

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Alfiah (2020) dalam penelitiannya menerangkan bahwa Dengan adanya sistem *website* di toko A&S yang awalnya memasarkan produk dengan cara konvensional namun untuk sekarang tidak lagi karena menggunakan sistem penjualan *website* yang memudahkan toko a&s untuk mempromosikan dan menjual produk-produk yang mereka punya dan dapat memudahkan user yang ingin membeli produk a&s dengan mudah, tanpa harus datang ke toko tersebut.

Widyastuti (2020) dalam penelitiannya menjelaskan bahwa dengan dibangunnya *website* sistem penjualan baju berbasis *web* tersebut dapat meningkatkan penjualan dengan cara media promosi untuk baju yang di jual, dengan adanya informasi penjualan produk baju berbasis *web* tersebut dapat memperluas daerah jangkauan bisnis baju tersebut.

Sinaga (2019) dalam penelitiannya menjelaskan bahwa Pembuatan *website* bisnis/usaha dapat meningkatkan popularitas bisnis secara *online* melalui pencarian di mesin pencari. Teknik SEO dapat dioptimalkan untuk meningkatkan popularitas *website* Bima Utomo *Waterpark* pada mesin pencari google, Pemilihan kata kunci/*keyword* yang tepat dapat membantu kecepatan pencarian produk maupun jasa, sehingga dapat meningkatkan brand awareness dari produk/jasa tersebut secara *online*, dan penelitian tersebut dapat dikembangkan dengan memanfaatkan teknik SEO yang lebih *advance* untuk dapat meningkatkan sensitifitas mesin pencari terhadap kata kunci *organic* maupun *non organic* yang telah ditetapkan, sehingga hasil yang diharapkan dapat bekerja lebih optimal.

2.2 Kerangka Pemikiran

Alur logika penelitian pembangunan *website* pada toko *Firefly Shoes* dengan metode SEO pada proposal ini ditunjukkan pada Gambar 2.1.



Gambar 2.1. Kerangka Pemikiran Penelitian

Latar belakang dari pembuatan *website* pada toko *Firefly Shoes* dengan SEO meliputi untuk meningkatkan penjualan toko *Firefly Shoes*, karena belum memiliki *website*, dan dengan menggunakan *website* pada toko *Firefly Shoes* dapat membantu pelanggan untuk melihat dan membeli produk melalui *website* dengan mudah. Dari latar belakang di atas bisa disimpulkan bahwa penulis ingin mengembangkan *website* berbasis *web* pada toko *Firefly Shoes* dengan *search engine optimization* (SEO), dalam melakukan pembangunan *website* tersebut ada beberapa tahap yang dilakukan, yaitu *communication*, *planning*, *modelling* dan *construction*. Berikut adalah penjelasan tiap tahap :

1. *Communication*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan pengguna demi memahami dan mencapai tujuan yang ingin dicapai. Pada tahap ini dilakukan pengumpulan data-data yang diperlukan untuk digunakan sebagai dasar dari pengembangan sistem informasi. Hasil dari komunikasi tersebut adalah inisialisasi proyek seperti menganalisis proyek seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data data yang diperlukan. Data tersebut dapat berupa studi lapangan (observasi), pengumpulan sumber sumber materi (studi pustaka) dan pencarian penelitian yang relevan. Penelitian relevan digunakan sebagai tolak ukur penulisan dan keterpaduan antara sumber-sumber materi.

2. *Planning*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan, produk kerja yang akan dihasilkan, penjadwalan kerja, dan tracking proses pengerjaan sistem

3. *Modelling*

Tahap modelling terdiri dari analisa dan perancangan, tahapan ini adalah tahap perancangan dan pemodelan arsitektursistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan antarmuka dan algoritma program.

4. *Construction*

Tahapan ini merupakan proses pembuatan *website* menggunakan PHP dan MySQL, setelah program jadi, tahap berikutnya dari *construction* yaitu melakukan uji coba pada program yang telah dibuat, yaitu dengan pengujian metode *Black box*. Dan yang terakhir adalah perbaikan pada program yang telah melewati tahap pengujian

2.3 Landasan Teori

2.3.1 Website

Menurut Sari (2020) *website* merupakan kontak suatu transaksi antara penjual dan Pelanggan dengan menggunakan media internet. Keuntungan yang diperoleh dengan menggunakan transaksi melalui *website* adalah untuk meningkatkan pendapatan dengan menggunakan penjualan online yang biayanya lebih murah dan juga sekaligus biaya-biaya operasional lainnya. Penjualan online merupakan bagian dari promosi dan promosi itu sendiri adalah salah satu bagian dari keseluruhan sistem pemasaran, maka dapat disimpulkan bahwa penjualan adalah suatu aktivitas atau kegiatan yang dilakukan oleh manusia yang saling menguntungkan satu dengan lainnya.

2.3.2 Website

Menurut Hidayat (2017) menyimpulkan bahwa *website* bisa diibaratkan sebagai sebuah rumah, toko, atau kantor. Sebuah rumah atau kantor harus memiliki alamat tetap, ada fisik bangunannya, serta ada isinya berupa ruangruang, peralatan, dan perabotan agar orang bisa beraktivitas di dalamnya. Demikian halnya dengan *website*. *Website* membutuhkan *domain name* sebagai alamatnya, *web hosting* sebagai fisik bangunannya, serta desain dan aplikasi *web* sebagai isinya. .

2.3.2.1 HTML (*Hyper Text Markup Language*)

Menurut Hidayat (2017) HTML (*Hypertext Markup Language*) adalah suatu bahasa yang digunakan untuk membuat sebuah halaman *web*. HTML itu bahasa yang fleksibel karena tidak tergantung pada suatu *platform* (sistem operasi) tertentu. HTML terdiri dari tag-tag yang mendefinisikan elemen tertentu pada sebuah halaman *web*. HTML merupakan bahasa yang tidak *case sensitive*, tidak

seperti bahasa pemrograman *server-side* seperti PHP atau ASP. HTML bisa disebut bahasa yang digunakan untuk menampilkan dan mengelola *hypertext*.

Menurut Lavarino & Yustati (2016) HTML dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN dan dipopulerkan pertama kali oleh *browser* Mosaic. Selama awal tahun 1990 HTML mengalami perkembangan yang sangat pesat. Setiap pengembangan HTML pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya. Sebelum suatu HTML disahkan sebagai suatu dokumen HTML standar, ia harus disetujui dulu oleh W3C untuk dievaluasi secara ketat. Setiap terjadi perkembangan suatu versi HTML, maka mau tak mau *browser* pun harus memperbaiki diri agar bisa mendukung kode-kode HTML yang baru tersebut. Sebab jika tidak, *browser* tak akan bisa menampilkan HTML tersebut.

2.3.2.2 PHP (PHP *Hypertext Preprocessor*).

Menurut Lavarino & Yustati (2016) Hypertext Preprocessor (PHP) adalah bahasa *serverside scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis, PHP banyak dipakai untuk pemrograman situs *web* dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirim ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman *web* lebih terjamin. PHP dirancang untuk membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*. Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

- 1) Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- 2) PHP memiliki tingkat akses yang lebih cepat.
- 3) PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
- 4) PHP juga mendukung akses ke beberapa database yang sudah ada baik yang bersifat *free/gratis* ataupun komersial. *Database* itu antara lain : MySQL, PostgreSQL, infomix, dan MicrosoftSQL Server. *Web server* yang

mendukung PHP dapat ditemukan dimana mana dari mulai Apache, IIS, AOServer, phttp. Fhttp. PWS, Lighttpd hingga Xitami dengan konfigurasi yang relative mudah.

Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.

2.3.2.3 MySQL

Menurut Lavarino & Yustati (2016) MySQL (MY Structure Query Language) adalah salah satu *Basis Data Management System* (DBMS) dari sekian banyak DBMS seperti *Oracle*, MS SQL, *Postagre SQL*, dan lainnya. MySQL berfungsi untuk mengolah Basis Data menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung atau *mensupport* dengan Basis Data MySQL.

Sedangkan MySQL merupakan Basis Data yang paling digemari dikalangan programmer web, dengan alasan bahwa program ini merupakan Basis Data yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai sebuah Basis Data server yang mampu untuk manajemen Basis Data dengan baik, mysql terhitung merupakan Basis Data yang paling digemari dan paling banyak digunakan dibanding Basis Data lainnya. Selain mysql masih terdapat beberapa jenis Basis Data server yang juga memiliki kemampuan yang juga tidak bisa dianggap enteng, Basis Data itu adalah *Oracle* dan PostgreSQL.

MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah database berisi 10,000 tabel dan 500 di antaranya memiliki 7 juta baris (kira-kira 100 *gigabyte* data). Database ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan. Walaupun memiliki kemampuan yang cukup baik, MySQL untuk sistem operasi Unix bersifat *freeware*, dan terdapat versi *shareware* untuk sistem operasi windows. Menurut pembuatnya, MySQL disebut seperti "*myessqueell*".

Sebagaimana database sistem yang lain, dalam SQL juga dikenal hierarki server dengan database-database. Tiap database memiliki tabel-tabel, tiap-tiap tabel memiliki field-field. Umumnya informasi tersimpan dalam tabel-tabel yang secara logic merupakan struktur dua dimensi terdiri atas baris dan kolom. *Field-field*

tersebut dapat berupa data seperti *int* , *realm char*, *date*, *time* dan lainnya. SQL tidak memiliki fasilitas pemrograman yang lengkap, tidak ada *looping* ataupun percabangan ,misalnya. Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman semisal C.

2.3.3 *Search Engine Optimization (SEO)*

Sinaga (2019) SEO (*Search Engine Optimization*) adalah serangkaian proses yang dilakukan secara sistematis dan bertujuan untuk meningkatkan volume serta kualitas trafik kunjungan melalui mesin pencari menuju situs *website* tertentu dengan memanfaatkan algoritma mesin pencari tersebut, yang disebut dengan *Page Rank*. Tujuan dari SEO adalah menempatkan sebuah situs *website* pada posisi teratas hasil pencarian berdasarkan kata kunci tertentu yang ditargetkan. Situs *web* yang menempati posisi teratas pada hasil pencarian memiliki peluang lebih besar untuk mendapatkan pengunjung.

Adanya SEO, suatu *website* dapat lebih meningkatkan kinerja *website* secara maksimal terutama dalam hal *organic visitor*, dan diharapkan dapat meningkatkan penjualan dari perusahaan. Menurut Artanto (2017) Secara garis besar, teknik yang digunakan dalam penerapan metode SEO ada dua, yaitu:

1) *SEO On Page*

Onpage SEO (atau lebih dikenal dengan optimalisasi halaman) merupakan sebuah teknik yang dilakukan dengan mengoptimalkan bagian bagian yang berbeda pada *website* yang pada prosesnya dapat mempengaruhi dalam hasil pencarian di mesin pencari. Berikut tahapan tahapan dari *Onpage* SEO yaitu :

- a. *Title Tags*
- b. *Meta Description*
- c. *Heading Tags*
- d. *URL Structure*
- e. *Image Optimization*
- f. *Konten*

2) *SEO Off Page*

Merupakan langkah optimasi SEO terhadap suatu *website* yang dilakukan diluar halaman (*eksternal*) *website* tersebut. Metode optimasi *Off page* sangat

erat kaitannya dengan *backlink*. *Backlink* adalah *link* dari *website* lain ke *website* yang dimaksud. Sebuah *backlink* dapat diartikan sebagai rekomendasi, semakin banyak yang merekomendasikan sebuah *website* maka akan semakin besar pula *website* tersebut dalam mendapatkan perhatian dari mesin pencari.

2.3.4 UML

Menurut Putra&Andriani (2019) UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Jenis-jenis UML yaitu ; *Usecase diagram*, *Activity diagram*, *Sequence diagram* dan *Deployment diagram*.

Adapun fungsi menggunakan UML diantaranya adalah :

- 1.) Dapat memberikan model yang siap untuk digunakan, merupakan bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
- 2.) Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai coding suatu program.
- 3.) Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
- 4.) Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun mesin.

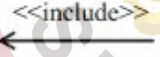
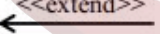
2.3.4.1 Use Case Diagram

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 Hertawan, 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 (Hendini, 2016). Simbol-simbol yang digunakan dalam *Use Case Diagram* ditunjukkan pada Tabel 2.1.

Tabel 2.1. Simbol-simbol *Use Case*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

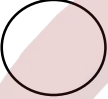
2.3.4.2 *Class Diagram*

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*, *generalitation* dan *aggregation*, *attributes*, *method*, dan *visibility* tingkat akses

objek eksternal kepada suatu operasi atau attribute (Hendini, 2016). Simbol-simbol yang digunakan *Class Diagram* ditunjukkan pada Tabel 2.2.

Tabel 2.2. Simbol-simbol *Class Diagram*

No.	Simbol	Deskripsi
1	Nama_kelas +atribut +operasi() Kelas	Kelas pada struktur sistem.
2	 Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3	 Asosiasi/association	Relasi antar kelas dengan makna umum, asosiasi juga disertai dengan <i>multiplicity</i> .
4	 Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, disertai <i>multiplicity</i> .
5	 Generalisasi	Relasi antar kelas dengan menggunakan makna generalisasi-spesialisasi.
6	 Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas.
7	 Agregasi	Relasi antar kelas dengan makna semua-bagian.

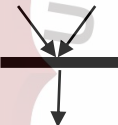
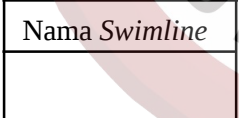
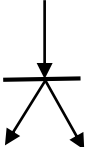
2.3.4.3 Activity Diagram

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 (Isa dan Hertawan, 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 2.3.

Tabel 2.3. Simbol-simbol *Activity Diagram* (Hendini, 2016).

N O	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> atau <i>rake</i> , digunakan untuk menunjukan adanya dekomposisi
5.		Status akhir	Status akhir yang dilakukan sistem.
6.		<i>Swimlane</i>	Pembagian <i>activity diagram</i> untuk menunjukan siapa melakukan apa
7.		<i>Fork</i>	Percabangan digunakan untuk menunjukan kegiatan yang dilakukan secara paralel untuk menggabungkan dua kegiatan paralel menjadi satu

2.3.4.4 *Sequence Diagram*

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antar objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence diagram* menggambarkan objek pada *use case* dengan mendeskripsikan

waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Isa dan Hertawan, 2017). Simbol-simbol yang digunakan dalam *Sequence diagram* ditunjukkan pada Tabel 2.4.

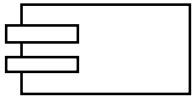
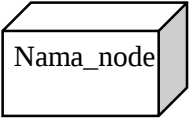
Tabel 2.4. Simbol-simbol *Sequence Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
2.		<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>
3.		<i>Control Class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
4.		<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
5.		<i>Message</i>	Simbol mengirim pesan antar <i>class</i>
6.		<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
7.		<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

2.3.4.5 *Deployment Diagram*

Deployment Diagram digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem. Simbol-simbol *deployment diagram* ditunjukkan pada Tabel 2.5.

Tabel 2.5. Simbol-simbol *Deployment Diagram* (Hendini, 2016).

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Component</i>	Pada <i>deployment diagram</i> , komponen-komponen yang ada diletakan didalam <i>node</i> untuk memastikan keberadaan posisi mereka
2.		<i>Node</i>	<i>Node</i> menggambarkan bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk <i>node</i> digambarkan sebagai sebuah kubus tiga dimensi
3.		<i>association</i>	Menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara elemen-elemen <i>hardware</i> .

2.3.5 Pengujian (*Testing*)

2.3.5.1 Metode Black Box

Hanifah (2016) Pengujian sistem bertujuan untuk melihat apakah sistem yang telah dibuat sudah sesuai dengan tujuan awal pembuatan dan layak untuk dipergunakan. Pengujian pada sistem menggunakan metode *Black Box*, tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah benar menampilkan pesan-pesan kesalahan jika terjadi kesalahan dalam penginputan data. *Black Box Testing* sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

Berikut adalah 10 tipe pengujian dari metode *Black Box* menurut :

1. *Equivalence Partitioning* : Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.
2. *Boundary Value Analysis / Limit Testing*: Mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input, Merupakan komplemen dari *Equivalence Partitioning*.
3. *Comparison Testing* : Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.
4. *Sample Testing* :
Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.
5. *Robustness Testing* :

Data input dipilih diluar spesifikasi yang telah didefinisikan, Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid

6. *Behavior Testing* :

Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data *stack*.

7. *Performance Testing* :

Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya : aliran data, ukuran pemakaian memori, kecepatan eksekusi.

8. *Requirement Testing* : Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.

9. *Endurance Testing* : Melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.

10. *Cause – Effect Relationship Testing* : Bagi-bagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja

2.3.5.2 Rancangan Kuesioner

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Kuesioner dapat berupa pertanyaan tertutup atau terbuka, dapat diberikan kepada responden secara langsung atau dikirim melalui pos atau bisa juga melalui internet. Pada penelitian ini penulis menggunakan jenis kuesioner tertutup yaitu kuesioner yang dibagikan kepada setiap responden dengan pertanyaan yang mengharapkan jawaban singkat atau responden dapat memilih salah satu jawaban alternatif dari pertanyaan yang disediakan (Sugiyono, 2017). Daftar pertanyaan kuisisioner untuk responden dapat dilihat pada lampiran 2.