

BAB II

LANDASAN TEORI

1.1. Tinjauan Pustaka

Pada Pada penelitian ini penulis telah melakukan tinjauan studi dari penelitian sebelumnya, adapun tinjauan studi yang diambil yaitu :

1. Penelitian yang dilakukan oleh Novan Wijaya (2017) meneliti tentang “Perancangan Aplikasi Promosi Songket Palembang Berbasis Android” dimana dalam penelitian tersebut penulis mengangkat masalah sulitnya dalam melakukan promosi mengenai kain songket Palembang masih menggunakan media cetak, brosur, dan event-event kesenian yang sering diadakan di daerah Palembang atau daerah lainnya sehingga penyampaian informasi tersebut kurang efektif dan efisien, dikarenakan perlu banyak biaya dan membutuhkan banyak waktu, dengan melakukan 2 tahap penelitian yaitu analisis sistem dan desain sistem sehingga perencanaan yang dilakukan berjalan dengan efektif dan efisien.
2. Penelitian yang dilakukan oleh Wahyudi (2015) meneliti tentang “Aplikasi Promosi Perguruan Tinggi Berbasis Multimedia Interaktif” dimana dalam penelitian tersebut peneliti ingin melakukan pengenalan perguruan tinggi agar masyarakat lebih mengenal/mengetahui tentang perguruan tinggi tersebut, penelitian ini menggunakan pendekatan R&D (Research And Development) Penelitian R&D adalah aktivitas riset dasar untuk mendapatkan informasi kebutuhan pengguna (needs assessment), kemudian dilanjutkan kegiatan

pengembangan (development) untuk menghasilkan produk dan menguji keefektifan produk tersebut.

3. Penelitian yang dilakukan oleh Hendi Wijaya dan Wellia Shintia Sari (2015) meneliti tentang “Rancang Bangun Mobile Commerce Berbasis Android Pada Toko Duta Buku Semarang” dimana peneliti merasakan permasalahan yang terjadi ketika seseorang akan berbelanja buku dan tidak mengetahui katalog elektronik, sehingga pembeli harus mengunjungi toko buku untuk mengetahui informasi mengenai buku yang dicari, berangkat dari tersebut penulis membangun aplikasi berbasis android mengenai katalog buku.
4. Penelitian yang dilakukan Dwi Prasestyo dan Khafizh Hastuti (2018) dengan judul “Penerapan Haversine Formula Pada Aplikasi Pencarian Lokasi Dan Informasi gereja Kristen di Semarang Berbasis Mobile” yang dilatar belakangi oleh sulitnya dan kurangnya informasi tempat, jadwal ibadah, jenis ibadah bagi umat kristiani, maka peneliti menerapkan metode algoritma Haversine dalam menghitung jarak tempuh terdekat pada suatu gereja.
5. Penelitian yang dilakukan oleh Ahmad Fauzi, Frengki Pernando dan Mugi Raharjo (2018) yang berjudul “Penerapan Metode Haversine Formula Pada Aplikasi Pencarian Lokasi Tempat Tambal Ban Kendaraan Bermotor Berbasis Mobile Android” dengan permasalahan kurangnya informasi untuk posisi yang akurat guna mengetahui penambalan ban yang terdapat disebuah kota, sehingga peneliti merancang aplikasi pencarian tamban ban dengan metode haversine untuk menghitung jarak antara titik permukaan bumi dengan menggunakan garis lintang (longitude) dan garis bujur (lattice).

1.2. Pengertian Android

Menurut Maiyana, E. (2018) Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

1.3. *Geolocation Information System*

Geolokasi adalah proses menemukan, menentukan, dan menyediakan lokasi yang tepat dari komputer, perangkat jaringan, atau peralatan. Ini memungkinkan lokasi perangkat berdasarkan koordinat dan pengukuran geografis, Geolokasi umumnya menggunakan Global Positioning System (GPS) dan teknologi terkait lainnya untuk menilai dan menentukan lokasi geografis. Menurut Hamidi (2018) Geolocation Information System merupakan sebuah sistem yang terdiri dari *software* dan *hardware*, data dan pengguna serta institusi untuk menyimpan data yang berhubungan dengan semua fenomena yang ada di muka bumi.

Geolocation dapat dipadukan dengan sebuah persamaan untuk menghitung jarak antara dua titik di bumi yaitu formula haversine dalam penerapannya. Pemanfaatan Geolocation dan haversine formula dapat memungkinkan untuk mengetahui lokasi berdasarkan latitude dan longitude, serta dapat melakukan perhitungan seberapa jarak terhadap suatu objek. Penelitian ini menghasilkan

sistem dengan pencarian lokasi toko sembako swadaya berdasarkan latitude dan longitude serta jarak tempuh dari suatu objek.

1.4. Haversine Formula

Dalam perancangan aplikasi ini, dengan menggunakan algoritma haversine untuk mendapatkan jarak terpendek dari dua titik di permukaan bumi berdasarkan garis bujur atau longitude dan garis lintang atau latitude.

Formula Haversine adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Formula Haversine merupakan suatu metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Berikut adalah rumus haversine:

Tabel 2.1 Rumus Haversine (Veness, 2017)

$$\Delta lat = lat2 - lat1 \quad \Delta long = long2 - long1$$

$$a = \sin^2(\Delta lat/2) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2(\Delta long/2)$$

$$c = 2 \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}) \quad d = R \cdot c$$

Dimana :

R = jari-jari bumi sebesar 6371(km)

Δlat = besaran perubahan latitude

$\Delta long$ = besaran perubahan longitude

$C =$ kalkulasi perpotongan sumbu

$d =$ jarak (km) 1 derajat = 0.0174532925 radian

1.5. Bahasa Pemrograman Java

Java menurut definisi dari Sun adalah mana untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer stand alone ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari java platform. Kata berdiri di atas sebuah mesin interpreter yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca bytecode dalam file .class dari suatu program sebagai representasi langsung dari program yang berisi bahasa mesin. Oleh karena itu, bahasa java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. Menurut Warno (2012) Bahasa Java bersifat case sensitive, sehingga harus memperhatikan penggunaan huruf besar dan kecil. Selain itu penulisan source code program tidak harus memperhatikan bentuk tertentu.

1.6. Firebase Realtime Database

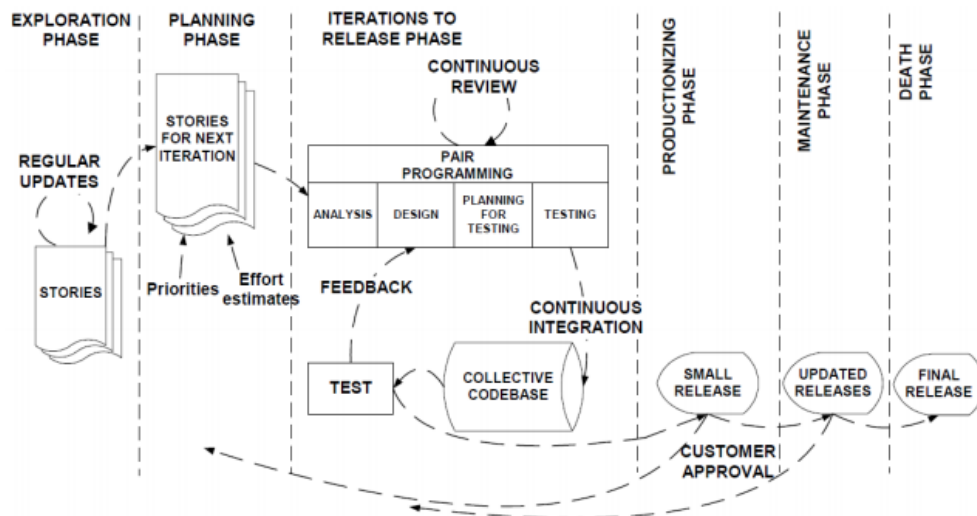
Menurut Kadir (2008) MySQL merupakan software yang tergolong sebagai DBMS (Database Management System) yang bersifat open source menyatakan bahwa software ini dilengkapi dengan source code (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk executable-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi

1.7. Metode Pengembangan Sistem

Terdapat beberapa metode dalam pengembangan sistem, salah satunya yaitu metode pengembangan *Extreme Programming*(XP). Menurut Adi S (2018) Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan–perubahan requirement yang sangat cepat.

Menurut Akbar A S (2017) Model pengembangan tradisional mengacu soal perencanaan, analisa, dan perancangan sistem, dengan waktu yang lama untuk masing-masing tahap karena luasnya cakupan, maka XP menawarkan cara yang berbeda. XP menawarkan tahapan-tahapan tersebut dalam waktu yang singkat dan berulang untuk. Berdasarkan uraian diatas maka dapat diambil kesimpulan bahwa model waterfall dimana semua proses yang dilakukan secara berurutan sesuai dengan urutan yang ada.

Berikut adalah gambaran model *Extreme Programming* :



Gambar 2.1 Metode Extreme Programming

Sumber: Akbar (2017)

Adapun tahapan pembangunan aplikasi katalog promo toko sembako swadaya dengan XP adalah sebagai berikut:

1. Planning (Perencanaan)

Tahapan ini dimulai dengan mendengarkan kumpulan kebutuhan aktifitas suatu sistem yang memungkinkan pengguna memahami proses bisnis untuk sistem dan mendapatkan gambaran yang jelas mengenai fitur utama, fungsionalitas dan keluaran yang diinginkan. Dalam pembangunan aplikasi katalog promo toko sembako swadaya berbasis android dan *geolocation information system* dengan formula *haversine* pada tahapan ini dimulai dari mengidentifikasi permasalahan yang timbul pada sistem yang sedang berjalan, kemudian dilakukan analisa kebutuhan pengguna terhadap sistem yang akan dibangun.

2. Design (Perancangan)

Pada tahapan perancangan dilakukan pembuatan pemodelan sistem berdasarkan hasil analisa kebutuhan yang didapatkan. Selain itu dibuatkan juga pemodelan basis data untuk menggambarkan hubungan antar data. Pemodelan sistem yang digunakan yaitu Unified Modelling Language (UML) yang terdiri dari beberapa diagram antara lain Use-Case Diagram, Activity Diagram.

3. Coding (Pengkodean).

Tahapan ini merupakan implementasi dari perancangan model sistem yang telah dibuat kedalam kode program yang menghasilkan prototipe dari perangkat lunak. Dalam pembangunan aplikasi katalog promo took sembako swadaya menggunakan bahasa pemrograman Java. Untuk implementasi basis data, Database Management System yang digunakan adalah Firebase.

4. Testing (Pengujian)

Tahapan ini merupakan tahapan pengujian terhadap aplikasi yang sudah dibangun, pada tahapan ini ditentukan oleh pengguna sistem dan berfokus pada fitur dan fungsionalitas dari keseluruhan sistem kemudian ditinjau oleh pengguna sistem. Metode yang digunakan dalam melakukan pengujian terhadap aplikasi katalog promo toko sembako swadaya adalah *Black-Box Testing* dengan melakukan pengujian terhadap masukan dan keluaran yang dihasilkan sistem.

5. *Software Increment* (Peningkatan Perangkat Lunak).

Tahapan ini merupakan tahap pengembangan sistem yang sudah dibuat secara bertahap yang dilakukan setelah sistem diterapkan dengan menambahkan layanan

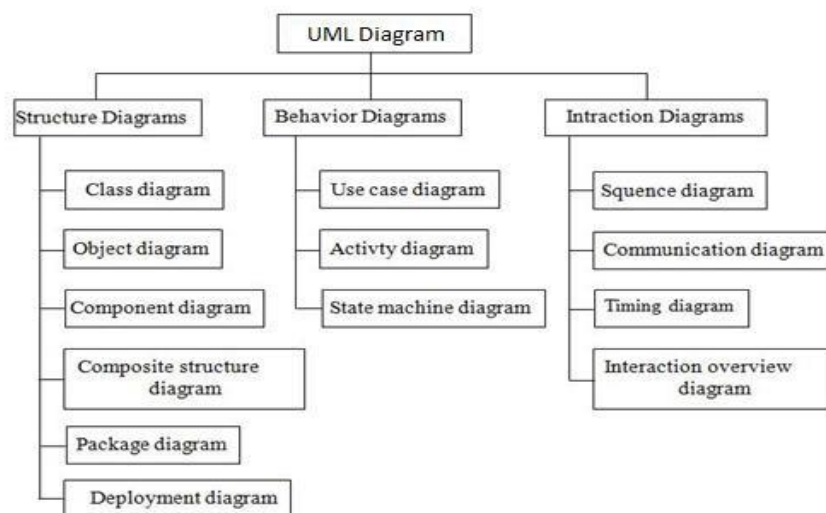
atau konten yang mengakibatkan bertambahnya kemampuan fungsionalitas dari sistem.

1.8. Bahasa Permodelan Terpadu (*Unified Modeling Language*)

Menurut Sukamto & Shalahudin (2014) Unified Modeling Language (UML) adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu diagram interchange specification, UML infrastructure, UML Superstructure, dan objek constraint language.

1.8.1. Diagram UML

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain:



Gambar 2.2 Diagram UML

Sumber: Sukamto & Shalahudin (2014)

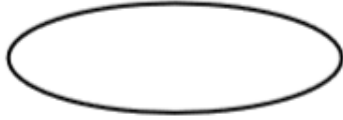


Berikut penjelasan singkat dari pembagian kategori tersebut :

1. *Structure* diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior* diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction* diagram yaitu diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem ini maupun interaksi antar subsistem pada suatu sistem.




1.8.2. Diagram Use Case

Menurut Craig Larman (2004) Diagram use case adalah gambaran yang sangat baik dari konteks sistem, yaitu menunjukkan batas suatu sistem, apa yang ada di luarnya, dan bagaimana ia digunakan. Ini berfungsi sebagai alat komunikasi yang merangkum perilaku sistem dan para aktornya. Secara besar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.2 Simbol Diagram Use Case

Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
<p>Aktor/<i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
<p>Asosiasi/<i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Tabel 2.3 Simbol Diagram Use Case (Lanjutan)

Simbol	Deskripsi
<p>Ekstensi/<i>extend</i></p> <p><i><<extend>></i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
<p>Generalisasi/<i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p>Menggunakan/<i>Include/uses</i></p> <p><i><<include>></i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.


Sumber: Sukanto R. A & Shalahudin (2014)

1.8.3. Diagram Class






Diagram kelas atau Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena class adalah deskripsi kelompok obyek-obyek dengan *property*, operasi dan relasi yang sama. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main, yaitu kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem, yaitu kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian use case, yaitu kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case.
4. Kelas yang diambil dari pendefinisian data, yaitu kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Tabel 2.3 Simbol Diagram Class

Simbol	Deskripsi
<p style="text-align: center;">Kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> <p style="text-align: center;">nama_kelas</p> <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> <p style="text-align: center;">+attribut</p> <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> <p style="text-align: center;">+operasi()</p> </div>	Kelas pada struktur sistem.
<p style="text-align: center;">Antarmuka/<i>interface</i></p> <div style="text-align: center; margin: 5px auto;">  </div> <p style="text-align: center;">nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Tabel 2.4 Simbol Diagram Class (Lanjutan)





Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Keberuntungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

1.8.4. Diagram Activity


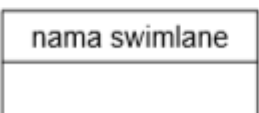
Menurut Rosa (2014) Activity Diagram adalah diagram aktivitas yang menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelempokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Tabel 2.4 Simbol Diagram Activity

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

Tabel 2.5 Simbol Diagram Activity (Lanjutan)

Simbol	Deskripsi
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Sukamto R. A., & Shalahudin (2014)

1.9. Pengujian (*Black Box Testing*)

Dalam penelitian ini dilakukan pengujian sistem, dengan menggunakan black box testing. Menurut Muslim N dan Sunyoto A (2012) Blackbox testing dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai behavioral testing, specification-based testing, *input/output*

testing atau *functional testing*. Dengan adanya *blackbox testing*, perekayasa *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. Beberapa kategori error yang akan diketahui melalui *blackbox testing* seperti:

1. Fungsi yang hilang atau tidak benar.
2. Error dari antar muka.
3. Error dari struktur data atau akses eksternal basis data.
4. Error dari kinerja atau tingkah laku.
5. Error dari insialisasi dan terminasi.

2.10. Statistik Deskriptif

Menurut Narimawati (2007) langkah-langkah yang dilakukan dalam statistik deskriptif tersebut adalah sebagai:

1. Setiap indikator yang dinilai oleh responden, diklasifikasikan dalam lima alternatif jawaban dengan menggunakan skala ordinal yang menggambarkan peringkat jawaban.
2. Dihitung total skor setiap variabel/subvariabel = jumlah skor dari seluruh indikator variabel untuk semua responden.
3. Dihitung skor setiap variabel/subvariabel = rata-rata dari total skor.
4. Untuk mendeskripsikan jawaban responden, juga digunakan statistic deskriptif seperti distribusi frekuensi dan tampilan dalam bentuk tabel atau grafik.
5. Untuk menjawab deskripsi tentang variabel penelitian ini, digunakan rentang kriteria penelitian sebagai berikut:

$$\text{Skor Total} = \frac{\text{skor aktual}}{\text{skor ideal}} \times 100\%$$

6. Skor aktual adalah jawaban seluruh responden atas kuesioner yang telah diajukan. Skor ideal adalah skor atau bobot tertinggi atau semua responden diasumsikan memilih jawaban dengan skor tertinggi. Penjelasan bobot nilai skor aktual dapat dilihat pada tabel berikut.

Tabel 2.6 Kriteria Presentase Tanggapan Responden

Jumlah Skor %	Kriteria
20,00 % - 36,00 %	Tidak Baik
36,01 % - 52,00 %	Kurang Baik
52,01 % - 68,00 %	Cukup
68,01 % - 84,00 %	Baik
84,01 % - 100 %	Sangat Baik

