

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Tinjauan dalam penelitian ini adalah artikel ilmiah yang berjudul Implementasi *Framework CodeIgniter* untuk Membangun Sistem Informasi Berbasis *Website* di SMA Kristen 2 Salatiga (Nur, 2017) Penelitian pada SMA Kristen 2 Salatiga tersebut memfokuskan pada penyelesaian masalah di SMA Kristen 2 Salatiga dalam menyampaikan informasi yang akan menghemat waktu, biaya dan tenaga. Dimana sistem informasi pendidikan sekolah di SMA Kristen 2 Salatiga masih sederhana membuat masyarakat, para siswa dan orangtua siswa kesulitan dalam mencari informasi tentang sekolah. Masyarakat yang membutuhkan informasi harus datang ke sekolah untuk bertanya secara langsung pada pihak sekolah. Sistem yang demikian menjadi kendala bagi sekolah untuk dikenal dan diketahui masyarakat luas. Siswa yang ingin mengetahui tagihan SPP (Sumbangan Pembinaan Pendidikan) harus datang ke bagian pembayaran sekolah pada jam kerja, dan mengantri dengan siswa yang lainnya. Cara kerja seperti ini juga akan membutuhkan banyak waktu dan tenaga, sehingga akan mengganggu jalannya kegiatan belajar mengajar di sekolah.

Berdasarkan masalah-masalah di atas penulis melakukan penelitian dengan mengambil judul “Implementasi *Framework CodeIgniter* untuk Membangun Sistem Informasi Berbasis *Website* di SMA Kristen 2 Salatiga”, sehingga diharapkan dengan menggunakan sistem informasi yang berbasis *web* masyarakat dan warga sekolah dapat mengakses informasi sekolah yang dapat dilakukan kapanpun dan dimanapun.

Aplikasi *website* yang akan dikembangkan menggunakan *Framework CodeIgniter* sebagai kerangka kerja, tampilan menggunakan *bootstrap*, dan *database* menggunakan MySQL. Pada penelitian ini dilakukan lima tahapan penelitian, yaitu: 1) Identifikasi Masalah. 2) Perancangan dan Pembuatan Sistem. 3) Implementasi dan Pengujian Sistem serta Analisis Hasil Pengujian. 4)

Penulisan Laporan Hasil Penelitian. Berikut contoh Tampilan *website* SMA Kristen 2 Salatiga yang disajikan pada Gambar 2.1.



Gambar 2.1 Tampilan *website* SMA Kristen 2 Salatiga

Penelitian yang kedua dilakukan oleh Fatsyahrina Fitriastuti dan Septiana Mundianarti: 2016 yang berjudul “Rancang Bangun Sistem Informasi Sekolah Berbasis *Web* menggunakan CMS Formulasi” melalui jurnal Informasi Interaktif. Penelitian ini mereka merancang sebuah Sistem informasi berbasis *web* menggunakan CMS Formulasi. Selama ini informasi mengenai SD Negeri Jlaban masih dilakukan secara manual, dalam pelayanan informasi kepada masyarakat luas sekolah ini belum memiliki *website* sekolah. Dibuatnya *website* ini dapat digunakan sebagai sarana untuk memperoleh informasi yang dibutuhkan masyarakat luas mengenai SD Negeri Jlaban. Pembuatan sistem informasi berbasis *web* ini menggunakan CMS Formulasi. CMS Formulasi memudahkan dalam pengaturan dan penyajian informasi dalam berbagai bentuk yang menarik dan memungkinkan informasi yang berbentuk dinamis yaitu dapat diubah dan diperbarui sewaktu-waktu sesuai dengan kebutuhan sekolah SD Negeri Jlaban. Metode yang digunakan dalam penelitian ini antara lain: studi literatur, pengumpulan data, SDLC(*System Development Life Cycle*) yang meliputi tahap *Analysis*, *Design*,

*Implementation, Testing, dan Maintenance.* Berikut merupakan contoh dari tampilan *website* SD Negeri Jlaban yang disajikan pada Gambar 2.2.



Gambar 2.2 Tampilan *website* SD Negeri Jlaban

Penelitian yang ketiga dilakukan oleh Fuat Hermawan: 2014 dalam Tugas Akhir Skripsi yang berjudul “Pengembangan dan Analisis Kualitas Sistem Informasi Kegiatan Ekstrakurikuler di SMK Negeri 1 Pandak”. Sistem informasi Kegiatan Ekstrakurikuler di SMK Negeri 1 Pandak dikembangkan untuk mempermudah pengelolaan data dan penyebaran informasi. Penelitian ini bertujuan untuk: 1) mengetahui cara dan langkah pengembangan sistem informasi kegiatan ekstrakurikuler dan 2) mengetahui kualitas perangkat lunak yang dikembangkan menggunakan standar kualitas ISO-9126.

Metode yang digunakan dalam penelitian ini adalah *Research and Development* (R&D) dengan model pengembangan *Waterfall*, yaitu 1) Analisis Kebutuhan; 2) Desain; 3) Implementasi; dan 4) Pengujian. Hasil dari penelitian ini adalah: 1) Sistem informasi di SMK Negeri Pandak Bantul dikembangkan dengan

menggunakan *Framework CodeIgniter* dan *CSS Bootstrap* serta mengacu pada model pengembangan *Waterfall*. Sistem ini memiliki fungsi untuk: manajemen admin, manajemen guru dan manajemen siswa dalam kegiatan ekstrakurikuler. Berikut merupakan tampilan SIMEKSKUL yang dapat dilihat pada Gambar 2.3.

MENU

Anda terakhir login pada: 2014-07-10 13:13:12.

**SIMKESKUL**  
SMK Negeri 1 Pandak

**SELAMAT DATANG DI SIMKESKUL**  
Sistem Informasi Kegiatan Ekstrakurikuler SMK Negeri 1 Pandak

NIS Siswa : 1236  
Nama Siswa : GG  
Kelas : X BB 1

**PENGUMUMAN**

Hari/Tanggal	Nama Ekstra	Pengumuman
12-06-2014	Sepakbola	Bawa Bola

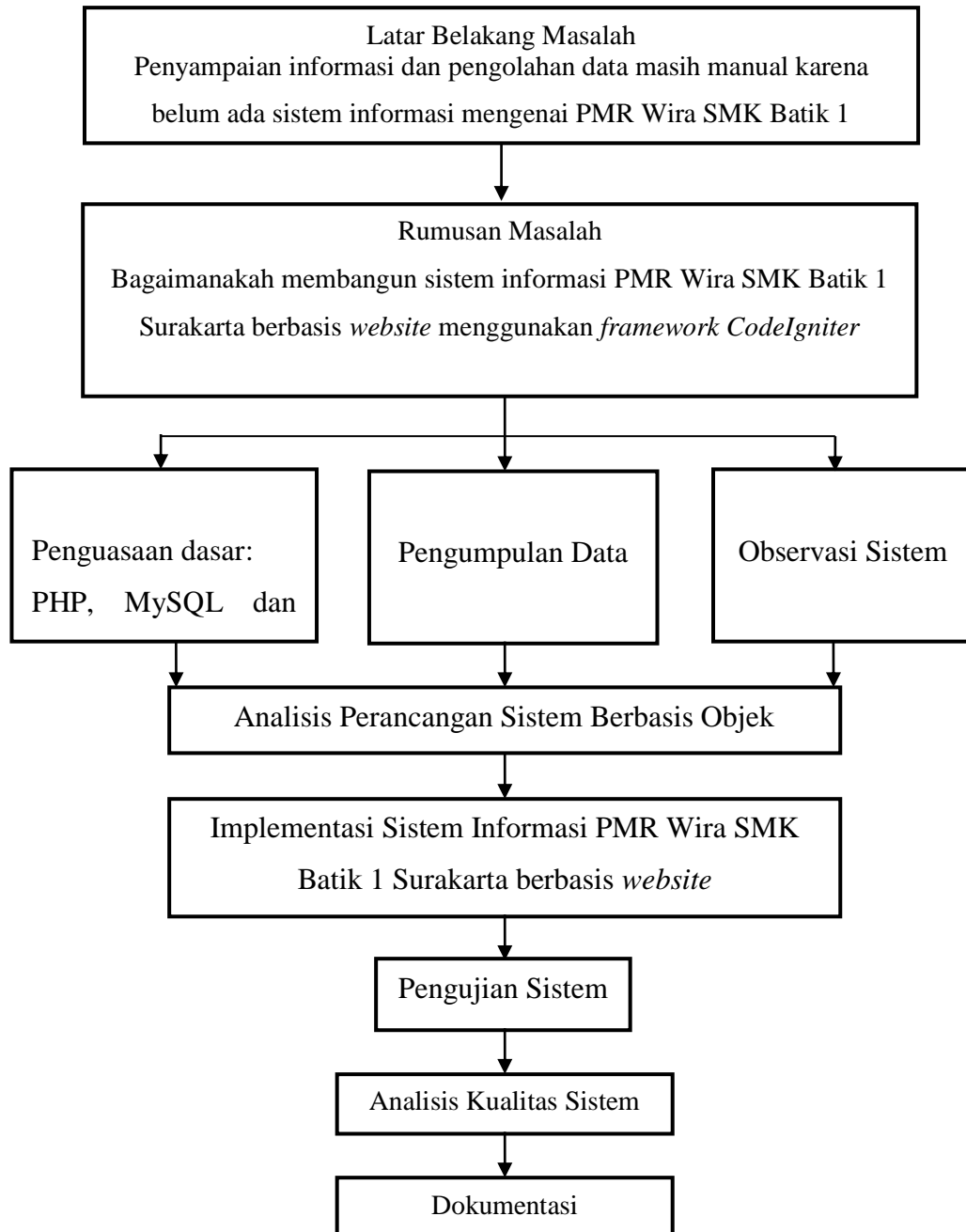
Gambar 2.3 Tampilan Sistem Informasi Ekstrakurikuler SMKN 1 Pandak

Berdasarkan tiga jurnal penelitian tersebut, dapat diketahui bahwa pada penelitian yang pertama memiliki persamaan dengan penelitian ini yaitu menggunakan *framework CodeIgniter* dalam perancangan sistem informasi. Kekurangan dari sistem informasi pada penelitian pertama adalah belum adanya *fitur* pendaftaran *online*. Penelitian kedua membangun sistem informasi sekolah berbasis *web* menggunakan CMS Formulasi. Penelitian tersebut memiliki kekurangan dalam penggunaan CMS yaitu penggunaan fungsi dan *fitur website* yang terbatas, sehingga dalam pengembangan *website*. Penelitian yang ketiga dilakukan pengembangan dan analisis kualitas sistem informasi kegiatan ekstrakurikuler di SMK Negeri 1 Pandak yang menggunakan *framework CodeIgniter*. Terdapat kekurangan dari sistem informasi tersebut mengenai fungsi-fungsi pelaporan data dari sistem informasi tersebut. Sehingga dari ketiga

referensi penelitian diatas dapat dibangun sebuah sistem informasi yang dapat menutup kekurangan dari penelitian terdahulu.

## 2.2. Kerangka Pemikiran

Kerangka pemikiran dari penelitian tugas akhir ini telah tercantum pada Gambar 2.4.



Gambar 2.4 Kerangka Pemikiran

Berdasarkan Gambar 2.4 maka kerangka pemikiran dapat diuraikan sebagai berikut:

#### 1. Latar Belakang

Pokok permasalahan yang mendasari perlu dibuatnya sistem informasi PMR Wira SMK Batik 1 Surakarta berbasis *website*. Pada *website* yang telah dimiliki SMK Batik 1 Surakarta hanya menampilkan informasi sekolah dan kegiatan yang ada. Oleh sebab itu *user* atau pengguna internet kesulitan dalam mencari informasi tentang PMR Wira SMK Batik 1 Surakarta, sehingga dibutuhkan sebuah *website* resmi sebagai media informasi yang cepat, tepat, dan akurat. Selama ini PMR Wira SMK Batik 1 Surakarta kesulitan dalam memperkenalkan diri pada masyarakat luar. Permasalahannya adalah penyampaian informasi mengenai kegiatan PMR Wira SMK Batik 1 Surakarta selama ini hanya menggunakan lembaran brosur yang dibagikan ke instansi atau perorangan sehingga tidak mencakup semua orang dan membutuhkan tenaga dan waktu yang lama. Semua kegiatan salah satunya *open recruitmen* atau pendaftaran anggota baru masih dilakukan secara manual, siswa harus datang ke ruang sekretariat PMR kemudian akan diberikan sebuah formulir pendaftaran yang akan dikumpulkan ke PMR, setelah itu dilakukan pendataan secara semi manual menggunakan *Microsoft Word*, sehingga kurang efektif dan efisien.

#### 2. Rumusan Masalah

Berdasarkan uraian pada latar belakang maka rumusan masalah penelitian Tugas Akhir ini adalah “Bagaimana membangun Sistem Informasi PMR Wira SMK Batik 1 Surakarta berbasis *website* menggunakan *Framework CodeIgniter*.”

#### 3. Penguasaan Dasar

Pada penelitian dilakukan percobaan untuk membuat sistem agar lebih menguasai *platform* yang digunakan untuk membuat sistem dan kinerja sistem. Penguasaan dasar PHP, MySQL, dan *CodeIgniter* merupakan kunci untuk membangun Sistem Informasi PMR Wira SMK Batik 1 Surakarta berbasis *website*, karena pembuatan sistem tersebut menggunakan PHP sebagai bahasa pemrograman, MySQL sebagai database sistem, dan *CodeIgniter* untuk *Framework* dalam pembuatan sistem.

#### 4. Pengumpulan Data

Tahap pengumpulan data dalam penelitian ini menggunakan metode observasi, wawancara, dan studi literatur. Tujuan dari pengumpulan data ini adalah untuk mengetahui kebutuhan dari PMR Wira SMK Batik 1 Surakarta.

#### 5. Observasi Sistem

Pada penelitian dilakukan pengamatan terhadap sistem yang sejenis sebagai referensi dalam membangun sistem informasi ini.

#### 6. Analisis Perancangan Sistem Berbasis Objek

Analisis dan perancangan pada penelitian ini dilakukan dengan bagaimana sistem informasi ini nantinya dibuat untuk memecahkan permasalahan yang ada.

#### 7. Implementasi sistem informasi PMR Wira SMK Batik 1 Surakarta

Penelitian akan dilakukan implementasi Sistem Informasi PMR Wira SMK Batik 1 Surakarta berbasis *website* menggunakan *framework CodeIgniter*.

#### 8. Pengujian Sistem

Tahap pengujian sistem akan dilakukan setelah sistem telah berhasil dibuat. Pengujian sistem dilakukan untuk mengetahui kelemahan dan kelebihan dan kelayakan dari sistem yang digunakan.

#### 9. Analisis Kualitas Sistem

Penelitian akan dilakukan analisis kualitas dari sistem yang sesuai dengan standar WebQual, untuk mengetahui sejauh mana sistem dapat memenuhi standar kualitas yang telah ditetapkan.

#### 10. Dokumentasi

Sistem yang telah diimplementasikan, dilakukan uji coba dan analisis kualitas akan diterapkan pada PMR Wira SMK Batik 1 Surakarta, kemudian akan dilakukan dokumentasi dari keseluruhan penyusunan Tugas Akhir.

### 2.3. Teori Pendukung

Teori pendukung menjelaskan mengenai teori yang dibutuhkan dalam pembuatan sistem informasi PMR Wira SMK Batik 1 Surakarta, seperti pengertian rancang bangun, sistem, sistem informasi, organisasi, Palang Merah Remaja (PMR) dan *website*.

### **2.3.1. Rancang Bangun**

Menurut (Fajriyah, dkk., 2017) Pengertian rancang bangun merupakan istilah umum untuk membuat atau mendesain sebuah objek dari awal pembuatan sampai akhir pembuatan.

### **2.3.2. Sistem**

Sistem adalah kumpulan dari beberapa elemen yang berinteraksi untuk tercapainya sebuah tujuan tertentu. Informasi juga berarti kumpulan data yang di olah menjadi bentuk yang lebih berguna dan lebih berarti yang menerimanya (Nuraini, 2015).

### **2.3.3. Informasi**

Informasi merupakan sebuah data yang telah diolah ke dalam bentuk yang lebih memiliki arti dan dapat digunakan untuk pengambilan keputusan. Data merupakan kumpulan fakta yang mewakili suatu keadaan, kondisi, atau peristiwa yang terjadi atau ada di dalam atau di dalam atau di lingkungan fisik organisasi (Prayitno, 2015).

### **2.3.4. Sistem Informasi**

Sistem informasi merupakan kumpulan dari subsistem yang saling berhubungan antara yang satu sama lain dan bekerjasama untuk mengolah data menjadi informasi yang berguna (Kadir, 2014).

### **2.3.5. Organisasi**

Menurut (Pratama, 2017) Organisasi atau *Organization* adalah struktur sosial formal, stabil, yang mengambil sumber daya dari lingkungan dan memprosesnya untuk menciptakan hasil. Modal dan tenaga kerja menjadi faktor utama yang disediakan lingkungan. Organisasi mengubah *input* tersebut kedalam bentuk barang atau jasa melalui fungsi produksi. Barang atau jasa ini dikonsumsi oleh lingkungan dan sebagai timbal baliknya, lingkungan akan menyediakan kembali faktor produksi tersebut.



### **2.3.6. Palang Merah Remaja**

Menurut (PMI, 2008), pengertian Palang Merah Remaja (PMR) adalah wadah pembinaan dan pengembangan anggota remaja Palang Merah Indonesia, yang selanjutnya disebut PMR. Terdapat di PMI Cabang di seluruh Indonesia, anggota yang ada lebih dari 3 juta orang, anggota PMR menjadi salah satu kekuatan PMI dalam melaksanakan kegiatan-kegiatan kemanusiaan dibidang kesehatan dan siaga bencana, mempromosikan prinsip-prinsip Dasar Gerakan Palang Merah dan Bulan Sabit Internasional, serta mengembangkan kapasitas Organisasi PMI.

Cara untuk menjadi anggota PMR, dapat bergabung dengan ekstrakurikuler PMR di sekolah juga bisa mengikuti kegiatan PMR di luar sekolah yang ada di PMI Cabang di daerah. Tingkatan PMR dibedakan menjadi tiga, yaitu:

- a. PMR Mula, bagi yang berumur 10-12 tahun atau relawan yang setara dengan Sekolah Dasar (SD), yang identik dengan warna hijau.
- b. PMR Madya, bagi yang berumur 12-15 tahun atau relawan yang setara dengan Sekolah Menengah Pertama (SMP), yang identik dengan warna biru.
- c. PMR Wira, bagi yang berumur 15-17 tahun atau relawan yang setara dengan Sekolah Menengah Atas (SMA), yang identik dengan warna kuning.
- d. Korps Sukarela (KSR) atau dikenal sebagai KSR PMI merupakan kesatuan di dalam sebuah perhimpunan PMI, yang merupakan wadah kegiatan atau wadah pengabdian bagi anggota biasa perhimpunan PMI yang siap menyatakan diri menjadi anggota KSR PMI serta telah memenuhi syarat menjadi anggota KSR PMI.
- e. Tenaga Sukarela (TSR) merupakan individu yang secara sukarela, tanpa paksaan dan sadar meluangkan waktu, bersedia menyumbangkan tenaga, pikiran, materi dan ketrampilan/ keahlian khusus yang dimiliki baik yang diperoleh melalui tingkat pendidikan formal maupun non formal.

PMR Wira merupakan satuan unit atau tempat kegiatan kepalang merahan yang berada dibawah naungan PMI dibidang kemanusiaan dan setiap kegiatannya mempunyai tugas yang bervariasi sesuai dengan kebutuhan yang dibutuhkan

masyarakat umum seperti kegiatan Donor Darah, kegiatan Dompot Kemanusiaan untuk Korban Bencana Alam, serta kegiatan sosial lainnya yang terkait dengan kemanusiaan.

Manajemen PMR adalah sebuah proses untuk dilakukan pembinaan dan pengembangan anggota remaja PMI agar dapat mendukung peningkatan kapasitas organisasi dan pelayanan PMI. Manajemen PMR memiliki tujuan untuk dapat membangun serta mengembangkan sebuah karakter PMR yang berpedoman pada Prinsip kepalang merahan untuk menjadi relawan masa depan. Siklus manajemen PMR yaitu Perekrutan, Pelatihan, Tri Bakti PMR, Pengakuan dan Penghargaan.

### **2.3.7. Website**

*Website* merupakan fasilitas yang tersedia di internet yang menghubungkan file dokumen dalam lingkup lokal maupun jarak jauh. Dokumen yang ada pada *website* disebut dengan *link* dan *web page* dalam *website* sehingga pengguna bisa berpindah dari satu halaman ke halaman lain (*hyper text*), diantara halaman yang disimpan dalam *server* yang sama diseluruh dunia. Halaman diakses dan dibaca melalui *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* dan aplikasi *browser* lainnya (Fritz, 2014).

## **2.4. Software Pendukung**

Pada sub bab ini merupakan penjelasan teori mengenai *software* pendukung yang digunakan dalam penelitian ini seperti HTML, PHP, *database*, MySQL, CSS, *Bootstrap*, dan *CodeIgniter*.

### **2.4.1. HTML**

HTML atau *Hypertext Markup Language* merupakan bahasa pemrograman yang bisa digunakan untuk membuat aplikasi berbasis *web*. Bahasa pemrograman ini ditulis dalam berkas format ASCII, supaya dapat menghasilkan tampilan wujud yang terintegrasi (Maya, dkk., 2015).

### **2.4.2. PHP**

Menurut (Hakim, 2014), PHP adalah bahasa pemrograman yang dapat digunakan untuk membuat sebuah halaman berbentuk HTML. File *.php* yang

dibuat akan diproses di dalam *server*, sedangkan halaman tersebut akan dikirim ke *browser* pengunjung hanyalah berupa tampilan HTML-nya. Halaman *website* yang dibuat menjadi bentuk dinamis, yakni dapat selalu berubah tanpa harus mengubah isi *website* secara manual. Informasi akan diproses ulang oleh *web server* sehingga akan didapatkan isi paling mutakhir dari halaman *web*.

### 2.4.3. Database

Database adalah sistem terkomputerisasi yang memiliki tujuan untuk memelihara data yang sudah diolah menjadi informasi dan membuat informasi tersedia saat dibutuhkan. Jadi database adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak bisa dipisahkan dengan kebutuhan akan database apapun bentuknya, baik berupa *file* teks ataupun *Database Management System* (DBMS) (Rosa & Shalahuddin, 2016).

### 2.4.4. MySQL

Menurut (Edy & Zaki, 2014) MySQL adalah sebuah *software database*. MySQL adalah database yang bisa berjalan sebagai *server* ataupun *client*. Produk *database* MySQL ini memiliki *General Public Licence*, dan bersifat *open source*, sehingga bebas untuk digunakan, diedarkan, maupun dikembangkan kembali tanpa harus khawatir dengan hak cipta. MySQL adalah database dengan tipe data rasional yang artinya MySQL dapat menyimpan datanya dalam bentuk tabel yang saling berhubungan.

### 2.4.5. CSS

Menurut (Hartono, 2013) CSS (*Cascading Style Sheet*) adalah kumpulan aturan-aturan pemformatan yang mengontrol tampilan konten dalam sebuah halaman *web*. Terdapat tiga jenis CSS, yaitu:

- a. *Inline style sheet* : cukup menambahkan atribut *style* di *tag* yang ingin kita berikan pemformatan.
- b. *Internal style sheet* : meletakkan aturan pemformatan dengan CSS dibagi `<head>` dari *html* dengan tambahan *tag* `<style>`.

- c. *External style sheet* : memisahkan antara file CSS dengan file HTML-nya.

#### **2.4.6. Bootstrap**

Menurut (Wahyu, 2014) *bootstrap* adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun desainer halaman *website*. Sesuai namanya, *website* yang dibuat oleh *bootstrap* memiliki tampilan halaman yang sama atau mirip dengan tampilan halaman twitter atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. Tampilan *website* yang dibuat *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik *desktop*, tablet ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun di-non-aktifkan sesuai keinginan. Dengan *bootstrap* kita juga bisa membangun *web* dinamis ataupun statis.

#### **2.4.7. CodeIgniter**

*CodeIgniter* merupakan salah satu *framework* PHP yang tangguh dan populer. *CodeIgniter* merupakan *framework* dengan ukuran kecil dan cukup mudah dikuasai. CI meskipun datang dengan manual namun memiliki fitur yang lengkap. CI merupakan aplikasi *open source* yang berupa *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun sebuah *website* dinamis dengan menggunakan bahasa pemrograman PHP. *CodeIgniter* memudahkan *developer* untuk membuat aplikasi *website* dengan cepat dibandingkan dengan membuatnya dari awal.

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

1. *Model* adalah bagian yang bertanggungjawab 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*.
3. *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).

## 2.5. Analisis Sistem

Analisis sistem adalah istilah yang mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem merupakan teknik pemecahan masalah yang menguraikan bagian komponen dengan mempelajari seberapa bagus bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka. Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi utama dalam menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Tahapan ini sangat penting karena menentukan bentuk sistem yang harus dibangun. Tahapan ini bisa merupakan tahap yang mudah jika klien sangat paham dengan masalah yang dihadapi dalam organisasinya dan tahu betul fungsionalitas dari sistem informasi yang akan dibuat. Tetapi tahap ini akan menjadi tahap paling sulit jika seorang klien tidak bisa menjabarkan kebutuhannya dan tertutup terhadap pihak luar yang ingin mengetahui detail proses-proses bisnisnya (Muhamad & Oktafianto, 2016).

## 2.6. Analisis Berorientasi Objek

Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) adalah tahapan untuk menganalisis spesifikasi kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek, apakah kebutuhan yang ada dapat diimplementasikan menjadi sebuah sistem berorientasi objek. Analisis ini dilakukan oleh orang-orang yang benar-benar memahami implementasi sistem yang berbasis atau berorientasi objek, tanpa pemahaman itu maka sistem yang dihasilkan bisa jadi tidak realistis jika diimplementasikan dengan berbasis objek. OOA biasanya menggunakan kartu CRC (*Component, Responsibility,*

*Collaboration*) untuk membangun kelas-kelas yang akan digunakan atau menggunakan UML (*Unified Modeling Language*) pada bagian diagram *use case*, diagram kelas, dan diagram objek (Rosa & Shalahuddin, 2016).

## 2.7. Metode Perancangan Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini sangat beragam (aplikasi bisnis, *real-time*, *utility* dan sebagainya) dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi.

Keuntungan menggunakan metodologi berorientasi objek adalah meningkatkan produktivitas, kecepatan pengembangan, kemudahan dan pemeliharaan, adanya konsistensi, meningkatkan kualitas perangkat lunak. Saat ini sudah banyak bahasa pemrograman berorientasi objek. Banyak orang berfikir bahwa pemrograman berorientasi objek identik dengan bahasa Java. Memang bahasa Java merupakan bahasa yang paling konsisten dalam mengimplementasikan paradigma pemrograman berorientasi objek. Namun sebenarnya bahasa pemrograman yang mendukung pemrograman berorientasi objek tidak hanya bahasa Java. Bahasa pemrograman yang mendukung pemrograman berorientasi objek antara lain: bahasa pemrograman *Smalltalk*, *Eiffel*, C++, PHP dan Java (Rosa & Shalahuddin, 2016).

## 2.8. UML

Menurut (Rosa & Shalahuddin, 2016) dijelaskan pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language (UML)*.

Ada 6 (enam) macam diagram dalam *Unified Modeling Language (UML)*, yaitu :

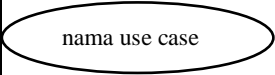
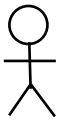

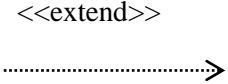
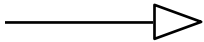
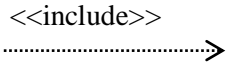
### 1. *Use Case Diagram*

Menurut (Rosa & Shalahuddin, 2016) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Syarat penamaan pada *use case* adalah nama harus simple dan mudah untuk dipahami. Ada dua hal utama pada *use case* yaitu aktor dan *use case*. Aktor berupa orang, proses, atau sistem lain yang dapat berinteraksi dengan sistem informasi yang dibuat di luar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol atau aktor adalah gambar orang namun aktor belum tentu merupakan orang. Sedangkan *use case* adalah fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor.

Berikut merupakan penjelasan mengenai simbol *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case Diagram*


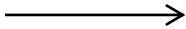
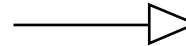

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama <i>use case</i> .
2.		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		<i>Extend</i>	Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
4.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.



## 2. *Class Diagram*

Menurut (Rosa & Shalahuddin, 2016) Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas terdiri dari atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Berikut merupakan penjelasan mengenai simbol *class diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN			
1.	<table border="1"> <tr> <td>Nama kelas</td> </tr> <tr> <td>+ Atribut</td> </tr> <tr> <td>+ 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.			

## 3. *Activity Diagram*

Menurut (Rosa & Shalahuddin, 2016) Diagram aktivitas atau *Activity Diagram* menggambarkan aktivitas dari sebuah sistem atau proses bisnis atau



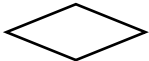


menu yang ada pada perangkat lunak. Perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan pengelompokan *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak.

Simbol *Activity Diagram* ditunjukkan pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Activity Diagram*



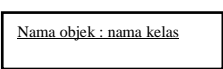

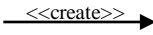
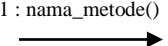
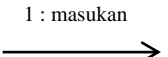
NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Status akhir	Status akhir yang dilakukan sistem.

#### 4. *Sequence Diagram*

Menurut (Rosa & Shalahuddin, 2016) *Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan

*message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen diagram maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 2.4 Tabel *Sequence Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		Aktor	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; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5.		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

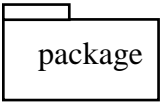
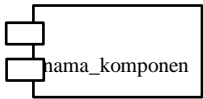
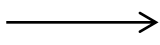


Banyak diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka

diagram sekuen yang harus dibuat juga semakin banyak. *Sequence* diagram dapat dilihat pada Tabel 2.4.

#### 5. *Component Diagram*

Menurut (Rosa & Shalahuddin, 2016) Diagram komponen atau *component diagram* menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah *user interface* yang menangani tampilan, *bussiness procesing* yang menangani fungsi-fungsi proses bisnis, *data* yang menangani manipulasi data, dan *security* yang menangani keamanan sistem. Simbol *component diagram* dapat dilihat pada Tabel 2.5.

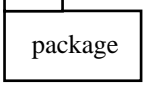
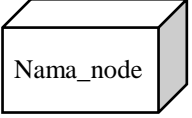
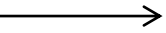

Tabel 2.5 Simbol *Component Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusuan dari satu atau lebih komponen.
2.		<i>Component</i>	Komponen sistem.
3.		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
4.		<i>Interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
5.		<i>Link</i>	Relasi antar komponen.

#### 6. *Deployment Diagram*

Berikut merupakan tabel mengenai simbol *deployment diagram* yang ditunjukkan pada Tabel 2.6.

Tabel 2.6 Simbol-simbol *Deployment*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih <i>node</i> .
2.		<i>Node</i>	Biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelum pada diagram komponen.
3.		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.		<i>Link</i>	Relasi antar <i>node</i> .

Menurut (Rosa & Shalahuddin, 2016) *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan sistem tambahan yang menggambarkan rancangan *device*, *node*, dan *hardware*. Sistem *client/server*, sistem terdistribusi murni, rekayasa ulang aplikasi.

## 2.9. Metode Pengujian WebQual

Pengujian perangkat lunak adalah sebuah elemen topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan atau *customer* (Rosa & Shalahuddin, 2016). Penelitian ini menggunakan model pengujian *WebQual* untuk mengukur kualitas dari *website* yang telah dibangun.

Menurut Iman Sanjaya dalam jurnal Penelitian IPTEK-KOM, *WebQual* merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari SERVQUAL yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. *WebQual* sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa interaksi dalam penyusunan dimensi dan butir pertanyaannya.

Teori *WebQual* yang terdapat dalam jurnal yang disusun oleh (Anwariningsih), terdapat tiga dimensi yang mewakili kualitas suatu *website*, yaitu kegunaan (*usability*), kualitas informasi (*information quality*) dan interaksi layanan (*service interaction*). Masing-masing dimensi terdiri dari beberapa pernyataan yang ditunjukkan pada Tabel 2.7 sampai Tabel 2.9.

Tabel 2.7 Dimensi Kegunaan (*usability*)

No	Deskripsi Indikator
1	<i>Website</i> mudah untuk dioperasikan.
2	Interaksi dengan <i>website</i> sangat mudah dimengerti dan tidak membingungkan.
3	Pengguna merasa mudah untuk bernavigasi dalam <i>website</i> .
4	<i>Website</i> mudah untuk digunakan.
5	<i>Website</i> memiliki tampilan yang menarik.
6	Desain <i>website</i> sesuai dengan tipe <i>website</i> .
7	<i>Website</i> menunjukkan kemampuannya.
8	<i>Website</i> dapat memberikan pengaruh / pengalaman positif bagi pengguna.

Tabel 2.8 Dimensi Kualitas Informasi (*informationquality*)

No	Deskripsi Indikator
1	<i>Website</i> menyajikan informasi yang akurat.
2	Informasi yang disajikan <i>website</i> dapat dipercaya.
3	Informasi yang disajikan tepat waktu atau <i>up to date</i> .
4	Informasi yang disajikan relevan.
5	Informasi yang disajikan mudah dipahami.
6	Informasi yang disajikan sangat detail.
7	Informasi disajikan dalam format yang sesuai.

Tabel 2.9 Dimensi Kualitas Interaksi (*interaction quality*)

No	Deskripsi Instansi
1	<i>Website</i> memiliki reputasi yang baik.
2	Pengguna merasa aman untuk melakukan transaksi atau interaksi dengan <i>website</i> ..
3	<i>Website</i> menjaga informasi pribadi pengguna.
4	<i>Website</i> memberi ruang untuk personalisasi.
5	<i>Website</i> memberi ruang untuk komunitas.
6	<i>Website</i> memudahkan pengguna untuk berkomunikasi dengan organisasi.
7	Pengguna merasa yakin dengan layanan/informasi yang disediakan karena sesuai dengan yang dijanjikan.

Tabel 2.10 Dimensi Keseluruhan

No	Deskripsi Indikator
1	Pendapat secara umum tentang <i>website</i> ini.

Beberapa tahapan yang harus dilakukan untuk melakukan pengujian *webqual* antara lain:

#### 2.9.1. Uji Instrumen

Menurut (Surya, 2017) uji instrumen digunakan untuk mengetahui deskripsi mengenai variabel-variabel dalam penelitian, uji instrumen terdiri dari uji validitas dan uji reliabilitas.

##### 2.9.1.1. Uji Validitas

Menurut (Suharsimi, 2013) validitas adalah suatu ukuran yang menunjukkan tingkat kehandalan dan kesahihan suatu instrumen. Instrumen yang valid berarti alat ukur yang digunakan untuk mendapatkan data itu valid. Menurut (Sugiyono, 2010) valid berarti instrumen tersebut dapat digunakan untuk mengukur apa yang hendak diukur. Suatu kuesioner dinyatakan valid apabila  $r$  hitung  $>$   $r$  tabel, sebaliknya kuesioner dapat dinyatakan tidak valid apabila  $r$  hitung  $<$   $r$  tabel. Pengujian validitas ini menggunakan batasan  $r$  tabel dengan signifikansi 0,05 dan uji 2 sisi.

### 2.9.1.2. Uji Reliabilitas

Menurut (Suharsimi, 2013) reliabilitas adalah sesuatu instrumen yang dapat dipercaya untuk digunakan sebagai alat pengumpul data karena instrumen tersebut sudah baik. Instrumen yang reliabel atau dapat dipercaya akan menghasilkan data yang dapat dipercaya, instrumen yang reliabel mengandung arti bahwa instrumen tersebut harus baik sehingga mampu mengungkap data yang bisa dipercaya.

Berdasarkan (Iman, 2012) terdapat aturan praktis yang dapat diterapkan terkait dengan nilai alpha, jika  $\alpha > 0,9$  berarti reliabilitas model sangat bagus,  $\alpha > 0,8$  berarti reliabilitas model bagus,  $\alpha > 0,7$  berarti reliabilitas model bisa diterima,  $\alpha > 0,6$  berarti reliabilitas model layak,  $\alpha > 0,5$  berarti reliabilitas model kurang bagus dan  $\alpha < 0,5$  berarti reliabilitas model tidak dapat diterima.

### 2.9.2. Uji Asumsi Klasik

Menurut (Yoedo Prabowo, Sulistiowati, Julianto Lemantara, 2016) uji asumsi klasik digunakan sebelum uji regresi linier berganda, ada data yang harus terpenuhi agar kesimpulan dari regresi bisa menjadi kuat, antara lain uji normalitas, uji multikolinearitas dan uji heteroskedastisitas.

#### 2.9.2.1. Uji Normalitas

Menurut (Surya, 2017) uji normalitas ini bertujuan untuk menguji apakah dalam model regresi, variabel pengganggu atau residual memiliki distribusi secara normal. Uji T dan Uji F mengasumsikan bahwa residual mengikuti distribusi normal, kalau asumsi ini dilanggar maka uji statistik menjadi tidak valid.

Salah satu cara untuk mendeteksi apakah residual berdistribusi normal atau tidak yaitu dengan uji statistik *one-sample kologorov-smirnov*. Dasar pengambilan keputusan dari *one-sample kologorov-smirnov* adalah:

1. Jika hasil *one-sample kologorov-smirnov* diatas tingkat signifikansi 0,05 menunjukkan pola distribusi normal, maka model regresi tersebut memenuhi asumsi normalitas.



2. Jika hasil *one-sample kologorov-smirnov* dibawah tingkat signifikansi 0,05 menunjukkan pola distribusi normal, maka model regresi tersebut tidak memenuhi asumsi normalitas.

#### 2.9.2.2. Uji Multikolinearitas

Menurut (Surya, 2017) pengujian multikolinearitas bertujuan untuk menguji apakah model regresi ditemukan adanya korelasi antara variabel bebas (*independent*). Model regresi yang baik seharusnya tidak terjadi korelasi diantara variabel *independen*. Jika variabel independen saling berkolerasi, maka variabel-variabel ini tidak ortogonal.

Variabel ortogonal merupakan variabel independen yang nilai korelasi antar sesama variabel independen sama dengan nol. Salah satu cara untuk mengetahui ada dan tidaknya multikolinearitas pada suatu model regresi adalah dengan melihat nilai *tolerance* dan VIF (*Variance Inflation Factor*).

1. Jika nilai *tolerance*  $> 0,10$  dan  $VIF < 10$ , maka dapat diartikan bahwa tidak terdapat multikolinearitas pada penelitian tersebut.
2. Jika nilai *tolerance*  $< 0,10$  dan  $VIF > 10$ , maka dapat diartikan bahwa terjadi gangguan multikolinearitas pada penelitian tersebut.

#### 2.9.2.3. Uji Heteroskedastisitas

Menurut (Surya, 2017) pengujian ini bertujuan untuk menguji apakah dalam model regresi terjadi ketidaksamaan varians 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 disebut heteroskedastisitas. Untuk mengetahui adanya heterokedastisitas adalah dengan melihat ada atau tidaknya pola tertentu dari grafik *Scatter Plot* dengan ketentuan:

1. Jika terdapat pola tertentu, seperti titik-titik yang ada membentuk pola tertentu yang teratur maka menunjukkan telah terjadi heteroskedastisitas.
2. Jika tidak ada pola tertentu, serta titik-titik menyebar diatas dan dibawah angka 0 pada sumbu Y, maka tidak terjadi heteroskedastisitas.

### 2.9.3. Analisis Regresi Linier Berganda

Menurut (Surya, 2017) metode analisis regresi linear berganda berfungsi untuk mengetahui pengaruh/hubungan variabel *independent* (bebas) dengan variabel *dependent* (terikat). Pengujian regresi linear berganda ini dilakukan dengan melakukan uji koefisien determinasi, uji parsial (Uji T) dan uji simultan (Uji F).

#### 2.9.3.1. Uji F (Uji Simultan)

Menurut (Ghozali, 2011) uji F digunakan untuk menunjukkan apakah semua variabel *independent* yang dimasukkan dalam model mempunyai pengaruh secara bersama-sama terhadap variabel *dependent*. Dasar penerimaan atau penolakan hipotesis dapat dilihat dengan membandingkan F hitung dengan F tabel, jika F hitung > F tabel maka  $H_0$  ditolak dan  $H_a$  diterima.

#### 2.9.3.2. Uji T (Uji Parsial)

Menurut (Ghozali, 2011) uji T pada dasarnya menunjukkan seberapa jauh pengaruh satu variabel *independent* secara individual dalam menerangkan variabel *dependent*. Penerimaan atau penolakan hipotesis dilakukan dengan kriteria sebagai berikut:

1. Jika nilai  $sig < 0,05$  atau  $t \text{ hitung} > t \text{ tabel}$  maka terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terikat).
2. Jika nilai  $sig > 0,05$  atau  $t \text{ hitung} < t \text{ tabel}$  maka terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terikat).

#### 2.9.3.3. Koefisien Determinasi ( $R^2$ )

Menurut (Surya, 2017) koefisien determinasi ( $R^2$ ) pada intinya mengukur seberapa jauh kemampuan model menerangkan variasi variabel *dependent*. Range nilainya antara 0-1, apabila nilai  $R^2$  kecil berarti kemampuan variabel-variabel *independent* dalam menjelaskan variasi variabel *dependent* sangat terbatas, dan sebaliknya apabila  $R^2$  besar berarti kemampuan variabel-variabel *independent* dalam menjelaskan variasi variabel *dependent* besar.