

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang serupa dan relevan dengan penelitian bertujuan untuk membuktikan keaslian penelitian, untuk tinjauan pustaka dapat dilihat pada tabel 2.1 dibawah ini:

Tabel 2.1 Penelitian Terdahulu

| | | |
|----|------------------|--|
| 1. | Judul | Implementasi Supply Chain Management (SCM) Toko Alat dan Bahan Bangunan Berbasis Web (Studi Kasus : TB . Bojong Indah) |
| | Penulis | Syamfithriani, Nugraha dan Darmawan |
| | Tahun | 2020 |
| | Metode | Waterfall |
| | Hasil Penelitian | Sistem informasi ini dapat membantu TB. Bojong Indah untuk mengelola persediaan barang masuk dan keluar, melakukan pemesanan barang, retur barang, penawaran barang dari Supplier, proses validasi pembayaran dan mempermudah penyampaian informasi antara TB. Bojong Indah dengan Supplier maupun sebaliknya. |
| 2. | Judul | Perancangan Dan Implementasi Sistem Informasi Supply Chain Management Pada Produksi Plastik Kemasan (Studi Kasus Pada PT Sinar Pelangi Kemasindo) |
| | Penulis | Hedin, Purwanto Khotimah dan Latief |
| | Tahun | 2022 |
| | Metode | Waterfall |
| | Hasil Penelitian | Sistem ini memberikan kemudahan bagi manajemen perusahaan untuk mengakses proses bisnis SCM, seperti pemantauan stok bahan baku untuk kegiatan produksi dan pengawasan terhadap kegiatan produksi untuk kelancaran pengiriman barang kepada customer |
| 3. | Judul | Implementasi Supply Chain Management Pada Toko Grosir Dan Eceran Berbasis Web |
| | Penulis | Alfino, Safitri dan Jamhur |

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

| | | |
|---|------------------|--|
| | Tahun | 2022 |
| | Metode | Waterfall |
| | Hasil Penelitian | Sistem ini dapat memudahkan Toko Anugerah dalam mengontrol persediaan barang karena data persediaan barang dapat ditampilkan oleh sistem sehingga persediaan barang dapat dipantau setiap waktu. |
| 4 | Judul | Implementasi Model Rapid Application Development Pada Sistem Informasi Persediaan Barang Dengan Metode FIFO |
| | Penulis | Siregar |
| | Tahun | 2020 |
| | Metode | RAD (Rapid Application Development) |
| | Hasil Penelitian | Implementasi dengan sistem RAD untuk mewujudkan sistem informasi persediaan barang telah mampu menghasilkan produk dengan hasil yang mumpuni. Aplikasi yang dirancang telah dapat melengkapi keperluan transaksi dan mengelola informasi persediaan barang dengan metode FIFO. |
| 5 | Judul | Rancang Bangun Sistem Informasi Inventory Pada Salon Kecantikan |
| | Penulis | Mufida, Rahmawati, dan Hertiana |
| | Tahun | 2019 |
| | Metode | Prototype |
| | Hasil Penelitian | Sistem informasi inventory yang dibuat mampu berjalan secara efektif, yaitu dengan memberikan informasi yang tepat dan cepat terhadap ketersediaan stok produk salon di gudang. |

(Syamfithriani, Nugraha, & Darmawan, 2020), melakukan penelitian tentang Implementasi Supply Chain Management (SCM) Toko Alat dan Bahan Bangunan Berbasis Web (Studi Kasus : TB . Bojong Indah). Permasalahan dalam penelitian ini adalah jumlah stock barang yang berlebih dan jumlah stock barang yang kurang barang tertentu sehingga berakibat pada berlebihnya jumlah persediaan suatu barang (upper stock) dan barang lain yang tidak memiliki

persediaan (lower stock). Tujuan penelitian ini yaitu untuk membangun Sistem Informasi pengelolaan barang dengan mengimplementasikan konsep Supply Chain Management (SCM) di TB. Bojong Indah untuk memaksimalkan pengelolaan barang dan mempermudah komunikasi antara TB. Bojong Indah dengan Supplier sehingga informasi mengenai kondisi barang dapat dengan cepat tersampaikan antara keduanya. Metode pengembangan sistem yang digunakan adalah *waterfall*. Sistem yang dibuat berbasis Web, dengan pengolah data (database) MySQL dan bahasa pemrograman yang digunakan PHP. Hasil dari penelitian ini adalah Sistem informasi ini dapat membantu TB. Bojong Indah untuk mengelola persediaan barang masuk dan keluar, melakukan pemesanan barang, retur barang, penawaran barang dari Supplier, proses validasi pembayaran dan mempermudah penyampaian informasi antara TB. Bojong Indah dengan Supplier maupun sebaliknya. Sistem informasi ini diharapkan dapat mempermudah aktivitas- aktivitas yang ada di TB. Bojong Indah.

(Hedin, Purwanto, Khotimah, & Latief, 2022), melakukan penelitian tentang Perancangan Dan Implementasi Sistem Informasi Supply Chain Management Pada Produksi Plastik Kemasan (Studi Kasus Pada PT Sinar Pelangi Kemasindo). Permasalahan dalam penelitian ini adalah PT Sinar Pelangi Kemasindo telah menggunakan sebuah sistem berbasis aplikasi sederhana, namun saat ini sistem yang digunakan perusahaan ini kurang efektif karena sistem tersebut tidak dapat memantau dan mengendalikan setiap proses bisnis yang dijalankan dan tidak dapat diakses secara online. Penelitian ini bertujuan untuk membangun sistem informasi Supply Chain Management dan mempermudah akses terhadap proses bisnis Supply Chain Management. Adapun metodologi penelitian yang digunakan

dalam perancangan sistem informasi stok barang pada CV Delta Vision Mandiri ini adalah metode *Research And Development* dan metode kepustakaan. Hasil penelitian ini yaitu sistem ini memberikan kemudahan bagi manajemen perusahaan untuk mengakses proses bisnis SCM, seperti pemantauan stok bahan baku untuk kegiatan produksi dan pengawasan terhadap kegiatan produksi untuk kelancaran pengiriman barang kepada customer.

(Alfino, Safitri, & Jamhur, 2022), melakukan penelitian tentang Implementasi Supply Chain Management Pada Toko Grosir Dan Eceran Berbasis Web. Permasalahan dalam penelitian ini adalah belum adanya sistem informasi untuk mengelola manajemen barang. Tidak hanya itu dalam pengelolaan stok produk sering terjadinya kelebihan stok dan kekurangan stok dikarenakan kurang adanya pengelolaan barang dengan baik. Penelitian ini bertujuan untuk merancang dan membangun suatu sistem untuk memenuhi semua kebutuhan konsumen dengan lengkap dan cepat, memberikan pelayanan terbaik serta harga yang terjangkau. Penelitian ini menggunakan model siklus hidup pengembangan sistem *waterfall*.. Perancangan sistem menggunakan UML (*Unified Modelling Language*) dan aplikasi yang dikembangkan berbasis *web service* dengan menggunakan *framework Codeigniter*. Hasil penelitian yang didapatkan yaitu sistem ini dapat memudahkan Toko Anugerah dalam mengontrol persediaan barang karena data persediaan barang dapat ditampilkan oleh sistem sehingga persediaan barang dapat dipantau setiap waktu.

Siregar (2020), melakukan penelitian tentang implementasi model *rapid application development* pada sistem informasi persediaan barang dengan metode FIFO. Permasalahan dalam penelitian ini adalah pengawasan terhadap persediaan

barang sangat mendesak untuk dilakukan oleh pengusaha yang bertujuan untuk mendukung pengambilan keputusan perusahaan sehingga didapatkan persediaan yang ideal. Sasaran dari hasil penelitian ini nantinya untuk meninjau sejauh mana prosedur-prosedur yang ada dan dapat menyajikan informasi sekaligus untuk menghasilkan suatu sistem informasi yang mutakhir sehingga tercapainya efisiensi kerja. Penelitian ini dilakukan dengan pola penjabaran serta pembuatan aplikasi memanfaatkan model RAD (*Rapid Application Development*) yang dilengkapi dengan bahasa pemrograman visual basic 2010 dan laporan *crystal report* dapat. Implementasi dengan sistem RAD untuk mewujudkan sistem informasi persediaan barang telah mampu menghasilkan produk dengan hasil yang mumpuni. Aplikasi yang dirancang telah dapat melengkapi keperluan transaksi dan mengelola informasi persediaan barang dengan metode FIFO.

Mufida, Rahmawati, dan Hertiana (2019), melakukan penelitian tentang rancang bangun sistem informasi inventory pada salon kecantikan. Permasalahan dalam penelitian ini adalah penanganan data yang masih dilakukan dengan menggunakan sistem manual mengakibatkan sering terjadinya kesalahan yang ada. Tujuan Penelitian ini adalah membangun sebuah sistem informasi inventory yang dapat menghasilkan sebuah laporan stok barang. Metode yang digunakan dalam pembuatan sistem yaitu metode *prototype*, bahasa pemrograman menggunakan PHP dan *MySQL* sebagai *database*. Hasil penelitian ini adalah sistem informasi inventory yang dibuat mampu berjalan secara efektif, yaitu dengan memberikan informasi yang tepat dan cepat terhadap ketersediaan stok produk salon di gudang.

Berdasarkan lima *literature* yang telah dipaparkan, maka penelitian ini memiliki perbedaan dari penelitian-penelitian terdahulu, diantaranya:

1. Pada penelitian ini membahas sistem informasi stok barang menggunakan metode FIFO (*First In First Out*) sama dengan penelitian Siregar (2020) sistem dengan metode yang sama yaitu FIFO namun pengembangan sistem yang dilakukan berbasis masih berbasis dekstop sehingga laporan yang di sajikan tidak dapat di pantau oleh pimpinan kapanpun dan dimanapun.
2. Terdapat fitur batas minimal stok sehingga terdapat notifikasi atau pengingat pada barang yang jumlah stoknya kurang dari batas minimal yang ditetapkan.
3. Metode pengembangan sistem yang dilakukan pada penelitian sebelumnya belum ada yang menggunakan model *Extreme Programming* (XP).

2.2 Sistem

Sistem sebagai urutan operasi klerikal, biasanya melibatkan beberapa orang didalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi. Adapun pendekatan yang lebih menekankan pada elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Hakim, Sukron dan Awaludin, 2019). Sistem mempunyai beberapa karakteristik menurut Samsudin, Abdurahman dan Abdullah (2019) sebagai berikut:

1. Batasan (*Boundary*)

Batas sistem merupakan garis abstraksi yang memisahkan antara sistem dan lingkungannya. Batas sistem ini bagi umat manusia sangat relatif dan tergantung kepada tingkat pengetahuan dan situasi kondisi yang dirasakan oleh orang yang melihat sistem tersebut.

2. Lingkungan (*Environment*)

Bentuk apapun yang ada diluar rung lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

3. Masukan (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan dan sinyal. Contoh, di dalam suatu unit sisten komputer, “program” adalah *maintenance* input yang digunakan untuk mengoperasikan komputernya dan “data” adalah signal input untuk diolah menjadi informasi.

4. Keluaran (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal yang lain yang menjadi input bagi sub sistem lain

2.3 Informasi

Informasi merupakan salah satu sumber daya penting dalam suatu organisasi digunakan sebagai bahan pengambilan keputusan. Sehubungan dengan hal itu, informasi haruslah berkualitas. Informasi juga dapat disimpulkan sebagai

data yang telah diproses dengan suatu cara untuk memberikan arti dan memperbaiki pengambilan keputusan (Hakim, Sukron dan Awaludin, 2019).

2.4 Sistem Informasi

Menurut Jogiyanto (2008) sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manjerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sedangkan menurut Ladjamudin (2013) sistem Informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

Berdasarkan uraian, penulis menyimpulkan bahwa sistem informasi adalah kumpulan komponen didalam suatu organisasi yang berfungsi sebagai pengolahan untuk menghasilkan sebuah laporan-laporan yang disajikan kepada pihak tertentu.

2.5 Supply Chain Management

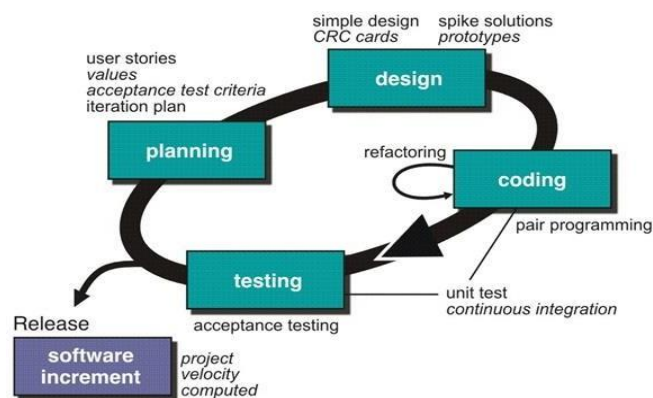
Supply Chain Management (SCM) adalah pengintegrasian sumber bisnis yang kompeten dalam penyaluran barang, mencakup perencanaan dan pengelolaan aktivitas pengadaan (Syamfithriani et al., 2020) dan logistik serta informasi terkait mulai dari tempat bahan baku sampai tempat konsumsi, termasuk koordinasi dan kolaborasi dengan jaringan mitra usaha (pemasok, manufaktur, retailer, distributor, pergudangan, transportasi dan konsumen) untuk memenuhi kebutuhan pelanggan. Sebuah model rantai pasokan sederhana terdiri dari empat komponen yaitu supplier, produsen, gudang atau pusat distribusi dan pengguna akhir .

2.6 Metode FIFO (*First In First Out*)

FIFO adalah akronim untuk *First In First Out* (pertama masuk, pertama keluar). Sebuah abstraksi yang berhubungan dengan cara mengatur dan memanipulasi data relatif terhadap waktu dan prioritas. Ungkapan ini menggambarkan prinsip teknik pengolahan antrian atau melayani permintaan yang saling bertentangan dengan proses pemesanan berdasarkan perilaku *first-come, first-served* (FCFS). Pengaturan stok barang dengan metode ini berdasarkan stok persediaan yang datang atau masuk pertama kali maka akan dijual atau digunakan terlebih dahulu. Hal ini bertujuan untuk menghindari *expired* dari barang-barang tersebut (Nur, 2018).

2.7 Metode Pengembangan Sistem

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi (Pressman, 2012).



Gambar 2.1 Model *Extreme Programming* (XP)
Sumber : (Pressman, 2012)

Dibawah ini adalah penjelasan tahapan *Extreme Programming* yaitu :

1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsi*ionalitas. Pada perencanaan terdapat *user stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria iteration plan* melakukan perhitungan kecepatan project selama *development*, *customer* dapat menambah *story*, merubah *value*, membagi *story* atau menghapusnya.

2. *Design* (Perancangan)

Perancangan yang simple, menarik, dan sederhana selalu memberikan hasil yang lebih disukai daripada gambaran-gambaran yang lebih kompleks. Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih. Terdapat *simple design* *CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

3. *Coding* (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke

pengkodean. Pada tahapan *pair programming* melakukan kerja sama untuk membuat kode dari satu story. Dan *refactoring* adalah proses restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya.

4. *Testing* (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali. Pada tahapan pengujian yaitu *unit test* *continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software release*. Selanjutnya terdapat tahapan *software increment project velocity computed* yaitu tahapan yang telah diimplementasikan dari *software release* yang nantinya akan diterapkan dalam suatu sistem

2.8 *Unified Modeling Language (UML)*

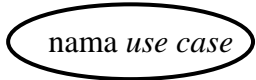


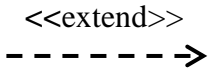
Alat pengembangan sistem yang digunakan pada penelitian ini yaitu *Unified Modeling Language (UML)*. *UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S dan Shalahudin, 2018). Berikut ini merupakan penjelasan

tentang masing-masing diagram yang ada pada *UML (Unified Modelling Language)*.


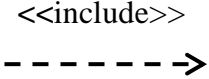
2.8.1 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara salah satu lebih aktor dengan sistem informasi yang akan dibuat (A.S dan Shalahudin, 2019).

Tabel 2.2 Simbol Use Case Diagram

| No. | Simbol | Deskripsi |
|-----|--|---|
| 1 | <p><i>Use case</i></p>  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor. |
| 2 | <p>Aktor / <i>actor</i></p>  | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. |
| 3 | <p>Asosiasi / <i>association</i></p>  | Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| 4 | <p>Ekstensi / <i>extend</i></p>  | Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan |

Tabel 2.2 Simbol Use Case Diagram (Lanjutan)


| | | |
|---|--|---|
| | | dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. |
| 5 | Generalisasi / <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. |
| 6 | <i>Include</i>  | Relasi <i>use case</i> tambahan ke <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini menjalankan fungsionalnya atau sebagai syarat dijalankan <i>use case</i> ini. |

Sumber: (A.S dan Shalahudin, 2019)


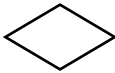


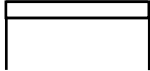
2.8.2 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Hal yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan oleh sistem (A.S dan Shalahudin, 2019).

Tabel 2.3 Simbol Activity Diagram

| No. | Simbol | Deskripsi |
|-----|--|---|
| 1 | Status awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |

Tabel 2.3 Simbol Activity Diagram (Lanjutan)

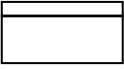



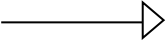
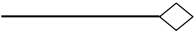
| | | |
|---|--|---|
| 2 | Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3 | Percabangan / <i>decision</i>  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 | Penggabungan / <i>join</i>  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5 | Status Akhir  | Status akhir yang dilakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 | Swimlane  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

Sumber: (A.S dan Shalahudin, 2019)

2.8.3 Class Diagram

Class diagram adalah sebuah *class* yang menggambarkan struktur sistem dari segi pendefinisian kelas kelas yang akan dibuat untuk membangun sistem. *Class diagram* juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan (A.S dan Shalahudin, 2019).

Tabel 2.4 Simbol *Class Diagram*

| No. | Simbol | Deskripsi |
|-----|---|--|
| 1 | Kelas / <i>class</i>  | Kelas pada struktur sistem. |
| 2 | Antarmuka / <i>interface</i>  | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek. |
| 3 | Asosiasi / <i>association</i>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 4 | Asosiasi berarah / <i>directed association</i>  | Relasi antar kelas dengan makna kelas kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 5 | Generalisasi  | Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus). |
| 6 | Agresi / <i>aggregation</i>  | Relasi antar kelas dengan makna semua bagian (<i>whole -part</i>). |

Sumber: (A.S dan Shalahudin, 2019)

2.9 Perangkat Lunak (*Software*)

Software adalah sekumpulan data elektronik yang disimpan dan disesuaikan oleh komputer. Data elektronik yang disimpan oleh komputer dapat

berupa program atau instruksi yang menjalankan perintah. (Handayani, Wijianto dan Anggoro, 2018)

2.9.1 Website

Website adalah kumpulan halaman yang menampilkan informasi data *teks*, data gambar diam atau gerak, data animasi, suara, *video* dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman Destiningrum dan Adrian (2017). Untuk menyediakan sebuah *website*, maka harus menyediakan unsur-unsur penunjangnya, yaitu:

1. Nama *Domain (URL)*

Nama *domain* atau biasa disebut dengan *domain name* atau *URL* adalah alamat unik di dunia *internet* yang digunakan untuk mengidentifikasi sebuah *website*.

2. Rumah Tempat *Website (Web hosting)*

Pengertian *web hosting* dapat diartikan sebagai ruangan yang terdapat dalam *harddisk* tempat menyimpan berbagai data, *file-file*, gambar dan lain sebagainya yang akan ditampilkan di *website*. Besarnya data yang bisa dimasukkan tergantung dari besarnya *web hosting* yang disewa atau dipunyai.

3. Bahasa Program (*Scripts Program*)

Bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* yang pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah *website*.

4. Desain *Website*

Setelah melakukan penyewaan *domain name* dan *web hosting* serta penguasaan bahasa program (*scripts program*), unsur *website* yang penting dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*.

5. Publikasi *Website*

Keberadaan situs tidak ada gunanya dibangun tanpa dikunjungi atau dikenal oleh masyarakat atau pengunjung *internet*. Karena efektif tidaknya situs sangat tergantung dari besarnya pengunjung dan komentar yang masuk. Untuk mengenalkan situs kepada masyarakat memerlukan apa yang disebut publikasi atau promosi *website*.

2.9.2 *MYSQL*

MySQL adalah sebuah *software database*. *MySQL* merupakan tipe data reasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan. Keuntungan menyimpan data di *database* adalah kemudahannya dalam penyimpanan dan menampilkan data karena dalam bentuk tabel (Prayitno dan Safitri, 2015). Kelebihan dari *MySQL* adalah dia menggunakan bahasa *query* standar yang dimiliki *SQL* (*Structure Query Language*). *SQL* adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan untuk semua program pengakses *database* seperti *Oracle*, *Postgres SQL* dan *SQL*.

2.9.3 *XAMPP*

XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dengan menggunakan *XAMPP* kita tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah

disediakan oleh XAMPP. Beberapa paket yang telah disediakan adalah *Apache*, *MySQL*, *Php*, *Filezilla* dan *Phpmysqladmin* (Putera dan Ibrahim, 2018).

2.9.4 PHP

PHP singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP merupakan *Software Open-Source* yang disebar dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya. PHP ditulis menggunakan bahasa C (Hutagalung dan Arif, 2018).

2.9.5 CPanel

Cpanel adalah salah satu *control panel* berbasis *linux* yang paling banyak digunakan di akun *web hosting*. Melalui panel ini, bisa dengan mudah mengelola semua layanan dalam satu tempat. Saat ini, *cpanel* merupakan panel standar yang paling banyak digunakan dan sebagian besar *web developer* sudah sangat familiar dengan tool ini. Selain intuitif dan mudah digunakan, *cpanel* juga memungkinkan untuk mengelola akun *web hosting* dengan maksimal. Membuat FTP *user* dan alamat *email* baru, memonitor *resource*, membuat *subdomain*, dan *install software* merupakan beberapa kegunaan *cpanel* (Lidar, 2020).

2.9.6 Browser

Browser adalah suatu aplikasi atau program yang dijalankan pada perangkat komputer untuk melihat konten yang ada pada media *world wide web* (www) dengan memanfaatkan jaringan internet. Teknologi *browser* yang

berkembang saat ini tidak hanya dapat menampilkan halaman yang berisi *text* atau tulisan saja, *browser-browser* populer sekarang dapat menampilkan gambar, musik, suara, video, file pdf dan data lainnya (Agus, Marisa dan Wijaya, 2017).

2.10 Pengujian ISO 25010

Standar *ISO/IEC 25010* pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. Dokumen standar *ISO/IEC 25010* sangat panjang. Hal ini dikarenakan orang memiliki motivasi berbeda yang memungkinkan untuk tertarik pada kualitas perangkat lunak (Rahayuda, 2017). Secara keseluruhan, *ISO/IEC 25010* memiliki 6 karakteristik untuk mengukur kualitas perangkat lunak secara menyeluruh, antara lain *functional suitability*, *reliability*, *performance efficiency*, *usability*, *maintainability*, dan *portability* (Estdale dan Georgiadou, 2018).

1. *Functional Suitability*

Pengujian *functional suitability* merupakan tingkat dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi yang spesifik.

2. *Reliability*

Pengujian *reliability* ini merupakan kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.

3. *Performance efficiency*

Pengujian *performance* ini dilakukan untuk mengukur karakteristik performa dari komponen aplikasi. Aspek ini merupakan aspek untuk

mengukur keandalan sistem informasi yang digunakan pengembangan aplikasi sistem informasi. yang sama.

4. *Usability*

Pengujian *usability* ini merupakan kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.

5. *Maintainability*

Pengujian *maintainability* dilakukan untuk menguji efektifitas dan efisiensi perangkat lunak untuk dimodifikasi atau dikembangkan.

6. *Portability*

Pengujian *portability* ini merupakan kemampuan perangkat lunak untuk di transfer dari satu lingkungan ke lingkungan lain.



Gambar 2.2 Model Pengujian Perangkat Lunak ISO 25010
Sumber : (Estdale dan Georgiadou, 2018).

Masing-masing karakteristik model *ISO 25010* dibagi menjadi beberapa sub-karakteristik kualitas. Tabel karakteristik model *ISO 25010* dapat dilihat pada tabel 2.5:

Tabel 2.5 Karakteristik *ISO 25010*

| Karakteristik | Indikator | Deskripsi |
|-------------------------------|-----------------------------------|--|
| <i>Functional Suitability</i> | <i>Functional Completeness</i> | Kemampuan perangkat lunak dalam menyediakan fungsi dapat mencakup semua tugas dan tujuan pengguna secara spesifik. |
| | <i>Functional Correctness</i> | Kemampuan perangkat lunak sejauh mana produk atau sistem mampu menyediakan hasil yang tepat sesuai dengan kebutuhan. |
| | <i>Functional Appropriateness</i> | Kemampuan perangkat lunak dalam fungsi yang disediakan mampu menyelesaikan tugas dan tujuan secara spesifik. |
| <i>Performance Efficiency</i> | <i>Resource Utilization</i> | Kemampuan perangkat lunak sejauh mana jumlah dan jenis sumber daya yang digunakan produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria |
| | <i>Capacity</i> | Kemampuan perangkat lunak sejauh mana batas maksimal sebuah produk atau sistem mampu memenuhi kriteria. |

Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)

| | | |
|------------------|--|--|
| | <i>Time Behaviour</i> | Kemampuan perangkat lunak sejauh mana respon dan lama proses sebuah produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria. |
| <i>Usability</i> | <i>Appropriateness</i> <i>Recognizability</i> | Kemampuan perangkat lunak sejauh mana pengguna mengetahui sebuah produk atau sistem sesuai dengan kebutuhan mereka. |
| | <i>Learnability</i> | Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan pengguna tertentu untuk mencapai tujuan mempelajari sebuah produk atau sistem secara efektif, efisien dan bebas dari resiko serta memenuhi kepuasan dalam konteks penggunaan. |
| | <i>Operability</i> | Kemampuan perangkat lunak sejauh mana produk atau sistem sistem mampu memenuhi kriteria dioperasikan dan dikontrol. |
| | <i>User Error</i> <i>Protection</i> | Kemampuan perangkat lunak sejauh mana produk atau sistem |

Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)

| | | |
|--------------------|--|--|
| | | melindungi pengguna dalam melakukan kesalahan. |
| | <i>User Interface</i> <i>Aesthetics</i> | Kemampuan perangkat lunak sejauh mana tampilan antarmuka memenuhi kesenangan dan kepuasan pengguna. |
| | <i>Accessibility</i> | Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan oleh pengguna secara luas untuk mencapai tujuan tertentu dalam konteks penggunaan |
| <i>Reliability</i> | <i>Maturity</i> | Kemampuan perangkat lunaksejauh mana sistem, produk, atau komponen memenuhi kriteria reliabilitas dibawah kondisi normal. |
| | <i>Availability</i> | Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen dapat beroperasi ketika diperlukan untuk digunakan. |
| | <i>Fault Tolerance</i> | Kemampuan perangkat lunak sejauh mana sistem, produk atau komponen. |
| | <i>Recoverability</i> | Kemampuan perangkat lunak ketika produk atau sistem |

Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)

| | | |
|------------------------|----------------------|--|
| | | mengalami kejadian atau kegagalan dapat mengembalikan data dan menjalankan kondisi sistem yang diharapkan. |
| <i>Maintainability</i> | <i>Modularity</i> | Kemampuan perangkat lunak dimana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain |
| | <i>Reusability</i> | Kemampuan perangkat lunak dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan asset lainnya. |
| | <i>Analyzability</i> | Kemampuan perangkat lunak dimana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi. |
| | <i>Modifiability</i> | Kemampuan perangkat lunak dimana perangkat lunak dapat |

Tabel 2.5 Karakteristik *ISO 25010* (Lanjutan)

| | | |
|--------------------|-----------------------|--|
| | | menghindari efek yang tidak diharapkan dari modifikasi yang dilakukan terhadap perangkat lunak. |
| | <i>Testability</i> | Kemampuan perangkat lunak dimana perangkat lunak memungkinkan modifikasi perangkat lunak untuk dilakukan validasi. |
| <i>Portability</i> | <i>Adaptability</i> | Kemampuan perangkat lunak dapat beradaptasi dengan perubahan lingkungan atau sistem yang berbeda. |
| | <i>Installability</i> | Kemampuan perangkat lunak dapat digunakan dalam lingkungan atau sistem tertentu. |
| | <i>Replaceability</i> | Kemampuan perangkat lunak dapat menggantikan perangkat lunak lain apakah ada kebergantungan kepada perangkat lunak lain digunakan. |

Sumber : (Estdale dan Georgiadou, 2018)