

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini menggunakan tinjauan Pustaka dari penelitian sebelumnya yang nantinya akan digunakan sebagai pendukung penelitian, berikut ini merupakan studi yang diambil dapat dilihat pada Tabel 2.1.

Tabel 2 1 Tinjauan Pustaka

| NO. | Nama | Judul | Masalah | Kesimpulan |
|-----|-----------------------|--|--|--|
| 1. | (Indyah et al., 2022) | Rancang Bangun Sistem Informasi Pengolahan Data Transaksi Jual Beli Di Toko Andis's Collection | Proses penjualan yang dilakukan pada Toko Andies's Collection sampai saat ini hanya melalui penjualan langsung dengan datang ke toko untuk melihat barangnya ataupun melalui telepon dan sosial media. Selain itu toko ini masih menggunakan sistem pembukuan secara manual dengan pencatatan aktivitas keluar masuk barang pada buku. Hal ini dapat mengakibatkan kekeliruan dalam pendataan barang dan akan berdampak juga pada transaksi kepada konsumen. Dengan ini dapat menghambat | Penataan bertujuan supaya memudahkan dalam melaksanakan penilaian apakah informasi yang sudah dikumpulkan telah cocok ataupun tidak. Data-data yang diperoleh dari hasil wawancara serta observasi tentang Toko Andis's Collection butuh dicoba penataan yaitu dengan metode menyaring serta pilih terhadap informasi yang dikira lebih berarti didalam menolong menanggulangi permasalahan yang diteliti, setelah informasi disaring serta diklasifikasikan akan diolah dengan memakai analisis |

Tabel 2 1 Tinjauan Pustaka (lanjutan)

| | | | | |
|----|----------------------|---|--|---|
| | | | aktivitas toko karena tidak terkontrolnya persediaan barang dengan baik. | informasi kualitatif yang bertujuan untuk penyederhanaan informasi. |
| 2. | (Sidik et al., 2021) | Sistem Informasi Pengolahan Data Penjualan Berbasis Web | Sistem pengolahan data penjualan pada bagian admin sales ini masih menggunakan sistem manual dimana proses menginput data pesanan yang masuk dari berbagai <i>marketplace</i> seperti : (lazada, shopee, tokopedia, dan bukalapak) juga media sosial seperti : (whatsapp, line, instagram, dan fanpage) masih menggunakan <i>microsoft word</i> dan <i>excel</i> dimana transaksi perharinya sangat banyak sehingga rentan akan <i>human eror</i> . Sama halnya pada bagian admin gudang dimana masih melakukan pengecekan dan pencatatan secara manual. | Setelah dilakukannya analisis dan perancangan sistem informasi pengolahan data penjualan pada PT Global Indojaya Orion. Sistem yang dibuat dapat mempermudah bagian admin untuk melakukan proses pengolahan data penjualan dan data gudang dalam melakukan proses pengiriman barang secara efektif. Sistem informasi pengolahan data penjualan dapat diatasi dengan membuat sistem yang terintegrasi dengan database menggunakan MySQL dan bahasa pemrograman yaitu PHP, serta membuat sistem berbasis website yang dapat mempermudah dalam proses input dan pencairan data. Penggunaan sistem tersebut dapat mempermudah |

Tabel 2 1 Tinjauan Pustaka (lanjutan)

| | | | | |
|----|---------------------------|---|--|--|
| | | | | proses permintaan barang pada PT Global Indojaya Orion dan dapat menghasilkan laporan-laporan pendataan barang apa saja yang sudah di transaksikan. |
| 3. | (Khairina & Rahman, 2022) | Sistem Informasi Pengolahan Data Pajak Keluaran Pada Pt. Idaman | Kendala yang dirasakan dan sering terjadi atas alur yang berjalan saat ini yaitu banyaknya data penjualan yang harus diolah sehingga hasil yang didapatkan kurang maksimal dan cukup menyita waktu kerja. Selain itu pula terkadang terjadi kesalahan <i>inpor</i> data ke dalam sistem atau aplikasi e-faktur dikarenakan terdapat transaksi yang isian bagan formnya kurang atau tidak terisi. | Kemudahan menambahkan data pelanggan, penjualan dan pengolahan data pajak keluaran, serta bisa mengubah dan menyesuaikan data tersebut sesuai kebutuhan yang ada, menjadikan sistem informasi pengolahan data ini lebih mudah digunakan oleh karyawan untuk mengolah data pajak keluaran yang ada. Berdasarkan hasil pengujian semua fungsi pada sistem informasi pengolahan data pajak keluaran ini mendapatkan hasil sesuai dengan apa yang diharapkan, dimana dari proses inpor data pelanggan dan penjualan sebagai proses inputnya yang akan terolah dan diproses |

Tabel 2 1 Tinjauan Pustaka (lanjutan)

| | | | | |
|----|-----------------|---|--|---|
| | | | | <p>lagi menjadi menjadi sebuah data pajak keluaran dan menjadi sebuah output berupa file CSV yaitu sebuah data pajak keluaran, yang akan di inpor ke aplikasi e-faktur yang berjalan sesuai dengan apa yang diharapkan.</p> |
| 4. | (Prayoga, 2019) | <p>Perancangan Sistem Informasi Pengolahan Data Penjualan Secara Kredit dan Controlling Stok Dengan Menerapkan Metode Backorder Pada Toko Master Menggunakan Bahasa Pemrograman Java dan Database MySQL</p> | <p>Pengolahan data penjualan kredit dan persediaan (Stok) barang pada Toko Master masih dilaksanakan secara manual sehingga kemungkinan terjadinya kesalahan dalam memproses data penjualan dan persediaan barang sangat besar serta menyulitkan untuk mengetahui penjualan kredit dan persediaan barang pada Toko Master tersebut. Informasi mengenai penjualan kredit dan stok barang ini sangat penting karena proses pembelian maupun penjualan berawal dari sistem ini, untuk itulah sangat diperlukan informasi yang akurat mengenai</p> | <p>Berdasarkan perancangan sistem informasi penjualan dan pengolahan stok barang serta analisa-analisa yang dilakukan pada Toko Master, maka penulis mengambil beberapa kesimpulan : Dengan desain sistem baru secara komputerisasi dan mengaplikasikan ke dalam Bahasa Pemrograman Java dalam perancangan sistem informasi penjualan kredit dan pengendalian stok barang dengan menerapkan metode <i>Backorder</i> dapat membantu perusahaan dalam menyajikan informasi yang akurat dan efisien. Dengan desain sistem baru</p> |

Tabel 2 1 Tinjauan Pustaka (lanjutan)

| | | | | |
|----|---------------------------|---|--|---|
| | | | <p>sistem pengolahan data ini. Selain itu Toko Master sering mengalami suatu kondisi dalam pendistribusian barang, dimana barang yang dipesan tidak tahu atau belum dapat disediakan baik seluruhnya ataupun sebagian. Hal ini mengakibatkan terganggunya transaksi penjualan sehingga menurunkan omset penjualan yang berimbas kepada kerugian.</p> | <p>diharapkan informasi tersebut dapat disimpan dengan menyediakan file-file dalam database, maka informasi tersebut sewaktu-waktu dapat di update dengan tepat. Dengan menggunakan sistem informasi penjualan kredit dan pengendalian stok barang dapat mengetahui stok akhir barang untuk melakukan order barang. Sistem stok barang yang baru menggunakan bahasa pemrograman Java, dimana pemakai sistem tidak perlu memikirkan biaya lisensi dari aplikasi tersebut karena bahasa pemrograman tersebut gratis dalam pemakaiannya.</p> |
| 5. | (Rifai & Mailasari, 2020) | Metode Waterfall pada Sistem Informasi Pengolahan Data Penjualan dan Pembelian Barang | <p>Seperti pada Toko Fang Fifi Cell yang bergerak dalam bidang penjualan handphone dan aksesoris ini pendataan masih sangat manual, mulai dari pendataan barang, transaksi,</p> | <p>Kesimpulan dalam pembuatan penelitian ini sebagai berikut : Dapat mempermudah pekerjaan dalam pengolahan data. Dapat mengurangi kesalahan-kesalahan pada sistem yang</p> |

Tabel 2 1 Tinjauan Pustaka (lanjutan)

| | | | | |
|--|--|--|---|--|
| | | | hingga pembuatan laporan. Pendataan barang dan transaksi masih menggunakan tulis tangan sehingga sering terjadi kesalahan pencatatan dan pencairan data yang sulit karena adanya penumpukan berkas. | masih menggunakan cara konvensional. Dapat disimpan dengan jumlah banyak dan aman karena telah menggunakan password. Dapat mengurangi terjadinya duplikasi data. |
|--|--|--|---|--|

Kesimpulan yang dapat diambil dari tinjauan literatur di atas yaitu masalah yang muncul karena pengolahan data penjualan yang masih menggunakan cara konvensional dapat teratasi dengan sistem yang terkomputerisasi dengan memanfaatkan aplikasi berbasis *web* sehingga proses transaksi yang dilakukan akan lebih efisien.

2.2 Keaslian Penelitian

Berdasarkan studi literature dari 5 peneliti terdahulu, maka keaslian penelitian yang dilakukan terdapat perbedaan yaitu:

1. Metode pengembangan sistem yang digunakan adalah *Extreme Programming*
2. Pengujian sistem dilakukan menggunakan standar ISO 25010
3. Aplikasi yang digunakan penulis berupa desktop

2.3 Rancang Bangun

Perancangan merupakan salah satu hal yang penting dalam membuat program. Adapun tujuan dari perancangan adalah untuk memberi gambaran yang jelas serta lengkap kepada pemrogram dan ahli teknik yang terlibat. Perancangan

harus berguna dan mudah dipahami sehingga dapat dengan mudah digunakan. Perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem tersebut dapat di implementasikan (Sherley, dkk., 2021)

2.4 Sistem Informasi

Sistem merupakan rangkaian dari kumpulan data yang disusun secara teratur dan saling berkaitan satu dengan yang lain untuk mencapai tujuan dan memudahkan penyampaian informasi dalam melaksanakan kegiatan oprasional organisasi atau perusahaan.

Informasi adalah hasil dari kumpulan data yang telah diproses dan diolah sedemikian rupa sehingga memiliki nilai guna bagi penggunanya, serta dapat membantu dalam proses pengambilan keputusan. Jadi dapat disimpulkan bahwa sistem informasi merupakan suatu proses pengolahan data dari berbagai kelompok kegiatan yang di proses sedemikian rupa dengan teknik tertentu untuk menghasilkan informasi yang berguna bagi penggunanya, yang selanjutnya digunakan dalam membantu pengambilan keputusan serta dapat memudahkan kegiatan kerja sehari-hari. (Achmad, dkk, 2021)

2.5 Pengolahan Data

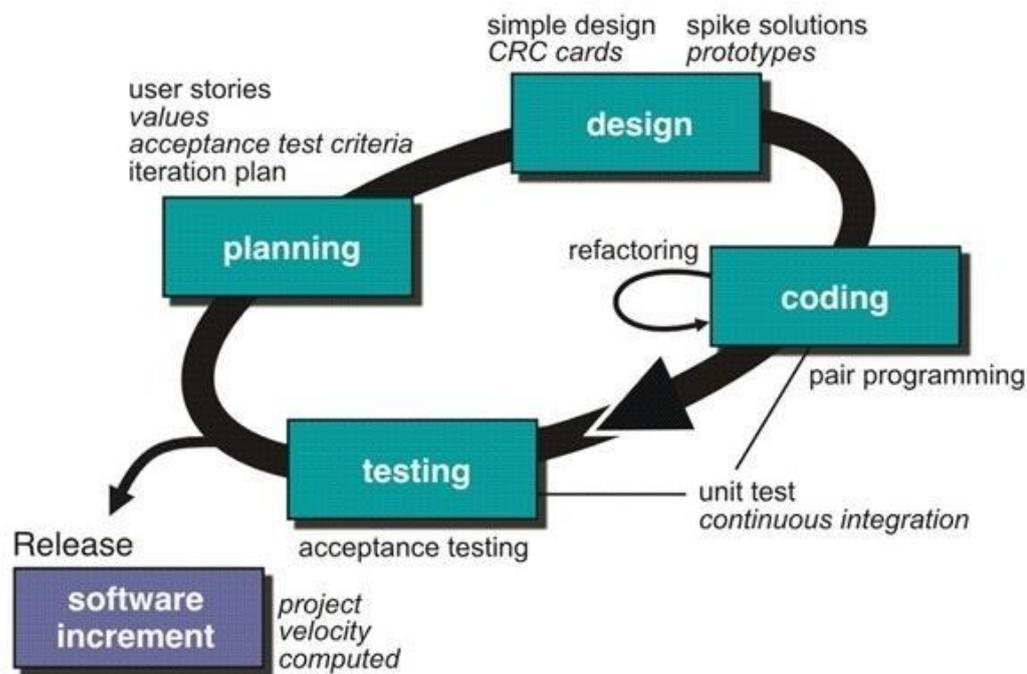
Pengolahan data adalah sebuah proses manipulasi data untuk menjadi sebuah informasi. Kumpulan data yang awal tidak memiliki informasi dapat disimpulkan jika dilakukan proses pengolahan data maka akan menghasilkan informasi. Hasil dari pemrosesan data tertentu yang bermakna serta dapat digunakan untuk mengambil keputusan. Pengolahan data terdiri dari beberapa

kegiatan yaitu pencairan data, pengumpulan data, pemilihan data, dan penggunaan data.

2.6 *Extreme Programming (XP)*

Extreme Programming (XP) merupakan metodologi yang digunakan untuk pengembangan perangkat lunak yang ditujukan dalam meningkatkan kualitas perangkat lunak terhadap perubahan serta kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pro pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Ada beberapa tahapan yang ada pada *Extreme Programming* yaitu terdiri dari Perencanaan (*Planning*) seperti memahami kriteria pengguna dan perencanaan pengembangan, designing seperti perancangan *prototype* dan tampilan, pengkodean juga termasuk dalam pengintegrasian, terakhir adalah testing (Ariyanti,dkk, 2020).

Menurut (Pressman, 2015), *extreme programming* adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas 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. Tahapan-tahapan dari *extreme programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing*



Gambar 2 1 Tahapan Extreme Programming

Sumber : (Pressman, 2012)

seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian dan yang terakhir adalah *testing* yaitu percobaan.

Unsur-unsur dari *extreme programming* meliputi *paired programming* pada tahapan *coding*, *unit testing* pada semua kode, penghindaran pemrograman fitur kecuali benar-benar diperlukan, struktur manajemen yang datar, kode yang sederhana dan jelas, dan seringnya terjadi komunikasi antara *programmer* dan

pelanggan ketika terjadi perubahan kebutuhan pelanggan sering berlalunya waktu berlalu. Metode ini membawa unsur-unsur yang menguntungkan dari praktek rekayasa perangkat lunak tradisional ke tingkat “ekstrem”, sehingga metode ini dinamai *extreme programming*. unsur-unsur yang menjadi karakteristik metodologi adalah kesederhanaan, komunikasi, umpan balik dan keberanian.

2.7 Metode Perancangan Sistem

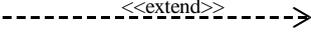
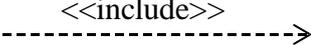
2.7.1 Unified Modelling Language (UML)

UML (*Unified Modelling Language*) adalah sebuah bahasa berbentuk grafik atau gambar yang digunakan untuk memvisualisasikan sebuah sistem yang akan dikembangkan berbasis *Object-Oriented*. UML juga memiliki standar penulisan sebuah sistem *blue print*, meliputi kelas dalam bahasa pemrograman yang spesifik, proses bisnis, skema *database* dan komponen lainnya yang diperlukan sistem (Mubarak dkk, 2019). Terdapat tiga diagram UML yang memiliki fungsi masing – masing yaitu:

a. *Use Case Diagram*

Use Case Diagram 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. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Dibawah ini adalah tabel yang berisi simbol-simbol dan penjelasan dari setiap simbol yang ada pada *use case diagram*, (A.S & Shalahuddin, 2016).

Tabel 2.2 Simbol *Use Case Diagram*

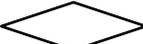
| Simbol | Deskripsi |
|--|--|
| <i>Use Case</i>  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertuar pesan antar unit atau aktor. |
| Aktor / <i>Actor</i>  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| Asosiasi / <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. |
| Ekstensi / <i>extend</i>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. |
| 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. |
| Menggunakan / <i>include</i> / <i>uses</i>   | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> lain untuk menjalankan fungsinya atau sebagai syarat. |

Sumber : (A.S & Shalahuddin, 2016)

b. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem proses atau proses bisnis atau menu yang ada pada perangkat lunak. Aktivitas yang dimaksud bukanlah aktivitas yang dilakukan oleh aktor tapi aktivitas yang dilakukan oleh sistem, (A.S & Shalahuddin, 2016). Berikut ini adalah simbol-simbol dari *activity diagram* :

Tabel 2.3 Simbol *Activity Diagram*

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

Tabel 2.3 Simbol *Activity Diagram* (lanjutan)

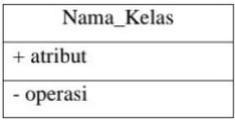
| | |
|---|--|
| Penggabungan / <i>join</i>  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status akhir  | Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki status akhir. |
| <i>Swimlane</i>  | Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi |

Sumber : (A.S & Shalahuddin, 2016)

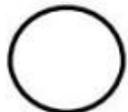
c. *Class Diagram*

Class diagram adalah sebuah diagram UML yang menggambarkan hubungan antar kelas dan memiliki penjelasan detail dari sebuah sistem di dalam model desain, dan juga memperhatikan entitas perilaku sistem. Komponen-komponen yang terdapat pada *class diagram* yaitu *class*, relasi, asosiasi, generalisasi dan agregasi, atribut, operasi, dan visibilitas, tingkat akses objek eksternal pada suatu operasi. Juga hubungan antar kelas disebut *multiplicity* dan *cardinality*.

Tabel 2.4 Simbol *Class Diagram*

| Simbol | Keterangan |
|---|---|
|  | Menggambarkan kelas yang terdapat pada struktur |

Tabel 2.4 Simbol *Class Diagram* (lanjutan)

| | |
|---|--|
|  | Asosiasi berarah: relasi antar kelas yang memiliki arti kelas satu digunakan oleh kelas yang lain dan biasanya disertai dengan <i>multiplicity</i> . |
|  | Generalisasi: relasi antar kelas yang memiliki makna umum khusus |
|  | Ketergantungan: <i>depedency</i> merupakan relasi antar kelas dengan arti ketergantungan antar kelas. |
|  | Agregasi: merupakan relasi antar kelas dengan arti semua bagian |
|  | Antar muka: yaitu menunjukkan interface hal ini sama dengan konsep interface pada pemrograman berorientasi objek. |
|  | Asosiasi: merupakan relasi antar kelas dengan makna umum dan biasanya disertai <i>multiplicity</i> . |

Sumber : (A.S & Shalahuddin, 2016)

2.7.2 Rancangan Antar Muka

User interface adalah cara program dan pengguna untuk berinteraksi. Istilah *user interface* terkadang digunakan sebagai istilah pengganti *Human Computer Interaction (HCI)* dimana semua aspek dari interaksi pengguna dan komputer. Semua yang terlihat di layar, membaca dalam dokumentasi dan dimanipulasi dengan *keyboard* (atau *mouse*) juga merupakan bagian dari *user interface*, (Lastiansah, 2012)

2.8 Metode Pengujian ISO 25010

Standar ISO/IEC 25010 pertama kali diperkenalkan melalui pertanyaan tentang definisi kualitas perangkat lunak dokumen standar ISO/IEC 25010 sangat panjang. Hal ini dikarenakan orang memiliki motivasi yang berbeda yang memungkinkan untuk tertarik pada kualitas perangkat lunak,

Model ini merupakan bagian dari *Software Product Quality Requirements and Evaluation (SQUARE)*, dimana model ini berkaitan dengan model kualitas perangkat lunak yang merupakan pengembangan dari model sebelumnya. Pada model ini terdapat beberapa sub-karakteristik tambahan dan beberapa sub-karakteristik yang dipindahkan ke karakteristik lain. Karakteristik atau faktor kualitas internal dan eksternal yang terdapat pada model ISO-25010 meliputi delapan karakteristik sebagai berikut:

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. *Performance Efficiency*

Pengujian *performance* ini dilakukan untuk mengukur karakteristik performadari komponen aplikasi. Aspek ini merupakan aspek untuk mengukur kendala sistem informasi yang digunakan pengembangan aplikasi sistem informasi.

3. *Compatibility*

Pengujian *compatibility* ini merupakan testing yang dilakukan dengan menjalankan produk, sistem atau komponen dengan perangkat keras dan/atau perangkat lunak dalam suatu sumber daya 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. *Reliability*

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

6. *Security*

Pengujian *security* dilakukan untuk menguji keamanan aplikasi *web* terhadap serangan *attack* kepada sistem.

7. *Maintainability*

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

8. *Portability*

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

2.8.1 Karakteristik Kualitas Perangkat Lunak Model ISO 25010

Masing-masing karakteristik kualitas perangkat lunak model ISO 25010 dibagi menjadi beberapa sub-karakteristik kualitas. Tabel Karakteristik Kualitas Perangkat Lunak Model ISO 25010 dapat dilihat pada tabel 2.5 berikut:

Tabel 2 4 Karakteristik ISO 25010

| Karakteristik | Sub Karakteristik | 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>Time Behavior</i> | Kemampuan perangkat lunak sejauh mana respond an lama proses sebuah produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria |
| | <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 |
| <i>Compatibility</i> | <i>Co-existence</i> | Kemampuan perangkat lunak sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien saat berbagi sumber daya dengan produk atau sistem lain tanpa memberi dampak terhadap produk atau sistem tersebut |
| | <i>Interoperability</i> | Kemampuan perangkat lunak sejauh mana dua atau komponen dapat bertukar informasi dan menggunakan informasi tersebut |
| <i>Usability</i> | <i>Appropriateness 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 |

Sumber: (Maliki & Wiharja, 2017)

Adapun alasan penggunaan ISO 25010 karena ISO sudah berstandar *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. Kualitas produk perangkat lunak ISO 25010 memiliki delapan karakteristik pendukung yang dapat digunakan sebagai acuan dalam menilai maupun memberikan masukan terhadap kualitas perangkat lunak yang akan dibangun yang akan menghasilkan nilai uji yang terukur.

2.9 Instrumen Pengujian

Menurut (Sugiyono, 2018), skala likert adalah skala yang dapat digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Pilihan terhadap masing-masing jawaban untuk tanggapan responden atas dimensi kualitas pelayanan (X) dan kepuasan tamu (Y) diberi skor sebagai berikut.

Tabel 2 5 Skala Likert

| Jawaban | Skor |
|---------------------------|------|
| Sangat Setuju (SS) | 5 |
| Setuju (S) | 4 |
| Kurang Setuju (KS) | 3 |
| Tidak Setuju (TS) | 2 |
| Sangat Tidak Setuju (STS) | 1 |

2.10 Point of Sale

Penafsiran dari *Point of Sale (POS)* yaitu ialah aktivitas yang berorientasi pada penjualan dan sistem yang menolong proses transaksi. Tiap POS terdiri dari hardware berbentuk (terminal/ Komputer, *Receipt Printer*, *Cash Drawer*, terminal pembayaran, *Barcode Scanner*) serta aplikasi berbentuk (*Inventory Management*,

Pelaporan, *Purchasing*, *Customer Management*, Standar Keamanan Transaksi, *Return Processing*) dimana kedua komponen tersebut digunakan setiap proses transaksi. *Point of Sales* (POS) bisa berbentuk suatu *checkout counters* dalam suatu toko ataupun tempat usaha dimana transaksi penjualan terjalin. POS hendak jadi sangat berarti di dunia bisnis sebab POS diibaratkan berbentuk terminal uang dimana tempat menerima pembayaran dari pembeli kepada pedagang,(F & T.G, 2017).

2.11 Figma

Figma merupakan aplikasi yang digunakan buat pembuatan desain berbasis *cloud* dengan konsep fungsionalitas *sketch*, serta bisa melaksanakan kerja sama antar desainer yang lain. figma bisa bekerja pada bermacam sistem pembedahan sebab bisa berjalan pada *browser*, *fitur slack* yang ada pada figma selaku media komunikasi dari satu *desainer* ke *desainer* yang lain bisa mempermudah sesuatu regu dalam berbicara secara *realtime* dalam melaksanakan *editing*, keunggulan lain ialah pengguna bisa dengan gampang memperoleh kode dalam wujud *css* sehingga mempermudah *front end* dalam pengimplementasian desain yang terdapat (Nugraha dkk., 2020).