x_i & x_{i2} & \dots & x_{ij} \end{pmatrix} \dots\dots\dots \text{Persamaan (2.2)}$$

7. Melakukan normalisasi matrik keputusan dengan cara menghitung nilai rating kinerja ternormalisasi (r_{ij}) dari alternatif A_i pada kriteria C_j .

$$r_{ij} \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} & \text{Jika } j \text{ atribut keuntungan (benefit)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ atribut biaya (cost)} \end{cases} \dots\dots\dots \text{Persamaan 2.3}$$

Keterangan :

- r_{ij} : Nilai rating kinerja ternormalisasi
- x_{ij} : Nilai atribut yang dimiliki dari setiap kriteria
- $\max_i x_{ij}$: Nilai terbesar dari setiap kriteria
- $\min_i x_{ij}$: Nilai terkecil dari setiap kriteria
- Benefit : Jika nilai terbesar adalah terbaik

Cost :Jika nilai terkecil adalah terbaik

Penjelasan persamaan 2.3 :

- a. Dikatakan kriteria keuntungan apabila nilai X_{ij} memberikan keuntungan bagi pengambil keputusan, sebaliknya kriteria biaya apabila X_{ij} menimbulkan biaya bagi pengambil keputusan.
- b. Apabila berupa kriteria keuntungan maka nilai X_{ij} dibagi dengan nilai $\text{Max}_i X_{ij}$ dari setiap kolom, sedangkan untuk kriteria biaya, nilai $\text{Min}_i X_{ij}$ dari setiap kolom dibagi dengan nilai X_{ij}
- c. Hasil dari nilai rating kinerja ternormalisasi (r_{ij}) membentuk matrik ternormalisasi (R)

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ r_{21} & r_{22} & \ddots & r_{2j} \\ \vdots & \vdots & \vdots & \vdots \\ r_i & r_{i2} & \dots & r_{ij} \end{pmatrix} \quad \dots \text{Persamaan (2.4)}$$

- d. Hasil akhir nilai preferensi (V_i) diperoleh dari penjumlahan dari perkalian elemen baris matrik ternormalisasi (R) dengan bobot preferensi (W) yang bersesuaian elemen kolom matrik (W).

$$V_i = \sum_{j=1}^n W_j r_{ij} \quad \dots \text{Persamaan (2.5)}$$

Keterangan :

V_i = rangking untuk setiap alternatif

W_j = nilai bobot dari setiap kriteria

r_{ij} = nilai rating kinerja ternormalisasi

- e. Dikatakan kriteria keuntungan apabila nilai X_{ij} memberikan keuntungan bagi pengambil keputusan, sebaliknya kriteria biaya apabila X_{ij} menimbulkan biaya bagi pengambil keputusan.
- f. Apabila berupa kriteria keuntungan maka nilai X_{ij} dibagi dengan nilai $\text{Max}_i X_{ij}$ dari setiap kolom, sedangkan untuk kriteria biaya, nilai $\text{Min}_i X_{ij}$ dari setiap kolom dibagi dengan nilai X_{ij}
- g. Hasil dari nilai rating kinerja ternormalisasi (r_{ij}) membentuk matrik ternormalisasi (R)

- h. Metode SAW memiliki beberapa kelebihan (Kusumadewi, 2006)
 1. Menentukan nilai bobot untuk setiap atribut kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif.
 2. Penilaian akan lebih tepat karena didasarkan pada nilai kriteria dari bobot preferensi yang sudah ditentukan.
 3. Adanya perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai *benefit* dan *cost*).
- i. Metode SAW memiliki beberapa kekurangan (Kusumadewi, 2006)
 1. Digunakan pada pembobotan lokal.
 2. Perhitungan dilakukan dengan menggunakan bilangan *crisp* maupun *fuzzy*.

1.3.5 Perumahan

Menurut UU No. 4 Tahun 1992 tentang Perumahan dan Permukiman, perumahan berada dan merupakan bagian dari permukiman, perumahan adalah kelompok rumah yang berfungsi sebagai lingkungan tempat tinggal atau lingkungan hunian yang dilengkapi dengan prasarana dan sarana lingkungan (pasal 1 ayat 2).

Menurut Ilham (2017), pembangunan perumahan diyakini juga mampu mendorong lebih dari seratus macam kegiatan industri yang berkaitan dengan bidang perumahan dan permukiman (sumber: kebijakan dan strategis nasional dan departemen permukiman dan prasarana permukiman).

1. Lahan atau tanah yang diperuntukkan untuk pemukiman itu, dimana kondisi tanah akan mempengaruhi harga satuan rumah yang akan dibangun diatas lahan itu.
2. Prasarana pemukiman, yaitu jalan lokal, saluran drainase, saluran air kotor, saluran air bersih, serta jaringan listrik yang semuanya juga menentukan kualitas pemukiman yang akan dibangun.
3. Perumahan (tempat tinggal) yang akan dibangun.
4. Fasilitas umum dan fasilitas sosial yaitu fasilitas pendidikan, kesehatan, peribadatan, lapangan bermain, dan lain lain dalam lingkungan.

1.3.6 *Hyper Text Markup Language (HTML)*

Menurut Kustiyahningsih (2011), *Hyper Text Markup Language (HTML)* adalah *text file* murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai *web page*. File-file HTML ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh *browser* yang ada di komputer *client (user)* sehingga isi informasinya dapat di tampilkan secara visual di komputer pengguna.

Contoh simple HTML

```
<html>
<body>
<p> Hello World!!! </p>
</body>
</html>
```

1.3.7 *Cascading Style Sheet (CSS)*

CSS digunakan untuk mendeskripsikan penyajian dari dokumen yang dibuat dalam bentuk *mark up language*. CSS merupakan sebuah dokumen yang berguna untuk melakukan pengaturan pada komponen halaman web (Bahtiar, dkk, 2012).

Menurut Abdulloh (2018), *Cascading Style Sheet* adalah dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga tampil dengan berbagai gaya yang diinginkan. *Cascading Style Sheet* memodifikasi HTML dengan memilih elemen HTML yang akan diatur kemudian memberikan properti yang sesuai dengan tampilan yang diinginkan.

1.3.8 *Hypertext Preprocessor (PHP)*

PHP adalah *server-side scripting language* yang awalnya dirancang untuk pengembangan web untuk menghasilkan halaman web yang dinamis (Nixon, 2014). PHP disisipkan ke dalam HTML untuk menghasilkan suatu halaman web yang dinamis. PHP dapat digunakan hampir di semua *web server*, di hampir semua sistem operasi dan platform, dan dapat digunakan dengan banyak sistem manajemen *database* relasional (RDBMS).

Kustiyahningsih (2011) menjelaskan bahwa PHP adalah skrip bersifat *server-side* yang ditambahkan ke dalam HTML. Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini client menggunakan kode-

kode PHP untuk mengirimkan permintaan ke *server*. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*.

1.3.9 MySQL

Abdhu Kadir, (2013) menyatakan bahwa MySQL adalah salah satu jenis *database server* yang sangat populer, hal itu disebabkan karena MySQL menggunakan SQL sebagai bahan dasar untuk mengakses *database*-nya. Selain itu, MySQL juga bersifat *open source*. Perangkat lunak yang populer digunakan yaitu phpMyAdmin yang merupakan perangkat lunak bebas biaya yang berguna untuk keperluan administrasi basis data MySQL.

Dengan demikian dapat disimpulkan bahwa MySQL adalah sebuah program pembuat basis data (*database*) yang bersifat *open source*, yang berarti siapa saja dapat menggunakan program tersebut. Selain itu MySQL mendapat dukungan dan banyak komunitas karena mudah dalam hal pengaturan basis data, stabil, dan pengamanan yang baik.

MySQL dalam operasi *client-server* melibatkan *server* daemon MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan besar. SQL singkatan dari *Structure Query Language* dan sering disebut *Sequel* saja.

SQL mulai dikembangkan tahun 70-an di laboratorium IBM, Stan Jose, California. Untuk mengakses sebuah *file database*, salah satu server *database* yang kecil dan mudah digunakan namun memiliki kehandalan dan *performa* tinggi.

MySQL merupakan RDBSM (*Relational Database Management System Management*). RDBSM adalah program yang memungkinkan penggunaan database untuk membuat, mengelola dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada *database* memiliki relasi antara suatu tabel dengan tabel lainnya.

1.3.10 Codeigniter

CodeIgniter merupakan salah satu *framework* PHP yang populer. *CodeIgniter* tergolong *framework* dengan ukuran kecil dan cukup mudah dikuasai. *codeigniter* juga datang dengan manual yang tergolong lengkap. *Codeigniter* merupakan aplikasi sumber terbuka yang berupa *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan *developer* untuk membuat aplikasi *web* dengan cepat mudah dibandingkan dengan membuatnya dari awal.

Fungsi penggunaan *CodeIgniter* pertama adalah *CodeIgniter* akan menghasilkan suatu struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file PHPnya dikarenakan *CodeIgniter* dibangun berbasis MVC (*Model, View, Controller*) yang memisahkan antara tampilan dan logika aplikasi.

1. *Model* adalah bagian yang bertanggung jawab terhadap operasi *database*, baik itu *create, read, update, dan delete*. *Model* berupa fungsi-fungsi operasional *database* yang dapat dipanggilkan oleh *controller*.
2. *View* adalah bagian menangani tampilan, bagian inilah yang bertugas untuk mempresentasikan data kepada *user*. *View* berbentuk struktur HTML yang berisikan variabel data yang dikirimkan oleh *Controller*.

Controller adalah bagian yang mengatur hubungan antar *Model* dan *View*. *Controller* adalah otak dari kinerja aplikasi. *Controller* terdiri dari fungsi-fungsi yang bersifat operasional dan logikal (Nur, 2017).

1.3.11 Database

Database merupakan data-data yang terkumpul dan tersusun dengan aturan dan ketentuan yang semuanya saling berelasi untuk dapat mempermudah penggunaan untuk mengolah dan memperoleh informasi. Di dalam *database* terdapat sekumpulan, tabel atau arsip yang semuanya saling berhubungan dan tersimpan di dalam sebuah media elektronik.

Menurut *Connoly dan Begg* (2015), basis data adalah suatu kumpulan data yang saling berhubungan secara dan dirancang untuk memenuhi kebutuhan dari suatu organisasi.

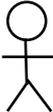
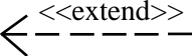
1.4 UML (*Unified Modelling Language*).

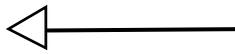
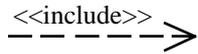
1.4.1 *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut. Sehingga pembuatan *use case* diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem (Isa dan Hartawan, 2017).

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Simbol *Use Case Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
2.		<i>Actor</i>	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i>
3.		<i>Association</i>	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
4.		<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

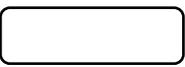
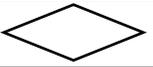
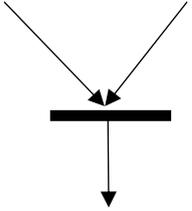
5.		<i>Generalization</i>	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
6.		<i>Include</i>	Merupakan didalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program

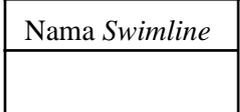
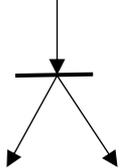
1.4.2 Activity Diagram

Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case* (Isa dan Hartawan, 2017).

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Start</i>	<i>Start point</i> , diletakan pada pojok kiri atas dan merupakan awal aktivitas
2		<i>Activities</i>	Menggambarkan suatu proses kegiatan.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
5.		<i>Final</i>	Status akhir yang dilakukan sistem.

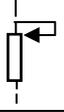
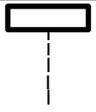
6.		<i>Swimlane</i>	Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa
7.		<i>Fork</i>	Percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel

1.4.3 Sequence Diagram

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem (Isa dan Hartawan, 2017).

Sequence diagram menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* ditunjukkan pada Tabel 2.3.

Tabel 2. 3 Simbol *Sequence Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem
2.		<i>Self message</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
3.		<i>Message</i>	Simbol mengirim pesan antar <i>class</i>
4.		<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
5.		<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

1.4.4 Class diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class Diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi kelas (*Class*), *relation associations*, *generalisation* dan *aggregation*, *attributes*, *method*, dan *visibility* tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality* (Hendini, 2016).

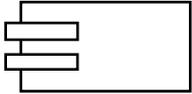
Tabel. 2.4 Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN			
1.	<table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">Nama kelas</td> </tr> <tr> <td style="padding: 2px;">+ Atribut</td> </tr> <tr> <td style="padding: 2px;">+ Operasi()</td> </tr> </table>	Nama kelas	+ Atribut	+ Operasi()	<i>Class</i>	Kelas pada struktur sistem
Nama kelas						
+ Atribut						
+ Operasi()						
2.	—————	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .			
3.	—————>	<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .			
4.	—————>	<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum – khusus).			
5.	----->	<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.			

1.4.5 Component Diagram

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini di laksanakan Irawan, dkk. (2018)

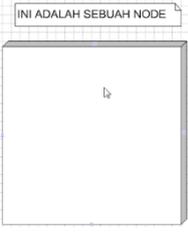
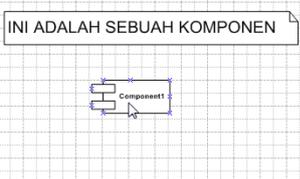
Tabel 2. 5 Simbol *Component Diagram*

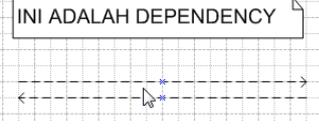
NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Component</i>	Pada <i>component diagram</i> , komponen-komponen yang ada diletakan didalam <i>node</i> untuk memastikan keberadaan posisi mereka
2.		<i>Dependency</i>	Simbol yang menjelaskan sebuah keterkaitan antara komponen, satu komponen dengan yang lain. Arah panah dalam simbol tersebut diarahkan pada komponen yang dipakai.

1.4.6 *Deployment Diagram*

Menurut Andhika (2015), *Deployment diagram* adalah salah satu jenis alat atau bahasa (*UML*) yang digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis *Object Oriented* yang akan dibangun. Simbol-simbol yang digunakan *Deployment Diagram* disajikan pada Tabel 2.6.

Tabel 2.6 Simbol *Deployment Diagram*

NO	Gambar	Nama	Keterangan
1.		Node	Node digunakan untuk menggambarkan infrastruktur apa saja yang terdapat pada sistem. Biasanya node digambarkan sebagai <i>server</i> , <i>pc</i> , dan lain-lain.
2		Component	Komponen digunakan untuk menggambarkan elemen-elemen apa saja yang terdapat pada suatu node.
3		Link	Communicates digunakan untuk menghubungkan antar node yang saling berinteraksi.

4		Dependency	Dependency digunakan untuk menggambarkan hubungan ketergantungan antar node atau komponen yang saling ketergantungan.
---	---	------------	--

1.5 Metode Pengembangan Sistem

Metode pengembangan sistem yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial (Sasmito, 2017). Metode *waterfall* memiliki tahapan-tahapan sebagai berikut :

1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

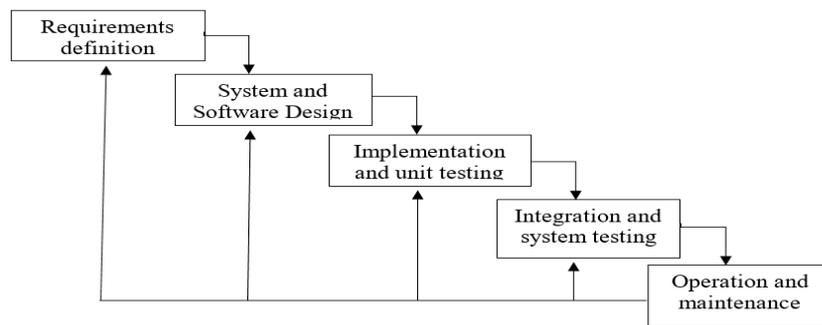
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap dengan untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian perangkat lunak dapat dikirimkan ke *customer*.

5. *Operation and maintenance*

Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapannya melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru. Bagan metode *waterfall* ditunjukkan pada Gambar 2.3.



Gambar 2.3 Metode *Waterfall*

1.6 Metode Pengujian *WebQual*

Metode *webqual* merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna. *Webqual* sudah mulai dikembangkan sejak tahun 1998 dan mengalami beberapa interaksi dalam penyusunan dimensi dan butir pertanyaannya. Dimensi dari *website* yang diinginkan oleh pengguna ada tiga yaitu dilihat dari dimensi kemudahan penggunaan (*Usability Quality*), kualitas informasi (*Information Quality*) dan kualitas interaksi layanan (*Service Interaction Quality*) (Nugraha dan Silfianti, 2016).

Model pengukuran mutu *website* dengan kuesionernya berdasarkan 3 dimensi *webqual* dapat dilihat pada Tabel 2.7.

Tabel 2.7 Indikator WebQual 4.0

Dimensi	Kuesioner Webqual 4.0
Usability	<i>I find the site easy to learn to operate</i>
	<i>My interaction with the site is clear and understandable</i>
	<i>I find the site easy to navigate</i>
	<i>I find the site easy to use</i>
	<i>The site has an attractive appearance</i>
	<i>The design is appropriate to the type of site</i>
	<i>The site conveys a sense of competency</i>
	<i>The site creates a positive experience</i>
Information Quality	<i>Provides accurate information</i>
	<i>Provides believable information</i>
	<i>Provides timely information</i>
	<i>Provides relevant information</i>
	<i>Provides easy to understand information</i>
	<i>Provides information at the right level of detail</i>
	<i>Present the information in appropriate format</i>
Service Interaction	<i>Has a good reputation</i>
	<i>It feels safe to complete transaction</i>
	<i>My personal information feels secure</i>
	<i>Creates a sense of personalization</i>
	<i>Convey a sense of community</i>
	<i>Make it easy to communicate with the organization</i>
	<i>I feel confident that goods/services will be delivered as promised</i>
Overall	Overall View of The Website

Beberapa tahap yang harus dilakukan untuk melakukan pengujian WebQual antara lain:

a. Uji Instrumen

Menurut Prakosad (2017), uji *instrument* digunakan untuk mengetahui deskripsi mengenai *variable-variabel* dalam penelitian, uji *instrument* terdiri dari uji validitas dan uji reliabilitas.

b. Uji Validitas

Validitas mengandung dua bagian yaitu bahwa instrumen pengukuran adalah mengukur secara aktual konsep dalam pertanyaan dan bukan beberapa konsep yang lain; dan bahwa konsep dapat diukur secara akurat. Oleh karena itu, suatu instrumen pengukur bisa dikatakan valid jika mengukur apa yang hendak diukur dan mampu

mengungkap data tentang karakteristik gejala yang diteliti secara tepat (Prakosad, 2017).

c. Reliabilitas

Reliabilitas adalah derajat sejauh mana ukuran menciptakan respon yang sama sepanjang waktu dan lintas situasi. Suatu alat ukur dikatakan reliabel jika hasil pengukuran dari alat ukur tersebut stabil dan konsisten. Dengan demikian reliabel adalah suatu keadaan di mana instrumen penelitian tersebut akan tetap menghasilkan data yang sama meskipun disebarkan pada sampel yang berbeda dan pada waktu yang berbeda. Uji reliabilitas akan dilakukan dengan menggunakan uji statistik *cronbach's alpha* (α) dengan ketentuan bahwa variabel yang diteliti dinyatakan reliabel apabila nilai *cronbach's alpha* (α) adalah di atas 0,6 (Prakosad, 2017)

d. Uji Asumsi Klasik

Menurut Prakosad (2017), uji asumsi klasik digunakan sebelum uji regresi liliier berganda, ada data yang harus terpenuhi agar kesimpulan dari regresi bisa menjadi kuat, antara lain uji normalitas, uji multikolinearitas, uji heteroskedastisitas dan uji autokorelasi.

e. Uji Normalitas

Pengujian ini bertujuan untuk mengetahui apakah data terdistribusi secara normal/tidak. Pengujian normalitas normalitas dapat dilihat dari titik-titik yang menyebar mengikuti garis diagonal pada *Normal PP-Plot Regression*, di mana jika titik-titik tersebut mengikuti garis diagonal maka dikatakan bahwa data yang digunakan adalah berdistribusi normal (Prakosad, 2017).

f. Uji Multikolinearitas

Pengujian ini bertujuan untuk menguji apakah dalam model regresi ditemukan adanya korelasi variabel independen. Model regresi yang baik seharusnya tidak terjadi korelasi variabel independen. Pendeteksiannya dilakukan dengan menggunakan *tolerance value* dan *Variance Inflation Factor* (VIF). Jika nilai *tolerance value* > 0.10 dan $VIF < 10$, maka tidak terjadi multikolinearitas, sebaliknya jika nilai *tolerance value* < 0.10 dan $VIF > 10$ maka terjadi multikolinearitas (Prakosad, 2017).

g. Uji Heterokedastisitas

Pengujian ini bertujuan untuk menguji apakah dalam model regresi terjadi

ketidaksamaan *variance* dari *residual* satu pengamatan ke pengamatan yang lain. Jika *variance* dari residual satu pengamatan ke pengamatan yang lain tetap maka disebut Homokedastisitas dan jika berbeda maka disebut Heterokedastisitas. Model yang baik adalah yang homokedastisitas atau tidak terjadi heterokedastisitas. Pendeteksiannya dilakukan dengan melihat grafik plot antara nilai prediksi variabel terikat yaitu ZPRED (Z predictor) dengan residualnya SRESID (*standardized residual*). Deteksi terjadinya heterokedastisitas dapat dilakukan dengan melihat penyebaran titik-titik pada sumbu Y. Jika data tersebut baik di atas maupun di bawah sumbu Y, maka dapat disimpulkan bahwa tidak terjadi gejala heterokedastisitas (Prakosad, 2017).

h. Uji Autokorelasi

Uji Autokorelasi merupakan pengujian yang dilakukan untuk menguji ada tidaknya pengaruh antara variabel pengganggu dalam masing-masing variabel bebas. Uji autokorelasi bisa menggunakan tes Durbin Watson dengan ketentuan sebagai berikut :

$dW < dL$: berarti ada autokorelasi positif (+).

$dL < dW < 4 - dL$: tidak dapat disimpulkan.

$dU < dW < 4 - dU$: berarti tidak terjadi autokorelasi.

$4 - dU < dW < 4 - dL$: tidak dapat disimpulkan.

$dW > 4 - dL$: berarti ada autokorelasi negatif (-).

Menurut Prakosad (2017) menyatakan bahwa uji autokorelasi merupakan pengujian asumsi dalam regresi dimana variabel dependen tidak berkorelasi dengan dirinya sendiri. Maksud korelasi dengan diri sendiri adalah bahwa nilai dari variabel dependen tidak berhubungan dengan nilai variabel itu sendiri, baik nilai variabel sebelumnya atau nilai periode sesudahnya. Dasar pengambilan keputusan adalah sebagai berikut :

Angka D-W di bawah -2 : berarti ada autokorelasi.

positif Angka D-W diantara -2 sampai +2 : berarti tidak ada.

autokorelasi Angka D-W di atas +2 : berarti ada autokorelasi negatif.

i. Analisis Regresi Linier Berganda

Menurut Prakosad (2017), mengemukakan analisis regresi linier berganda digunakan untuk melakukan prediksi, bagaimana perubahan nilai *variable dependen* bila nilai variabel independen dinaikan atau diturunkan nilainya. Analisis ini digunakan dengan melibatkan dua atau lebih variabel bebas antara variabel dependen (Y) dan variabel independen (X₁, X₂, dan X₃), cara ini digunakan untuk mengetahui kuatnya hubungan antara beberapa variabel bebas secara serentak terhadap variabel terkait dan dinyatakan dengan rumus.

Teknik analisis data yang digunakan dalam penelitian ini adalah teknik analisis kuantitatif dengan menggunakan model Regresi Linier Berganda (*Multiple Linier Regresion Method*) yang akan diolah dengan program *SPSS for Windows* versi 20. Metode ini digunakan untuk mengetahui pengaruh variabel bebas terhadap variabel tergantung. Rumuskan analisis regresi linier berganda sebagai berikut :

$$Y' = a + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \quad \text{persamaan (2.3)}$$

Keterangan :

Y' = Kepuasan Pengguna, yang merupakan variabel dependen.

a = Konstanta, merupakan nilai terkait yang dalam hal ini adalah Y' pada saat variabel independennya adalah 0 (X₁, X₂, X₃=0).

β_1 = Koefisien regresi berganda variabel independen X₁ terhadap variabel Y', bila variabel X₂ dan X₃ dianggap konstan.

β_2 = Koefisien regresi berganda variabel independen X₂ terhadap variabel Y', bila variabel X₁ dan X₃ dianggap konstan.

β_3 = Koefisien regresi berganda variabel independen X₃ terhadap variabel Y', bila variabel X₁ dan X₂ dianggap konstan.

X₁ = Kualitas Penggunaan yang merupakan variabel independen ke-1.

X₂ = Kualitas Informasi yang merupakan variabel independen ke-2.

X₃ = Kualitas Interaksi yang merupakan variabel independen ke-3.

j. Pengujian Hipotesis

Menurut Prakosad (2017), uji hipotesis dilakukan untuk menganalisis dan menarik kesimpulan terhadap permasalahan yang diteliti. Pengujian ini

dimaksudkan untuk mengetahui apakah terdapat pengaruh variabel bebas terhadap variabel tidak bebas. Uji hipotesis yang digunakan adalah uji F dan uji t. Berdasarkan program SPSS, jika probabilitas hasil (*p value*) dari nilai $F < 5\%$ maka hipotesis yang diajukan diterima; dan jika probabilitas hasil (*p value*) dari nilai $F > 5\%$ maka hipotesis yang diajukan ditolak. Demikian juga apabila probabilitas hasil (*p value*) dari nilai $t < 5\%$ maka hipotesis yang diajukan diterima; dan jika probabilitas hasil (*p value*) dari nilai $t > 5\%$ maka hipotesis yang diajukan ditolak.

k. Uji F (Uji Simultan)

Menurut Prakosad (2017), uji F dimaksud untuk menguji hipotesis apakah variabel-variabel bebas secara bersama-sama mempunyai pengaruh terhadap variabel tergantung. Tahap-tahap uji F adalah :

1. Merumuskan hipotesis

$$H_0 : b_1 = b_2 = b_3 = 0$$

Artinya variabel bebas secara bersama-sama tidak mempunyai pengaruh terhadap variabel tergantung.

$$H_a : b_1 = b_2 = b_3 \neq 0$$

Artinya variabel bebas secara bersama-sama mempunyai pengaruh terhadap variabel tergantung.

2. Menentukan tingkat signifikan

Tingkat signifikan yang diharapkan (α) adalah 5% dengan *degree of freedom* (df) = (k-1) dan (n-k) guna menetapkan nilai F tabel, di mana n= jumlah responden dan k = jumlah variabel bebas

3. Menentukan nilai F hitung, dengan rumus :

$$R^2 / (k - 1)$$

$$F_{hit} = \frac{R^2}{(1 - R^2) / (n - k)}$$

4. Membandingkan nilai F hitung dengan F tabel, bila:

$F_{hit} < F_{tab}$ = berarti H_0 diterima dan H_1 ditolak

$F_{hit} > F_{tab}$ = berarti H_1 diterima dan H_0 ditolak.

l. Uji T (Uji Parsial)

Menurut Prakosad (2017), uji t dimaksudkan untuk menguji hipotesis apakah variabel bebas secara parsial mempunyai pengaruh terhadap variabel tergantung. Tahap-tahap uji t adalah :

Merumuskan hipotesis

H₀: $b_1 = 0$

Artinya variabel bebas secara parsial tidak mempunyai pengaruh terhadap variabel tergantung.

H_a: $b_1 \neq 0$

Artinya variabel bebas secara parsial mempunyai pengaruh terhadap variabel tergantung.

m. Uji Koefisien Determinasi Berganda

Koefisien determinasi merupakan koefisien yang nilainya dimaksudkan untuk mengetahui seberapa besar variasi perubahan dalam satu variabel independen. Koefisien korelasi digunakan untuk menentukan koefisien determinasi. Dalam konteks ini, koefisien determinasi merupakan kuadrat dari koefisien korelasi yang dinotasi dengan R^2 Prakosad (2017). Oleh karena itu, semakin kuat korelasi diantara variabel yang diamati maka semakin besar pula koefisien determinasi yang dihasilkan. Koefisien determinasi dinyatakan dalam persen (%) sehingga harus dikalikan dengan 100%. Artinya adalah bahwa persentase dari variasi perubahan dalam variabel Y adalah disebabkan oleh adanya variasi perubahan dalam variabel

