

## BAB II LANDASAN TEORI

### 2.1. Penelitian Terdahulu

Penelitian terdahulu yang digunakan dalam penelitian ini adalah :

**Tabel 2. 1 Tabel Penelitian Terdahulu**

Judul	Penulis	Tahun
Sistem Pendukung Keputusan Menentukan Kualitas Merek Tepung Terigu Terbaik Menggunakan Metode <i>Profile Matching</i>	Citra Tri Ulfada, Irvan, Fachrul Rozi Lubis	2022
Sistem Pendukung Keputusan Penentuan Kelayakan Usaha Mikro Kecil Menengah Dengan Menggunakan Metode Profile Matching Pada UPTD PLUT KUMKM Provinsi Lampung	Risa Dwi Kurniawati, Imam Ahmad	2021
Analisa Metode <i>Profile Matching</i> Dalam Menentukan Kelayakan Pemberian Kredit Kepemilikan Rumah (KPR)	Teuku Radillah, Leonard Tambunan, Budy Satria, Muhammad Iqbal	2021
Profile Matching Untuk Pemilihan Produk Asuransi Terbaik	Sumanto	2020
Analisis Metode <i>Profile Matching</i> Pada Rekomendasi Cat Dinding Rumah Berdasarkan Konsumen	Sekar Rizky Rani, Ratika Rizka Lubis, Agus Perdana Windarto	2018

1. (Ulfada & Lubis, 2022). Metode *profile matching* sebagai solusi dalam membantu menentukan merek tepung terigu terbaik. Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga diketahui perbedaan kompetensinya (disebut juga gap). Semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar yang berarti memiliki peluang lebih besar untuk menempati posisi tersebut. Karena banyaknya jenis merek tepung terigu yang beredar dipasaran menyulitkan pembeli dalam mengetahui

kualitas dari berbagai merek tepung terigu. Untuk itu pembeli dapat memilih jenis, sifat dan mutu merek tepung terigu sesuai dengan yang di kehendaki dan juga sesuai dengan daya beli. Dengan mengenali ciri fisik Merek Tepung Terigu berkualitas seperti bentuk tekstur, warna, dan juga harga. Merek tepung terigu yang baik biasanya memiliki kadar protein yang tinggi. Selain itu juga bisa dinilai dari segi warna, apakah tepung terigu bersih atau tidak, dan dari berbagai aspek fisik lainnya yang akan menjadi kriteria dalam penilaian. Dengan adanya metode *profile matching* ini, merek tepung terigu yang akan menjadi kandidat akan dihitung berdasarkan kriteria penilaian fisik dari setiap merek yang mendapat nilai tertinggi sebagai yang terbaik.

2. (Kurniawati & Ahmad, 2021) Penelitian ini dilakukan untuk membuat sebuah sistem yang dapat menentukan kelayakan para UKM secara cepat, tepat dan sistematis. Dikarenakan jumlah UKM yang sangat banyak maka PLUT KUMKM tidak mungkin bisa menentukan para UKM layak atau tidaknya menjadi UKM dan UKM tidak mengetahui usaha mereka masuk kedalam kategori mana, seperti usaha mikro, kecil dan menengah. Sistem pendukung keputusan ini menggunakan metode *Profile Matching*. Tujuan pembuatan sistem ini adalah mengukur kelayakan UKM berdasarkan aspek finansial dan non finansial serta melakukan pengelompokkan kategori UKM. Diharapkan dapat melakukan pengambilan keputusan yang diharapkan dapat mempermudah dalam proses penentuan kelayakan UKM. Memudahkan PLUT dalam menentukan kelayakan UKM. Sistem ini dibangun dengan Bahasa pemrograman PHP dan database *MYSQL*. Pengujian terhadap sistem ini menggunakan *Black Box Testing* yang menguji aspek *functionality*. Hasil

pengujian *functionality* menghasilkan nilai sebesar 100% dengan kesimpulan diterima secara kelayakan fungsi oleh pengguna.

3. (Radillah, 2021). KPR (Kredit Kepemilikan Rumah) merupakan pembelian rumah secara kredit atau cicilan dengan jangka waktu dan suku bunga tertentu. Permasalahan yang Seringkali terjadi pada proses pemberian KPR adalah dalam memberikan analisa rekomendasi kelayakan rumah seperti kesalahan dalam menganalisa kredit sering menjadi menyebabkan pembayaran kredit macet, dan mengganggu sistem perputaran dana antara pencairan pemberian kredit dengan siklus piutang yang ditagihkan. Selain itu belum adaya filter dalam penentuan kelayakan pemberian kredit secara komputerisasi yang dilengkapi dengan sistem penunjang keputusan yang dapat mendeskripsikan faktor-faktor pendukung yang menjadi bobot perhitungan dalam pengambilan keputusan untuk menentukan kelayakan pemberian kredit rumah. Untuk menganalisa kelayakan pemberian KPR tersebut dibutuhkan suatu sistem pendukung keputusan untuk memberikan analisa yang baik dengan menggunakan metode profile matching sebagai acuan dalam memberikan keputusan kelayakan kredit rumah, dan dari implementasi metode profile matching ini menghasilkan analisa perhitungan yang akurat yaitu standar batas kelayakan ni (Nilai total aspek) harus diatas 2,7. Batas nilai tersebut merupakan parameter yang diperoleh dari akumulasi (penjumlahan) total nilai aspek (ni) terendah yang didapat dari bobot nilai kapasitas GAP, sehingga calon konsumen yang memperoleh diawah nilai batas tersebut akan dikategorikan tidak layak.

4. (Sumanto, 2020). Di era sekarang ini semakin banyak perusahaan asuransi bermunculan dengan menawarkan berbagai macam program yang tentunya menjadi daya tarik bagi calon nasabah yang ingin menggunakan jasa asuransi tersebut. Diantara program-program yang di tawarkan oleh pihak perusahaan asuransi adalah asuransi kesehatan, asuransi jiwa dan asuransi pendidikan. Menurut Kebanyakan orang akhirnya menyesal setelah masuk asuransi dikarenakan mereka kurang memahami akan kebutuhan mereka di dalam berasuransi, sehingga yang terjadi mereka menganggap asuransi itu tidak bermanfaat dan tidak begitu dibutuhkan karna sudah ditanggung oleh perusahaan ditempat mereka bekerja. Dalam penelitian kali ini dengan menggunakan metode profile matching menjadi pilihan untuk menentukan pemilihan produk asuransi. Memilih produk asuransi sering terjadi masalah-masalah yang dihadapi oleh calon nasabah, yaitu kurangnya pemahaman nasabah terhadap produk-produk yang ditawarkan. Untuk itu diperlukan suatu metode profile matching yang dapat mempermudah nasabah dalam mendapatkan informasi serta dapat membantu calon nasabah memahami asuransi agar tidak salah keliru dalam memilih produk-produk yang ditawarkan. Profile Matching merupakan proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap), Semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar. digunakan untuk menghitung kriteria produk asuransi mana yang sesuai untuk nasabah dan dapat membantu calon nasabah melakukan keputusan yang tepat untuk memilih produk asuransi. Setelah dilakukan pembobotan nilai melalui perhitungan

dengan menggunakan metode Profile Matching dimana asuransi jiwa mendapatkan ranking pertama dengan nilai total 3,47. asuransi pendidikan mendapatkan ranking kedua dengan nilai total 3,43. Dan asuransi kesehatan mendapatkan ranking ketiga dengan nilai total 3,39.

5. (Rani et al., 2018). Rumah adalah salah satu tempat tinggal dimana cat dinding merupakan salah satu bagian yang mempengaruhi suatu bangunan yang memberikan kesan keindahan pada bangunan itu sendiri. Dengan banyaknya jenis, variasi dan merk cat dinding yang beredar, banyak konsumen kesulitan dalam menemukan merk cat dinding yang sesuai, terlebih bagi konsumen yang awam dikarenakan kurangnya pengetahuan tentang spesifikasi dan jenis cat yang baik untuk dipakai sebagai pelapis dinding rumah. Berdasarkan permasalahan tersebut, maka peneliti mengangkat topik rekomendasi cat dinding rumah dengan metode Profile Matching. Berdasarkan perhitungan yang dilakukan menggunakan metode Profile Matching. Diperoleh hasil dimana alternatif 4 dengan nilai 1,4 menjadi alternatif pertama yang direkomendasikan dan disusul dengan alternatif pertama dengan nilai 1,1. Adapun tujuan dari penelitian ini yaitu untuk dapat merekomendasikan merk cat dinding yang sesuai dengan menggunakan metode Profile Matching. Diharapkan penelitian ini dapat memberikan masukan kepada konsumen dalam memilih cat dinding rumah berdasarkan kriteria yang sesuai dan menjadi masukan kepada pemilik.

## **2.2. Tinjauan Pustaka**

### **2.2.1. Sistem**

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar (Limbong et al., 2020).

Sistem merupakan suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan (Febriani, 2018).

Berdasarkan pengertian diatas dapat disimpulkan sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

### **2.2.2. Informasi**

Informasi merupakan data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat (Aji et al., 2020).

Informasi adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dan kualitas dari peningkatan informasi (Siagian et al., 2021).

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian informasi adalah data yang diolah agar bermanfaat dalam pengambilan keputusan bagi penggunanya.

### **2.3. Sistem Pendukung Keputusan**

Sistem pendukung keputusan (*Decision Support System*) merupakan sistem informasi intraktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. DSS biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. DSS seperti itu disebut aplikasi DSS. Aplikasi DSS menggunakan CBIS (*Computer based information system*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifikasi yang tidak terstruktur. Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan (Limbong et al., 2020).

Sistem Pendukung Keputusan merupakan suatu sistem interaktif yang mendukung keputusan dalam proses pengambilan keputusan melalui alternatif-alternatif yang diperoleh dari hasil pengolahan data, informasi dan rancangan model. Sistem Pendukung Keputusan merupakan penggabungan sumber-sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah-masalah semi struktur.

### **2.4. Profile Matching**

Metode *profile matching* atau pencocokan profil adalah metode yang sering sebagai mekanisme dalam pengambilan keputusan dengan mengasumsikan bahwa

terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati (Verdian & Wantoro, 2019). Langkah-langkah dalam penyelesaian perhitungan dengan menggunakan metode *Profile Matching* yaitu:

1. Aspek Penilaian.

Langkah pertama yang harus dilakukan yaitu menentukan aspek-aspek penilaian pada *core factor* (faktor utama) dan *secondary factor* (faktor kedua).

2. Pemetaan GAP Kompetensi

GAP kompetensi adalah perbedaan antara kriteria yang dimiliki seseorang dengan kriteria yang diinginkan. Rumus GAP kompetensi yaitu:  $GAP = \text{Nilai Kriteria} - \text{Nilai Minimal}$ .

3. Pembobotan

Apabila pemetaan GAP sudah selesai dilakukan, maka hasil dari pemetaan tersebut diberi bobot nilai sesuai dengan patokan tabel bobot nilai GAP, seperti yang terlihat pada tabel berikut

**Tabel 2. 2 Nilai Pembobotan**

Selisih	Bobot Nilai	Keterangan
0	5	Tidak ada selisih (kompetensi sesuai dengan yang dibutuhkan)
1	4,5	Kompetensi individu kelebihan 1 tingkat / level
-1	4	Kompetensi individu kekurangan 1 tingkat / level
2	3,5	Kompetensi individu kelebihan 2 tingkat / level
-2	3	Kompetensi individu kekurangan 2 tingkat / level
3	2,5	Kompetensi individu kelebihan 3 tingkat / level
-3	2	Kompetensi individu kekurangan 3 tingkat / level
4	1,5	Kompetensi individu kelebihan 4 tingkat / level
-4	1	Kompetensi individu kekurangan 4 tingkat / level



4. Perhitungan dan pengelompokkan *Core Factor* dan *Secondary Factor*.

Setelah bobot nilai GAP ditentukan, maka dibagi menjadi 2 kelompok yaitu *Core Factor* dan *Secondary Factor*. Rumus untuk menghitung *Core Factor* adalah sebagai berikut:

$$NCF = \frac{\sum NC}{\sum IC} \quad (1)$$

Keterangan :

NCF : nilai rata-rata *core factor*

NC : jumlah total nilai *core factor*

IC : jumlah item *core factor*

Sedangkan rumus untuk menghitung *Secondary Factor* adalah sebagai berikut:

$$NSF = \frac{\sum SC}{\sum IS} \quad (2)$$

Keterangan :

NSF : Nilai rata-rata *Secondary Factor*

NS : Jumlah total nilai *Secondary Factor*

IS : Jumlah item *Secondary Factor*

5. Perhitungan Nilai Total

Dari perhitungan *Core Factor* dan *Secondary Factor* dari tiap tiap aspek, kemudian dihitung nilai total dari setiap aspek yang diperkirakan berpengaruh pada kinerja setiap profil. Perhitungan Nilai Total dapat ditunjukkan pada persamaan dibawah ini :

$$N = (X)\% NCF + (X)\% NSF \quad (3)$$

Keterangan :

N : Nilai total tiap aspek

- NCF : Nilai rata rata *core factor*  
 NSF : Nilai rata rata *secondary factor*  
 (X)% : Nilai presentase yang diinputkan

#### 6. Perhitungan Nilai ranking

Hasil akhir dari proses *profile matching* adalah ranking dari kandidat yang diajukan untuk mengisi suatu jabatan/posisi tertentu. Perhitungan Nilai Total dapat ditunjukkan pada persamaan dibawah ini

$$\text{Ranking} = 70\% \text{ NCF} + 30\% \text{ NSF} \quad (4)$$

Keterangan :

- NCF : Nilai *Core Factor*  
 NSF : Nilai *Secondary Factor*

### 2.5. Kriteria Sistem Pendukung Keputusan

Suatu pendekatan dasar dalam pengambilan keputusan bertujuan untuk memilih pilihan terbaik dari sejumlah alternatif dengan mempertimbangkan beberapa kriteria yaitu kesatuan (*unity*), kompleksitas (*complexity*), saling ketergantungan (*interdependence*), pengukuran (*measurement*), konsistensi (*consistent*), sintesis (*synthesis*), *trade off*, penilaian dan konsensus (*judgement and consensus*), dan pengulangan proses (*process repetition*) (Latif et al., 2018).

Kriteria yang digunakan dalam penentuan kelayakan produk ini yaitu

#### 2.5.1 Jahitan

Standar mutu jahitan dapat didefinisikan sebagai ukuran patokan (standar) yang menjadi acuan dalam menentukan kualitas (mutu) suatu produk pakaian pria maupun pakaian wanita. Untuk mencapai standar mutu yang diharapkan maka harus dilakukan pengendalian mutu yang biasa dikenal dengan istilah *quality*

*control*. Dalam industri desain fashion sendiri, *quality control* mencakup semua usaha yang dilakukan untuk mengendalikan, menyeleksi dan menilai kualitas suatu produk pakaian pria maupun pakaian wanita agar sesuai standar mutu yang diinginkan. Dengan kata lain tujuan utama dari *quality control* yaitu untuk memperoleh keuntungan dengan cara yang fleksibel, untuk menjamin agar pelanggan merasa puas dan investasi bisa kembali, dan untuk memberikan keuntungan jangka panjang bagi perusahaan (*fitinline.com, 19 Mei 2022, "standar mutu jahitan dan jenis pengendalian mutu di industri garmen", Fitinline*).

### **2.5.2 Warna Sablon**

Warna berpengaruh dikarenakan tinta sablon memiliki warna khas dan efek warna yang juga khas sehingga apabila ditempelkan pada warna dasar kaos yang tidak tepat malah akan memberikan hasil yang jelek atau tidak maksimal. Oleh karena itu, perlu dipahami apa saja bahan kaos yang umum dan juga baik digunakan untuk mendapatkan hasil sablon yang maksimal sehingga dapat anda persiapkan terlebih dahulu serta tidak asal-asalan dalam menentukan bahan kaos (*blog.porinto.com, 19 Mei 2022, "jenis sablon terbaik", Porinto*).

### **2.5.3 Sizechart**

Pola menjadi salah satu jenis pola konstruksi datar yang berasal dari Bunka Fashion College, sebuah sekolah menjahit baju barat untuk anak maupun wanita yang didirikan oleh Isaburo Namiki pada tahun 1919 di negara Jepang dan berspesialisasi dalam pengajaran desain mode. Beberapa bagian tubuh yang perlu diukur dalam proses pembuatan pola baju dengan sistem So-En diantaranya: Lingkar badan yang diukur pas pada badan tanpa tambahan ukuran untuk kelonggaran, lingkar pinggang yang diukur pas pada bagian pinggang tanpa

kelonggaran, panjang punggung yang diukur dari bagian  tengah garis leher belakang sampai ke garis pinggang, lingkaran kerung lengan yang diukur dari kerung lengan pada pola badanl lingkaran panggul yang diukur pas pada bagian tebal panggul, tinggi panggul yang diukur dari pinggang ke garis lingkaran panggul (*fitinline.com, 19 Mei 2022, “7 metode pembuatan pola dasar baju dan pengaruhnya terhadap tingkat kenyamanan busana”, Fitinline*).

#### **2.5.4 Bentuk Sablon**

Sablon merupakan hasil cetak yang dibuat dengan menggunakan template. Template ini mengacu pada alat yang digunakan dalam pencetakan yaitu layar atau screen dalam penyablonan. kriteria hasil sablon yang baik yaitu rapi, tidak terlalu tebal, kuat dan tidak mudah luntur, tintanya terserap kedalam kain dan tidak meninggalkan bekas, *rubber, plastisol, pigmen, glow in the dark* (*www.konveksibajujogja.net, 19 Mei 2022, “Kriteria & Jenis Sablon Baju Kaos Olahraga Online Murah Dan Berkualitas”, Konveksi*).

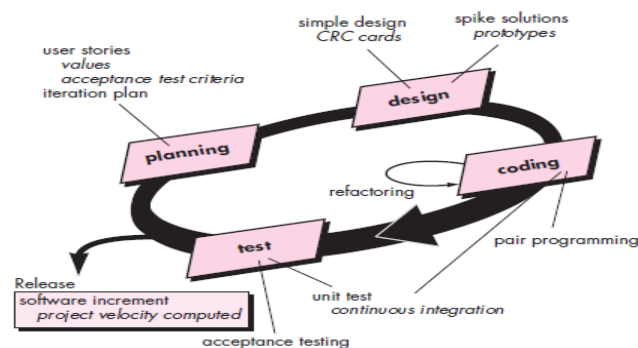
#### **2.6. Metode Pengembangan Sistem**

*Agile software engineering* merupakan salah satu model proses iteratif yang memberikan suatu alternatif yang layak dari berbagai macam metode konvensional untuk membangun berbagai jenis perangkat lunak dan berbagai macam tipe proyek pengembangan perangkat lunak. *Agile software development* adalah Metode dari beberapa kumpulan prinsip untuk pengembangan *software* di mana persyaratan dan solusi melalui upaya kolaboratif dari antar tim fungsional dan klien sebagai pendukung perencanaan adaptif, perkembangan evolusi, awal pengiriman, dan perbaikan terus-menerus, dan itu mendorong respon yang cepat dan fleksibel untuk

dirubah. Prinsip-prinsip ini mendukung definisi dan evolusi dari banyak metode pengembangan perangkat lunak (Sulistiani et al., 2020).

*Agile software development* interaksi dan personel lebih penting daripada proses dan alat, *software* yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan *client* lebih penting daripada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana. Namun demikian, sama seperti model proses yang lain, *agile software development* memiliki kelebihan dan tidak cocok untuk semua jenis proyek, produk, orang dan situasi. *Agile software development* memungkinkan model proses yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat cepat ditanggapi.

*Extreme Programming (XP)* adalah metode pengembangan *software* yang cepat, efisien, beresiko rendah, *fleksibel*, terprediksi, *scientific*, dan menyenangkan. *Extreme Programming (XP)* merupakan suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat (Indra et al., 2020). Alasan menggunakan metode XP karena sifat dari aplikasi yang di kembangkan dengan cepat melalui tahapan-tahapan yang ada meliputi : *Planning* (perencanaan), *Design* (perancangan), *Coding* (Pengkodean), dan *Test* (pengujian).



**Gambar 2.1** *Extreme Programming*

Tahapan dalam metode pengembangan sistem *Extreme Programming* yaitu :

### **2.6.1 Planning**

Pada tahap perencanaan ini dimulai dari pengumpulan kebutuhan yang membantu tim teknis untuk memahami konteks bisnis dari sebuah aplikasi. Selain itu pada tahap ini juga mendefinisikan *output* yang akan dihasilkan, fitur yang dimiliki oleh aplikasi dan fungsi dari aplikasi yang dikembangkan. *Planning* Membentuk *user stories*, menentukan *cost*. Semua *story* segera diimplementasikan (dalam beberapa minggu) *Story* dengan *value* tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama. *Story* dengan resiko paling tinggi akan diimplementasikan lebih dulu. Setelah project pertama direlease dan didelivery, XP team memperhitungkan kecepatan project.

### **2.6.2 Design**

Metode ini menekankan desain aplikasi yang sederhana, untuk mendesain aplikasi dapat menggunakan *Class-Responsibility-Collaborator (CRC) cards* yang mengidentifikasi dan mengatur *class* pada *object-oriented*. Design menggunakan CRC card, untuk mengenali dan mengatur *object oriented class* yang sesuai dengan *software increment*.

### **2.6.3 Coding**

Konsep utama dari tahapan pengkodean pada *extreme programming* adalah *pair programming*, melibatkan lebih dari satu orang untuk menyusun kode.

### **2.6.4 Test**

Pada tahapan ini lebih fokus pada pengujian fitur dan fungsionalitas dari aplikasi.

## 2.7. UML

*Unified Modeling Language (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. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

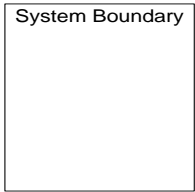

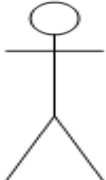

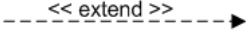


*Unified Modeling Language (UML)* adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Jacobson & Booch, 2021).

### 2.7.1. Use Case Diagram

Menurut Jacobson and Booch (2021), *use case diagram* adalah diagram yang menggambarkan interaksi antara sistem, eksternal sistem dan *user*. Dengan kata lain, diagram ini menjelaskan sistem tersebut dan bagaimana cara *user* berinteraksi dengan sistem. *Use case diagram* menunjukkan hubungan statis antara *actor* dan *use case* dalam sistem. Mereka menyediakan pandangan awal dari struktur sistem. *Use case* berguna dalam membangun dan mengkomunikasikan pandangan umum sistem. Mereka menyediakan satu titik awal untuk desain, khususnya objek identifikasi dan diagram urutan. Elemen-elemen diagram *use case* adalah *use case*, aktor, kegunaan, dan tanda panah. Aktor diwakili oleh segala sesuatu yang berada diluar sistem, seperti jenis pengguna atau sistem eksternal, yang berinteraksi dengan sistem.

Simbol-simbol yang digunakan dalam *use case diagram*

**Tabel 2. 3 Simbol-simbol Use Case Diagram**

Simbol	Nama Simbol	Keterangan
	<i>Boundary System</i>	Digambarkan dengan kotak disekitar <i>use case</i> , untuk menggambarkan jangkauan sistem anda ( <i>scope of of your system</i> ). Biasanya digunakan apabila memberikan beberapa alternatif sistem yang dapat dijadikan pilihan.
	<i>Use-Case</i>	Gambaran fungsional sistem yang akan di buat, agar pengguna lebih mengerti penggunaan <i>system</i>
	<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga dianggap sebagai <i>actor</i> .
	<i>Association</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>actor</i> dengan <i>use case</i> .
	<i>Extend</i>	Metode yang hanya berjalan di bawah kondisi tertentu.
	<i>Include</i>	Metode yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi.
	<i>Generalization</i>	Sebuah elemen yang menjadi spesialisasi dari elemen yang lain.

**Sumber :** (Jacobson & Booch, 2021).

### 2.7.2. Activity Diagram


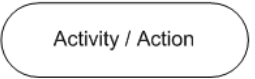

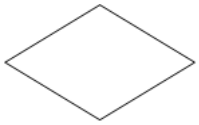
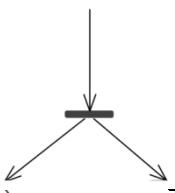
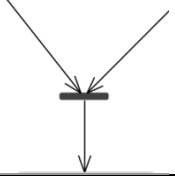


Menurut Jacobson and Booch (2021), *activity diagram* digunakan untuk menggambarkan alur dari proses bisnis atau langkah-langkah *use case* secara berurutan. Diagram ini juga digunakan untuk menggambar *action* (tindakan) yang



akan dieksekusikan ketika suatu proses sedang berjalan dan beserta hasil dari proses eksekusi tersebut.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**Tabel 2. 4 Simbol-simbol *Activity Diagram***

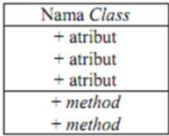




Simbol	Nama Simbol	Keterangan
	<i>Initial node</i>	Awal sebuah proses.
	<i>Action</i>	Urutan tindakan membentuk total aktivitas yang ditunjukkan oleh diagram.
	<i>Flow</i>	Kebanyakan aliran tidak membutuhkan kata-kata untuk mengidentifikasi mereka kecuali keluar dari keputusan.
	<i>Decision</i>	Aliran yang keluar ditandai untuk menunjukkan kondisi.
	<i>Fork</i>	bar hitam dengan satu alur masuk dan dua atau lebih alur keluar, aksi di bawah percabangan dapat terjadi dalam urutan apapun atau bahkan secara bersamaan.
	<i>Join</i>	bar hitam dengan dua atau lebih alur masuk dan satu alur keluar untuk menyatukan lagi alur aksi yang dipisahkan oleh <i>fork</i> .
	<i>Activity Final</i>	mewakili akhir proses atau aktivitas.
	<i>Swimlane</i>	pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

**Sumber :**(Jacobson & Booch, 2021).

### 2.7.3. Class Diagram

*Class diagram* merupakan gambaran grafis dari struktur objek statis dari sebuah sistem yang menunjukkan kelas objek yang tersusun dari hubungan antara kelas- kelas objek yang lain. *Class diagram* digunakan untuk menggambarkan tampilan desain statis sebuah sistem (Jacobson & Booch, 2021).

**Tabel 2. 5 Simbol-simbol *Class Diagram***

Simbol	Nama Simbol	Keterangan
	<i>Class</i>	<i>Class</i> terdiri atas 3 bagian yaitu bagian atas, bagian tengah, dan bagian bawah. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan <i>property</i> /atribut <i>class</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .
	<i>Association</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> .
	<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>composition</i> terhadap <i>class</i> tempat dia bergantung tersebut.
	<i>Dependency</i>	Kadang kala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.
	<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi

**Sumber :** (Jacobson & Booch, 2021).

## 2.8. ISO 25010

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. Berikut ini merupakan karakteristik atau faktor kualitas internal dan eksternal yang terdapat pada model ISO-25010 (Peters & Aggrey, 2020).

**Tabel 2. 6 ISO 25010**

Indikator	Sub – Indikator
<i>Functional Suitability</i>	2.6.4.1 <i>Functional Completeness</i> 2.6.4.2 <i>Functional Correctness</i> 2.6.4.3 <i>Functional Appropriateness</i>
<i>Performance Efficiency</i>	1. <i>Time behaviour</i> 2. <i>Resource utilization</i> 3. <i>Capacity</i>
<i>Compatibility</i>	1. <i>Co-existence</i> 2. <i>Interoperability</i>
<i>Usability</i>	1. <i>Appropriateness Recognisability</i> 2. <i>Learnability</i> 3. <i>Operability</i> 4. <i>User error protection</i> 5. <i>User interface aesthetics</i> 6. <i>Accessibility</i>
<i>Reliability</i>	1. <i>Maturity</i> 2. <i>Availability</i> 3. <i>Fault tolerance</i> 4. <i>Recoverability</i>
<i>Security</i>	1. <i>Confidentiality</i> 2. <i>Integrity</i> 3. <i>Non-repudiation</i> 4. <i>Accountability</i> 5. <i>Authenticity</i>
<i>Maintainability</i>	1. <i>Modularity</i> 2. <i>Reusability</i> 3. <i>Analyzability</i>

Berikut ini merupakan pengertian dari masing-masing faktor dan sub-faktor yang terdapat pada model ISO-25010, antara lain :

1. *Functional Suitability*, seberapa jauh sistem mampu memberikan fungsi yang memenuhi kriteria kebutuhan yang ada (Peters & Aggrey, 2020).
  - a. *Functional Completeness*: Seberapa jauh rangkaian fungsi tersebut mencangkup semua tujuan dan tugas penggunanya.
  - b. *Functional Correctness*: Seberapa jauh produk tersebut dapat memberikan/membenarkan hasil yang sesuai dengan tingkatan tertentu sesuai kebutuhan.
  - c. *Functional Appropriateness*: Seberapa jauh fungsi dari sistem tersebut memberikan fasilitas dalam penyelesaian tugas dan tujuan yang sudah ditentukan
2. *Performance Efficiency*, seberapa jauh tingkat kemampuan kapasitas sistem yang relative baik dengan jumlah sumber daya yang digunakan (Peters & Aggrey, 2020).
  - a. *Time Behaviour*: Seberapa jauh respon dan waktu proses dan tingkat hasil dari suatu sistem pada saat fungsi tersebut dijalankan dalam melengkapi persyaratan yang diberikan.
  - b. *Resource Utilization*: Seberapa banyak sumber daya yang dipakai pada produk saat fungsi tersebut dijalankan sebagai pemenuh persyaratan.
  - c. *Capacity*: Seberapa jauh batas maksimal atau daya tampung dari produk berdasarkan parameter sistem yang memenuhi persyaratan.
3. *Compatibility*, seberapa jauh sistem tersebut mampu bertukar informasi dalam menjalankan sistem lainnya yang digunakan dan secara bersamaan di

berbagai lingkungan perangkat lunak maupun keras yang sama (Peters & Aggrey, 2020).

- a. *Co-existence*: Seberapa jauh sistem tersebut mampu melakukan fungsi yang lebih efisien dengan cara berbagi lingkungan dan sumber daya antara satu sistem dengan yang lain. Tidak perlu memberikan dampak merugikan untuk sistem tersebut.
  - b. *Interoperability*: Seberapa jauh satu, dua atau lebih dan sistem tersebut dapat bertukar data serta memakai informasi yang sudah ditukarkan sebelumnya.
4. *Usability*, yaitu seberapa jauh sistem tersebut mampu digunakan oleh penggunanya dalam menempuh tujuan yang sudah ditentukan dengan efektifitas, efisiensi, dan kepuasan. (Peters & Aggrey, 2020).
- a. *Appropriateness Recognizability*: Seberapa jauh pemakai mampu mengenali produk tersebut, apakah sudah sesuai dengan keperluan mereka atau kurang sesuai.
  - b. *Learnability*: Seberapa jauh produk tersebut agar dapat dipakai oleh pengguna agar dapat mencapai pembelajaran yang sudah ditentukan dalam penggunaan produk. Terutama dibutuhkan pencapaian Kefektivitas, efisiensi, kebebasan dari resiko serta kepuasan dalam penggunaan tertentu.
  - c. *Operability*: Seberapa jauh produk tersebut memiliki kemampuan yang membuat produk tersebut lebih mudah untuk digunakan.
  - d. *User Error Protection*: Seberapa jauh sistem tersebut mampu melindungi pengguna jika terjadi suatu kegagalan.

- e. *User Interface Aesthetics*: Seberapa jauh antarmuka dari pengguna/user mampu untuk menciptakan suatu interaksi yang baik untuk pengguna.
  - f. *Accessibility*: Seberapa jauh produk tersebut dapat digunakan oleh berbagai kalangan dengan jangkauan karakteristik dengan mencapai tujuan dalam konteks tertentu.
5. Reliability, seberapa jauh sistem dapat menjalankan fungsi yang ditentukan selama batas waktu yang ditentukan (Peters & Aggrey, 2020).
- a. *Maturity*: Seberapa jauh sistem, produk/kreasi atau unsur yang melengkapi keperluan untuk keunggulan dalam operasi standart.
  - b. *Availability*: Seberapa jauh metode, produk/kreasi, atau unsur siap dioperasikan dan bisa diakses ketika dibutuhkan untuk digunakan.
  - c. *Fault Tolerance*: Seberapa jauh metode, produk/kreasi, atau unsur yang akan beroperasi seperti halnya walaupun ada kegagalan hardware (perangkat keras) atau software (perangkat lunak).
  - d. *Recoverability*: Seberapa jauh, ketika suatu hal terjadi kelangsungan kendala atau rintangan, suatu sistem atau struktur tersebut dapat mengembalikan data yang terkena kerusakan langsung dan dapat membentuk kembali ke sistem yang diinginkan.
6. *Security*, seberapa jauh sistem dapat melindungi data dan informasi yang diakses (Peters & Aggrey, 2020).
- a. *Confidentiality*: Seberapa jauh produk tersebut memastikan jika data tersebut hanya bisa dimasuki sama pihak yang berkuasa saja.
  - b. *Integrity*: Seberapa jauh sistem tersebut mencegah terjadinya jalur masuk yang tidak valid, adanya perubahan program ataupun data.

- c. *Non-repudiation*: Seberapa jauh sistem tersebut dapat memebrikan bukti jika suatu tindakan telah terjadi sehingga tidak akan ada penyangkalan pada peristiwa tersebut.
  - d. *Authenticity*: Seberapa jauh identitas seseorang dapat dibuktikan sebagai yang diklaim.
  - e. *Accountability*: Seberapa jauh tindakan dari suatu identitas dapat dilacak dan ditelusuri.
7. *Maintainability*, seberapa jauh keefektifan dan keefisiensian sistem dapat dirawat (Peters & Aggrey, 2020).
- a. *Modularity*: Seberapa jauh sistem dapat memperkecil dampak terhadap komponen lain jika terjadi modifikasi pada salah satu komponen.
  - b. *Reusability*: Seberapa jauh sistem dapat dipakai sehingga dapat membangun aset lain.
  - c. *Analysability*: Seberapa jauh sistem dapat mengkaji dampak perubahan sistem untuk mendiagnosis kekurangan.
  - d. *Modifiability*: Seberapa jauh suatu hasil atau metode tersebut bisa berguna secara efektif dan efisien dirubah tanpa ada memperlihatkan kegagalan atau diturunkannya suatu kualitas produk tersebut.
  - e. *Testability*: Seberapa jauh nilai level efektivitas dan efisiensi yang sesuai dengan kebutuhan pengujian dan bisa diterapkan oleh suatu sistem, produk atau komponen serta dicek juga dipengujian tersebut apakah nilai itu telah terpenuhi atau belum.
8. *Portability*, seberapa jauh sistem dapat ditransfer atau dipindahkan dari satu perangkat keras atau lunak ke *hardware* (perangkat keras) atau software

(perangkat lunak) yang lain pada lingkungan operasional yang berbeda (Peters & Aggrey, 2020).

- a. *Adaptability*: Seberapa jauh suatu hasil atau metode tersebut yang efektif maupun efisien dan dilakukan penyesuaian untuk hardware (perangkat keras), *software* (perangkat lunak) maupun lingkungan Koperasional / penggunaanyang berbeda.
- b. *Installability*: Seberapa jauh perangkat sistem tersebut berhasil diinstal (dipasangkan) dan / atau dihapus (dilepaskan) dalam tindakan tertentu.
- c. *Replaceability*: Seberapa jauh penggunaan sistem tersebut bisa merubah sistem lainnya dan perubahan/penggantian tersebut ditentukan dari tujuan yang sama pada lingkungan yang sama.