

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini, penulis melakukan tinjauan pustaka terhadap beberapa penelitian-penelitian terdahulu sebagai penunjang penelitian yang dilakukan penulis. Dibawah ini adalah tinjauan literatur yang telah diteliti sebelumnya:

Tabel 2.1 Tinjauan Pustaka

1	Judul	Analisis dan Perancangan Sistem Informasi <i>martketplace</i> Berbasis Web Pada UMKM Batik Rindani Jambi
	Penulis	Dedy Setiawan dan Lutfi
	Tahun	2018
	Latar Belakang Masalah	UMKM kesulitan dalam melakukan Penjualan dan promosi produk yang menjangkau pasar yang lebih luas sehingga berdampak kurang efektifitas dan efisiensi tata kelola Penjualan dan promosi produk khususnya UMKM Batik Rindiani baik dari segi biaya, tenaga maupun waktu
	Metode Pengembangan	SDLC (<i>sistem development life cycle</i>)
	Objek Penelitian	UMKM Produk fashion (Batik)
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Meningkatkan efektifitas dan efisiensi tata kelola Penjualan dan promosi produk baik dari segi biaya, tenaga, maupun waktu.
	Hasil Penelitian	Aplikasi <i>Martketplace</i> berbasis Web.
	Lokasi Penelitian	Umkm batik rindani, kec. Jelutung, jambi.
2	Judul	Perancangan <i>Martketplace</i> Sebagai Media Pemasaran Kerajinan Bambu
	Penulis	F. Lia dwi cahyanti, fajar sarasati, widiastuti, elly firasari
	Tahun	2021
	Latar Belakang Masalah	UKM Kerajinan Bambu Brajan di Dukuh Brajan, Desa Sendangagung, Kecamatan Minggir, Kabupaten Sleman yang mayoritas masih menggunakan Penjualan tradisional dan

		belum dapat beralih ke sistem komputerisasi Penjualan produk-produk mereka. Hal tersebut dikarenakan beberapa kendala diantaranya adalah SDM yang belum memahami internet, Penjualan masih dilakukan door to door, hanya mengandalkan reseller dan event seperti pameran. Produk UKM Kerajiana Bambu Brajan berkualitas ekspor karena produk yang dihasilkan sangat unik serta bernilai jual tinggi. Strategi yang dilakukan untuk membantu pemasaran masih dirasa kurang membantu meningkatkan produktivitas dan pemasarannya. Cara promosi dan Penjualan yang masih tradisional dan manual mengakibatkan beberapa masalah yaitu produk kurang dikenal masyarakat luas, pemasaran dan Penjualan produk jangkauannya sempit serta omset Penjualan yang belum sesuai target yang ditentukan
	Metode Pengembangan	Waterfall
	Objek Penelitian	Ukm produk kerajinan bambu
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Untuk merancang <i>marketplace</i> sebagai media pemasaran kerajinan bambu pada UKM di desa Brajan. Dengan dibuatnya website <i>marketplace</i> sebagai pemasaran produk diharapkan mampu membantu memasarkan produk sehingga dapat dikenal masyarakat luas
	Hasil Penelitian	Sistem informasi UMKM Kerajinan Bambu Brajan berbasis web
	Lokasi Penelitian	Dukuh brajan, desa sendangagung, kec. Minggir, kab. Sleman
3	Judul	Pembuatan Aplikasi <i>Marketplace</i> Berbasis Web Pada UMKM Kopi UD. Arta Merta Banjar Badingkayu Desa Pengeragoan, Jembrana.
	Penulis	I wayan ardiyasa, i putu candra wibawa
	Tahun	2018
	Latar Belakang Masalah	Didapati kesulitan dalam memasarkan produk kopi bubuk kemasan. Promosi dan Penjualan produk kopi bubuk kemasan ini masih manual dimana produk ini dijual dengan cara dititipkan ke warung-warung kecil di sekitar

		rumah serta menerima pesanan giling kopi dari warga setempat. Pelaku usaha biasanya mengikuti kegiatan pameran makanan yang diadakan oleh pemkab jembrana provinsi bali. Dengan cara manual tersebut membuat Penjualan tidak mampu meningkat secara signifikan
	Metode Pengembangan	Metode dalam pengabdian masyarakat (Observasi, Studi Literatur, Sosialisasi ke Mitra, Implementasi Sistem)
	Objek Penelitian	Umkm produk kopi
	Bahasa Pemrograman	Wordpress CMS (PHP)
	Tujuan Penelitian	Membantu mitra mengelola produk kopi serta mempromosikan produk kopi kemasan secara digital.
	Hasil Penelitian	Aplikasi <i>marketplace</i> berbasis web untuk membantu mitra melakukan fungsi manajemen (pengelolaan, promosi, Penjualan) produk kopi kemasan.
	Lokasi Penelitian	Banjar badingkayu, desa pengeragoan, jembrana, bali.
4	Judul	<i>Marketplace</i> Pada Umkm Gerabah Desa Wiralaga Di Mesuji
	Penulis	Nosiel dan Isnandar Agus
	Tahun	2020
	Latar Belakang Masalah	Desa Wiralaga II kabupaten mesuji yang merupakan desa kerajinan gerabah memiliki potensi untuk ditingkatkan pemasarannya namun kenyataannya dalam pemasaran hanya untuk kalangan tertentu sehingga usaha yang dijalankan tidak maksimal dan terkadang mengalami kerugian yang cukup besar
	Metode Pengembangan	<i>Prototype</i>
	Objek Penelitian	Umkm produk kerajinan gerabah
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Membantu Usaha mikro, kecil dan menengah (UMKM) dalam meningkatkan pemasaran produk-produk UMKM yang ada didesa wiralaga II
	Hasil Penelitian	Aplikasi Web <i>Marketplace</i> untuk UMKM desa Wiralaga
	Lokasi Penelitian	Desa wiralaga ii, kab. Mesuji.

5	Judul	Rancang Bangun <i>Marketplace</i> Berbasis Web untuk UMKM Batik
	Penulis	Syahrul nugraha, ryan hamonangan, raditya danar dana, edi tohidi, umi hayati
	Tahun	2022
	Latar Belakang Masalah	Masalah terletak pada proses bisnis yang tidak menggunakan konsep Business to Customer (B2C), artinya pengrajin hanya mendapatkan upah dari pembuatan batik, sedangkan toko mendapatkan untung yang lebih besar dari Penjualan batik tersebut.
	Metode Pengembangan	Software development life cycle (sldc)
	Objek Penelitian	Ukm produk batik
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Tujuan dari penelitian ini yaitu untuk memutus distribusi Penjualan yang sangat menguntungkan toko Penjualan batik sedangkan pengrajin hanya mendapatkan untung dari upah pembuatan batik.
	Hasil Penelitian	Sistem informasi Penjualan online berbasis web pada UMKM Batik Trusmi
	Lokasi Penelitian	Cirebon
6	Judul	Rancang Bangun Aplikasi <i>Marketplace</i> Pada Ukm Berbasis Web Sebagai Sarana Untuk Meningkatkan Pemasaran Produk (Studi Kasus : Ukm Hoki Donut's)
	Penulis	Widyagustina ismayanti, sewaka
	Tahun	2021
	Latar Belakang Masalah	Terbatasnya jaringan pemasaran seperti media maupun fasilitas pemasaran menyebabkan sulitnya untuk memasarkan produk-produknya, sehingga pelanggannya juga terbatas hanya area kota Jakarta dan sekitar rumah produksinya saja
	Metode Pengembangan	<i>Waterfall</i>
	Objek Penelitian	UMKM Hoki Donut's (Makanan: aneka kue)
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Membangun dan mengembangkan website bagi UMKM Hoki Donut's yang dapat digunakan sebagai media informasi atau media pemasaran hasil produk-produknya agar dapat

		diketahui oleh khalayak luas guna menghadapi persaingan pemasaran di dunia teknologi
	Hasil Penelitian	Sistem Informasi Penjualan <i>marketplace</i> berbasis web UMKM Hoki Donut's
	Lokasi Penelitian	Kebon pala, cawang, jakarta timur
7	Judul	Rancang Bangun Sistem Informasi <i>Marketplace</i> di UKM Aneka Kebaya Berbasis Web “(Studi Kasus : Baju Kebaya dan Rok Batik di Koto Tengah Simalanggang)”
	Penulis	Nadya prima, ahmaddul hadi
	Tahun	2021
	Latar Belakang Masalah	Semua pengolahan data nya masih manual yaitu pengolahan data produk dan laporan Penjualan. Pembuatan data produk dan laporan masih menggunakan kertas yang tentunya memakan banyak biaya untuk mencetaknya
	Metode Pengembangan	<i>Prototype</i>
	Objek Penelitian	UMKM Baju Kebaya dan Rok Batik
	Bahasa Pemrograman	<i>Framework laravelr, php</i>
	Tujuan Penelitian	Membuat sistem informasi <i>Marketplace</i> di UKM Aneka Kebaya
	Hasil Penelitian	Sistem informasi <i>marketplace</i> umkm aneka kebaya.
	Lokasi Penelitian	Koto tengah simalanggang, padang
8	Judul	Rancang Bangun Aplikasi E Commerce Berbasis Website Pada Umkm Batu Cincin Mandiri Surabaya
	Penulis	Abdul rizal bungsu, ayovi poerna wardhanie, nunuk wahyuningtyas
	Tahun	2021
	Latar Belakang Masalah	Adanya pandemi covid-19 yang melanda Indonesia pada bulan Maret 2020 menyebabkan penurunan omset Penjualan pada UMKM Batu Cincin, dikarenakan pada saat ini tidak memungkinkan bagi pelanggan untuk datang langsung ke toko dan dengan adanya penerapan PSBB (Pembatasan Sosial Berskala Besar) oleh pemerintah mengakibatkan berkurangnya jam operasional usaha dan lambatnya saluran distribusi. Berdasarkan permasalahan tersebut,

	Metode Pengembangan	Waterfall
	Objek Penelitian	Umkm batu cincin
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Membuat sebuah aplikasi <i>martketplace</i> yang dapat meningkatkan Penjualan serta dapat menjangkau wilayah pemasaran yang lebih luas.
	Hasil Penelitian	Penelitian ini menghasilkan sebuah aplikasi <i>martketplace</i> berbasis website yang dapat mengelola data, melakukan proses Penjualan, dan pembuatan laporan.
	Lokasi Penelitian	Surabaya
9	Judul	Rancang Bangun <i>Marketplace</i> Hasil Panen Dan Olahan Hasil Tani Berbasis Website
	Penulis	Tjut awaliyah zuraiyah, halimah tus sadiah, yetty husnul hayati, lia dahlia iryani, diki candra suandi
	Tahun	2021
	Latar Belakang Masalah	Serikat Petani Indonesia menyatakan bahwa masalah utama yang dialami petani hingga saat ini adalah kesulitan pemasaran hasil panen. Harga hasil panen banyak ditetapkan oleh tengkulak sehingga para petani mengalami kerugian karena modal kerja yang sudah dikeluarkan tidak sebanding dengan hasil Penjualan hasil panen yang diperoleh. Selain itu kondisi pandemi Covid 19 juga mempengaruhi proses Penjualan hasil panen.
	Metode Pengembangan	<i>Waterfall</i>
	Objek Penelitian	UMKM Hasil Panen dan Olahan hasil tani
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Membuat sistem Penjualan online atau <i>martketplace</i> yang dapat memfasilitasi para petani untuk menjual hasil panen dan olahan taninya langsung baik kepada distributor maupun langsung kepada konsumen
	Hasil Penelitian	Produk berupa <i>martketplace</i> hasil panen dan olahan tani yang memiliki fitur menampilkan daftar produk dan harganya, fitur pemesanan, fitur penawaran, fitur laporan Penjualan produk dan grafik Penjualan.
	Lokasi Penelitian	Pakuan, bogor

10	Judul	Rancang Bangun Aplikasi <i>Marketplace</i> Untuk Usaha Penjualan Helm
	Penulis	Tyas armando, ade dwi putra
	Tahun	2020
	Latar Belakang Masalah	Toko Edi Helm Bandar Lampung yang masih menggunakan sistem konvensional, dimana mengharuskan konsumen datang langsung ke toko untuk melihat, memilih dan membeli helm yang diinginkan. Mengakibatkan informasi produk yang di jual maupun informasi toko itu sendiri masih kurang efektif dengan jangkauan promosi yang tidak luas.
	Metode Pengembangan	Extreme programming
	Objek Penelitian	Ukm Penjualan helm
	Bahasa Pemrograman	Php
	Tujuan Penelitian	Merancang dan membangun website yang mendukung aplikasi <i>marketplace</i> pada Toko Edi Helm Bandar Lampung untuk mempermudah dalam melayani pembelian secara online dan pengelolaan barang.
	Hasil Penelitian	Sebuah website <i>marketplace</i> yang akan membantu Toko Edi Helm Bandar Lampung dalam melakukan Penjualan secara online sehingga dapat menjangkau lebih banyak minat konsumen dan memperoleh pasar yang besar.
Lokasi Penelitian	Bandar lampung	
11	Judul	Rancang Bangun <i>Marketplace</i> Untuk Pemasaran Dan Penjualan Produk Ukm Desa Sindangsari
	Penulis	Erliyana
	Tahun	2022
	Latar Belakang Masalah	UMKM pada desa Sindangsari masih melakukan kegiatan perdagangan secara tradisional dimana pemasaran tidak meluas dan tidak mengalami perkembangan dengan baik, terlebih lagi Covid-19 yang sangat membuat UMKM di desa sindangsari mengalami penurunan yang signifikan secara ekonomi.
	Metode Pengembangan	<i>Prototype</i>

Objek Penelitian	UMKM Produk Makanan Desa Sindangsari.
Bahasa Pemrograman	PHP dengan <i>Framework Codeigniter</i>
Tujuan Penelitian	Menerapkan metode <i>extreme programming</i> untuk membangun aplikasi website <i>martketplace</i> dalam pemasaran dan Penjualan produk UMKM desa Sindangsari, Lampung Selatan.
Hasil Penelitian	Diharapkan menghasilkan Aplikasi <i>Marketplace</i> yang dapat membantu Penjualan dan pemasaran produk makanan UMKM desa Sindangsari
Lokasi Penelitian	Desa sindangsari, kec. Tanjung bintang, kab. Lampung selatan.

2.2 Landasan Teori

2.2.1 *Marketplace*

Menurut (Supriatman dan Sidiq, 2022), *Marketplace* adalah pembelian, Penjualan dan pemasaran barang serta jasa melalui sistem elektronik. Seperti televisi, radio dan jaringan komputer atau internet . Electronic Commerce atau *Marketplace* dapat diartikan sebagai sarana dalam Penjualan maupun pembelian barang atau jasa dengan menggunakan media elektronik baik melalui radio, televisi, handphone, bahkan internet. *Marketplace* dibagi menjadi 3 jenis :

- a. *Marketplace* bisnis ke konsumen (B2C) melibatkan Penjualan produk dan layanan secara eceran kepada pembeli perorangan. Pada penelitian ini jenis *Marketplace* yang ada adalah bisnis ke konsumen (B2C).
- b. *Marketplace* bisnis ke bisnis (B2B) melibatkan Penjualan produk dan layanan antar perusahaan.
- c. *Marketplace* konsumen ke konsumen (C2C) melibatkan konsumen yang menjual secara langsung ke konsumen.

Tujuan dari *Marketplace* adalah sebagai berikut:

- a. Membantu dalam proses transaksi dalam Penjualan maupun pembelian barang dengan mengakses internet.
- b. Pengolahan yang berorientasi pada pelayanan, kombinasi konvensional dan virtual: responsif, dinamis, informatif, dan komunikatif.
- c. Model pembayaran kartu kredit atau transfer.

Marketplace memiliki beberapa komponen standar yang dimiliki dan tidak dimiliki transaksi bisnis yang dilakukan secara *offline*, yaitu :

- a. Produk : banyak jenis produk yang bisa dijual melalui internet seperti komputer, buku, musik, pakaian, mainan, dan lain- lain. Produk yang dijual dalam penelitian ini adalah produk olahan makanan.
- b. Tempat menjual produk (*a place to sell*) : tempat menjual adalah internet yang berarti harus memiliki domain dan hosting.
- c. Cara menerima pesanan : email, telepon, sms, web *payment*. Pemesanan atau komunikasi dengan Penjual dalam penelitian ini menggunakan Aplikasi Whatsapp.
- d. Cara pembayaran : Cash, cek, bankdraft, kartu kredit, internet *payment*. Pembayaran dalam penelitian ini dilakukan dengan menggunakan internet *payment*.
- e. Metode pengiriman : pengiriman bisa dilakukan melalui paket, *salesman*, atau didownload jika produk yang dijual memungkinkan untuk itu (misalnya *software*). Pada penelitian ini metode pengiriman yang akan digunakan ialah melalui *salesman*, dan untuk luar kabupaten digunakan pengiriman paket.

2.2.2 UMKM

Menurut (Hanim, 2018) yang dimaksud dengan usaha mikro, kecil dan menengah (UMKM) adalah unit usaha produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan

atau badan usaha disemua sektor ekonomi. Menurut (Hanim, 2018) UMKM dapat diklasifikasikan menjadi 4 kelompok yaitu:

- a. *Livelihood Activities*, merupakan UMKM yang digunakan sebagai kesempatan kerja untuk mencari nafkah, yang lebih umum dikenal sebagai sektor informal, contohnya adalah pedagang kaki lima.
- b. *Micro Enterprise* merupakan UMKM yang memiliki sifat pengrajin tetapi belum memiliki sifat kewirausahaan.
- c. *Small Dynamic Enterprise*, merupakan UMKM yang telah memiliki jiwa kewirausahaan dan mampu menerima pekerjaan subkontrak dan ekspor.
- d. *Fast Moving Enterprise*, merupakan UMKM yang memiliki jiwa kewirausahaan dan akan melakukan tranformasi menjadi Usaha Besar (UB).

Ketentuan tentang usaha kecil yang berlaku selama ini perlu disesuaikan dengan kondisi masa kini, antara lain terkait dengan Badan usaha yang menjadi jati diri dari pelaku usaha. Undang-Undang UMKM secara normatif memberikan rumusan tentang kriteria UMKM sebagai berikut:

- a. Usaha mikro adalah usaha produktif milik orang perorangan dan/atau badan usaha perorangan. Memiliki kekayaan bersih paling banyak Rp. 50.000.000,00 (lima puluh juta rupiah) tidak termasuk tanah dan bangunan tempat usaha atau memiliki hasil Penjualan tahunan paling banyak Rp. 300.000.000,00 (tiga ratus juta rupiah)
- b. Usaha kecil adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau bukan cabang perusahaan yang dimiliki, dikuasai atau menjadi bagian, baik langsung maupun tidak langsung dari usaha menengah atau usaha besar yang memenuhi kriteria usaha kecil. Memiliki kekayaan bersih lebih dari Rp. 50.000.000,00 (lima puluh juta rupiah) sampai

dengan paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah) sampai dengan paling banyak Rp. 2.500.000.000,00 (dua miliar lima ratus juta rupiah)

- c. Usaha menengah adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian, baik langsung maupun tidak langsung dengan usaha kecil atau usaha besar dengan jumlah kekayaan bersih atau hasil Penjualan tahunan. Memiliki kekayaan bersih lebih dari Rp. 500.000.000,00 (lima ratus juta rupiah) sampai dengan paling banyak Rp. 10.000.000.000,00 (sepuluh miliar rupiah) tidak termasuk tanah dan bangunan tempat usaha, atau memiliki hasil Penjualan tahunan lebih dari Rp. 2.500.000.000,00 (dua miliar lima ratus juta rupiah) sampai dengan paling banyak Rp. 50.000.000.000,00 (lima puluh miliar rupiah)
- d. Usaha besar adalah usaha ekonomi produktif yang dilakukan oleh badan usaha dengan jumlah kekayaan bersih atau hasil Penjualan tahunan lebih besar dari usaha menengah, yang meliputi usaha nasional milik negara atau swasta, usaha patungan, dan usaha asing yang melakukan kegiatan ekonomi di Indonesia.

2.2.3 Penjualan

Penjualan adalah kegiatan ekonomi yang dilakukan dalam kehidupan sehari-hari (Utami, 2021). Tidak hanya perusahaan besar yang melakukan Penjualan, Penjualan dilakukan oleh manusia secara umum. Menjual berbagai jenis jasa dan barang termasuk ke dalam Penjualan. Berikut adalah beberapa definisi *selling* atau Penjualan menurut para ahli:

- a. Philip Kotler Philip Kotler dalam buku berjudul Manajemen Pemasaran (2008) menyebutkan bahwa Penjualan adalah proses sosial manajerial di mana individu dan

kelompok mendapatkan apa yang mereka butuhkan dan inginkan, menciptakan, menawarkan, dan mempertukarkan produk yang bernilai dengan pihak lain.

- b. William G. Nickels William G. Nickels dalam buku *Understanding Business* (1998) menyebutkan bahwa Penjualan yaitu proses di mana sang Penjual memuaskan segala kebutuhan dan keinginan pembeli agar dicapai manfaat baik bagi sang Penjual maupun sang pembeli yang berkelanjutan dan menguntungkan kedua belah pihak.
- c. Sofjan Assauri Sofjan Assauri dalam buku berjudul *Manajemen Pemasaran* (2002) menyebutkan Penjualan merupakan kegiatan manusia yang diarahkan untuk memenuhi atau memuaskan kebutuhan dan keinginan melalui proses pertukaran.
- d. Winardi dalam buku *Pengantar Manajemen Penjualan (Sales Management)* (1991) menyebutkan bahwa Penjualan adalah proses di mana sang Penjual memuaskan segala kebutuhan dan keinginan pembeli agar dicapai manfaat baik bagi sang Penjual maupun sang pembeli agar dicapai manfaat baik bagi sang Penjual maupun sang pembeli.

Ada beberapa bentuk Penjualan yaitu Penjualan tunai, Penjualan kredit, Penjualan tender, Penjualan ekspor, Penjualan konsinyasi, dan Penjualan grosir. Berikut penjelasannya:

- a. Penjualan tunai

Penjualan tunai adalah Penjualan yang pembayarannya dilakukan secara tunai atau kontan (Idris, 2021). Pada Penjualan tunai, pembeli langsung membayar produk yang jual Penjual dengan harga yang disepakati. Penjualan tunai menjadi bentuk Penjualan dalam penelitian ini dimana pembeli akan membayarkannya langsung kepada Penjual dengan harga yang telah disepakati.

- b. Penjualan kredit

Penjualan kredit adalah Penjualan yang pembayarannya tidak dilakukan secara tunai (Idris, 2021). Pembeli akan membayar dengan cara angsuran atau kredit sesuai dengan kesepakatan dengan Penjual. Pada penelitian ini tidak menggunakan Penjualan kredit dalam membayar produk.

c. Penjualan tender

Penjualan tender adalah Penjualan yang dilakukan secara lelang. Di mana Penjual akan mempresentasikan produknya kepada pembeli, pembeli kemudian akan memilih produk terbaik yang mereka inginkan (Idris, 2021). Penjualan tender tidak menjadi bentuk Penjualan dalam penelitian ini dikarenakan Penjualan produk tidak dilakukan secara lelang.

d. Penjualan ekspor

Penjualan ekspor adalah Penjualan yang dilakukan antar negara. Produk yang diproduksi negara tertentu akan dijual (diekspor) ke negara lain untuk mendapatkan keuntungan (Idris, 2021). Pada penelitian ini tidak melakukan Penjualan ekspor tetapi Penjualan produk berada didalam negeri.

e. Penjualan konsinyasi

Penjualan konsinyasi adalah Penjualan yang dilakukan kepada perantara. Contoh paling umum Penjualan konsinyasi adalah reseller. Di mana Penjual akan menjual barang kepada reseller, dan reseller akan menjualnya lagi kepada orang lain. ketika produk tidak habis dijual, maka barang bisa dikembalikan pada Penjual pertama (Idris, 2021). Pada penelitian ini produk yang dijual berasal langsung dari produsen, tidak melalui perantara.

f. Penjualan grosir

Penjualan grosir adalah Penjualan dalam volume besar. Penjualan grosir menjual produk dengan harga yang lebih murah namun dalam jumlah banyak untuk kembali dijual (Idris, 2021). Dalam penelitian ini tidak melakukan Penjualan secara grosir.

Tujuan Penjualan dalam perusahaan adalah untuk mencapai volume Penjualan tertentu, memperoleh laba, dan menunjang pertumbuhan serta perkembangan perusahaan. Sehingga Penjualan bertujuan untuk menjual produk secara efektif, pada posisi tetap, bahkan meningkat, dan juga menghasilkan keuntungan. Di mana keuntungan tersebut akan digunakan untuk mengembangkan perusahaan menjadi lebih besar dan stabil.

2.2.4 Pemasaran

Pemasaran adalah aktivitas mempercepat perpindahan barang maupun jasa yang dijual dari produsen atau distributor hingga ke tangan konsumen (Idris, 2021). Pemasaran juga merupakan sebuah strategi bisnis yang mengacu pada aktivitas yang dilakukan perusahaan untuk mempromosikan Penjualan suatu produk. Tujuan dari pemasaran adalah tentunya untuk memaksimalkan keuntungan dengan membuat strategi Penjualan. Dalam pemasaran, perusahaan membutuhkan strategi yang harus disesuaikan dengan target yang disasar. Sehingga seringkali pemasaran banyak melibatkan artis, olahragawan, figur publik, hingga selebrgam. Contohnya dalam strategi pemasaran produk kosmetik, tentulah yang digunakan adalah artis wanita yang selama ini populer dengan aspek perawatan dan kecantikan. Pemasaran tak hanya soal aktivitas beriklan, namun juga mencakup berbagai aspek. Bahkan penentuan desain kemasan pun masuk dalam strategi pemasaran. Semakin menarik kemasan, semakin menarik pula calon konsumen untuk membelinya.

Ada banyak sekali strategi pemasaran yang dikenal hingga saat ini berikut beberapa di antaranya:

- a. Iklan

Iklan adalah jenis pemasaran paling umum dan mudah ditemui. Iklan adalah strategi pemasaran berbayar yang menggunakan beragam media yang meliputi televisi, surat kabar, radio, media daring, hingga *billboard* (Idris, 2021). Pada penelitian ini tidak melakukan pemasaran dengan iklan.

b. Internet

Pemasaran via internet mulai populer sejak era digital. Biasanya pemasaran ini dilakukan dengan beriklan melalui website, mesin pencari, media online, hingga media sosial (Idris, 2021). Pada penelitian ini menggunakan media internet untuk pemasaran produk yaitu website.

c. MLM

MLM atau *multi level marketing* adalah pemasaran dengan menjangkau konsumen dari Penjualan langsung dan biasanya menerapkan sistem berjenjang (Idris, 2021). Seringkali metode MLM tak hanya menjual namun juga seringkali merekrut orang menjadi tenaga pemasarnya. Pada penelitian ini tidak melakukan pemasaran dengan metode MLM.

d. *Branding*

Branding pemasaran adalah metode untuk mengenalkan produk lebih dikenal masyarakat sebagai calon konsumen (Idris, 2021). *Branding* bisa dilakukan dengan beragam cara, seperti menjaga kualitas, pelayanan pasca Penjualan, hingga logo. Pada penelitian ini tidak melakukan branding dalam pemasaran produk.

Menurut (Idris, 2021), berikut ini beberapa fungsi dari pemasaran:

a. Peningkatan Penjualan

Fungsi pemasaran adalah tentunya meningkatkan Penjualan yang pada akhirnya meningkatkan laba. Semakin banyak aktivitas pemasaran, maka semakin besar peluang produk atau jasa terjual.

b. Pengenalan produk

Pengenalan produk jadi salah satu fungsi dari pemasaran. Dengan adanya aktivitas pemasaran, maka produk akan lebih mudah dikenal oleh konsumen.

c. Riset

Fungsi pemasaran adalah satunya terkait riset. Dengan pemasaran, maka bisa diketahui secara detail target konsumen, sebaran konsumen, dan produk yang disukai konsumen.

d. Kepuasan konsumen

Strategi pemasaran adalah tidak cukup dengan membuat produk berkualitas saja, tetapi juga terkait kepuasan konsumen. Karena kepuasan konsumen akan menjadikan Penjualan produk di masa mendatang tumbuh.

e. Kompetisi

Tujuan pemasaran adalah tentunya sebagai kompetisi. Untuk memenangkan persaingan, maka perusahaan harus melakukan strategi pemasaran dengan menonjolkan keunggulannya dibandingkan produk pesaing.

Terdapat komponen dalam pemasaran. Neil Borden mempopulerkan gagasan bauran pemasaran dan konsep Empat P pada 1950-an. Apa saja komponen empat P dalam *marketing*:

a. *Product*

Produk mengacu pada barang atau barang yang direncanakan untuk ditawarkan kepada pelanggan. Produk harus berusaha untuk memenuhi ketidakhadiran di pasar, atau memenuhi permintaan konsumen untuk jumlah yang lebih besar dari produk yang sudah tersedia. Sebelum mereka dapat menyiapkan *marketing* yang tepat, tenaga *marketing* perlu memahami produk apa yang dijual, apa yang menonjol dari pesaingnya, apakah produk tersebut juga dapat

dipasangkan dengan produk sekunder atau lini produk lain, dan apakah ada produk pengganti di pasar. Pada penelitian ini produk yang dijual ialah produk olahan makanan.

b. Price

Price atau harga mengacu pada berapa banyak perusahaan akan menjual produk. Ketika menetapkan harga, perusahaan harus mempertimbangkan harga biaya per unit, biaya pemasaran, dan biaya distribusi. Perusahaan juga harus mempertimbangkan harga produk pesaing di pasar dan apakah titik harga yang diusulkan cukup untuk mewakili alternatif yang masuk akal bagi konsumen. Pada penelitian ini produk yang dijual akan diperhitungkan harga yang sesuai kemudian ditambahkan dengan biaya pengiriman produk kepada pelanggan.

c. Place

Place atau tempat mengacu pada distribusi produk. Pertimbangan utama termasuk apakah perusahaan akan menjual produk melalui etalase fisik, *online*, atau melalui kedua saluran distribusi. Ketika dijual di etalase, penempatan produk fisik seperti apa yang didapatnya? Jika dijual online, penempatan produk digital seperti apa yang didapat. Pada penelitian ini tempat Penjualan ialah UMKM desa sindangsari, lampung selatan dengan tempat Penjualan produk ialah media internet yaitu website.

d. Promotion

Marketing adalah promosi. Promosi adalah kampanye komunikasi pemasaran terpadu. Promosi mencakup berbagai kegiatan seperti periklanan, Penjualan, promosi Penjualan, hubungan masyarakat, pemasaran langsung, sponsorship, dan pemasaran gerilya. Promosi bervariasi tergantung pada tahap siklus produk di mana produk tersebut berada. Tenaga *marketing* adalah diharapkan bisa memahami bahwa konsumen mengaitkan harga dan distribusi produk dengan kualitasnya, dan mereka mempertimbangkan hal ini ketika

merancang strategi pemasaran secara keseluruhan. Pada penelitian ini promosi dilakukan secara online yaitu media sosial dan website.

2.2.5 Website

Web atau Website adalah kumpulan halaman yang berisi informasi tertentu dan dapat diakses dengan mudah oleh siapapun, kapanpun, dan dimanapun melalui internet (Wijayanti, 2021). Website pertama kali dibuat oleh Tim Berners-Lee pada akhir 1980-an dalam project *World Wide Web* (W3). Terdapat 5 unsur vital yang terdapat pada Website, yaitu:

1. *Domain*, yaitu alamat sebuah Website.
2. *Hosting*, yaitu *server* tempat dimana semua file website disimpan serta dapat diakses dan dikelola melalui internet.
3. Konten, yaitu informasi atau isi dari Website tersebut.
4. Bahasa Pemrograman (Kode), yaitu bahasa pemrograman yang membuat atau membangun website tersebut.
5. *Design Tampilan*, yaitu riasan yang mempercantik atau memperindah isi dari konten website tersebut.
6. Secara umum, ada dua jenis website berdasarkan sifatnya, yaitu website statis dan dinamis. Jika informasi yang disediakan tetap atau tidak berubah disebut website statis. Sebaliknya, jika informasi dalam website selalu berubah-ubah disebut website dinamis.

2.2.6 PHP

PHP (*PHP: Hypertext Preprocessor*) adalah sebuah bahasa pemrograman *server side scripting* yang bersifat *open source* (Awwaabiin, 2021). Sebagai sebuah *scripting language*, PHP menjalankan instruksi pemrograman saat proses *runtime*. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses. PHP merupakan bahasa pemrograman *server-side*, maka *script* dari

PHP nantinya akan diproses di *server*. Jenis *server* yang sering digunakan bersama dengan PHP antara lain *Apache*, *Nginx*, dan *LiteSpeed*. Selain itu, PHP juga merupakan bahasa pemrograman yang bersifat *open source*. Pengguna bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan mereka.

2.2.7 Codeigniter

Codeigniter adalah sebuah *framework* khusus web dan application development yang hadir sebagai platform open-source (BsourceCode, 2017). Aplikasi ini diciptakan untuk para *developer* yang ingin membangun situs web menggunakan bahasa pemrograman php. Tujuan *Codeigniter* adalah agar para *developer* dapat mengembangkan proyek mereka dengan jauh lebih cepat.. Tak hanya itu, *Codeigniter* juga menyediakan ui serta struktur yang sederhana dan logis untuk mengakses *built-in-library*. Intinya, *Codeigniter* memungkinkan *developer* untuk tetap fokus pada proyek mereka dengan mengurangi jumlah kode yang diperlukan untuk tugas-tugas tertentu. Fitur-fitur yang terdapat dalam aplikasi *Codeigniter mvc*, *built-in-library*, *extremely light weight*, *source code* yang kecil.

2.2.8 Visual Studio Code

Visual Studio Code merupakan sebuah teks editor multiplatform buatan Microsoft (Tekno, 2022). Visual Studio Code pertama kali diperkenalkan di tanggal 29 April 2015 oleh Microsoft di konferensi Build 2015. VS Code tersedia untuk Windows, Linux, serta MAC OS. Teks Editor ini mendukung banyak bahasa pemrograman seperti JavaScript, TypeScript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang di Visual Studio Code seperti C++, C#, Python, Go, PHP. VS Code bersifat *open source*. VS Code menyediakan Intellisense, Git Integration, Debugging, dan fitur Ekstensi. Fitur-fiturnya akan terus bertambah seiring dengan

perkembangan versi Visual Studio Code. Perkembangan versi Visual Studio Code dilakukan setiap bulan.

2.2.9 Xampp

Xampp adalah suatu platform web *server* bersifat *open source* yang dikembangkan oleh *Apache* (Rosyida, 2022). Xampp biasa digunakan untuk membuat database. Selain itu, xampp juga digunakan untuk menguji program yang telah dibuat di *server* lokal sebelum nantinya di-upload di *server* utama. Tanpa sebuah web *server* atau xampp, *developer* atau programmer tidak bisa mengakses website yang telah dibuatnya di lokal (laptop). Hal ini karena sebuah website akan bisa berjalan ketika terhubung dengan suatu web *server*. Xampp merupakan gabungan dari beberapa komponen. XAMPP adalah singkatan dari X (*Cross platform*), A (*Apache*), M (MySQL/MariaDB), P (PHP), dan P (Perl) yang adalah program-program yang tersedia di *software* ini.

a. X = *Cross platform*

Kode penanda dari *software Cross platform* yang berarti dapat dijalankan di banyak sistem operasi seperti Windows, Linus, Mac OS, dan Solaris.

b. A = *Apache*

Aplikasi web *server* gratis dan bisa dikembangkan oleh banyak orang (*open source*) untuk menciptakan halaman website yang benar berdasarkan kode program PHP yang ditulis oleh pengembang web *developer*.

c. M = MySQL / MariaDB

Aplikasi database *server* yang menerapkan bahasa pemrograman SQL (*Structured Query Language*) yang berfungsi untuk mengelola dan membuat sistem database yang terstruktur dan sistematis seperti mengolah, mengedit, dan menghapus daftar melalui database.

d. P = PHP

Bahasa pemrograman khusus berbasis web untuk kebutuhan pada sisi *server* (back end), sehingga bisa digunakan untuk membuat halaman website menjadi lebih dinamis dengan menerapkan *server-side scripting*.

e. P = Perl

Bahasa pemrograman untuk memenuhi berbagai kebutuhan (*Cross platform*) yang bisa berjalan di banyak sistem operasi sehingga sangat fleksibel, misalnya sebagai penunjuk eksistensi dari PHP. Biasanya digunakan untuk membuat website dinamis seperti CMS (*Content Management System*) WordPress.

Lima komponen penting dalam XAMPP adalah sebagai berikut:

a. Control panel

Layanan yang memberikan akses lebih leluasa dalam mengatur database, mengunggah file, mengelola fitur juga melakukan konfigurasi terkait proyek website.

b. *Htdocs*

Merupakan nama sebuah folder bagian dari XAMPP yang berfungsi untuk menyimpan berbagai file dan dokumen yang akan ditampilkan di website. Kapasitas penyimpanan dalam *Htdocs* bergantung pada storage *harddisk* yang ada di komputer.

c. *PhpMyAdmin*

Program untuk mengolah database website yang sedang dikembangkan. Kamu bisa mengakses *PhpMyAdmin* dengan berkunjung ke situs <http://localhost/phpMyAdmin>.

d. *Config*

Menu yang digunakan untuk melakukan beberapa konfigurasi dasar XAMPP.

e. *Netstat*

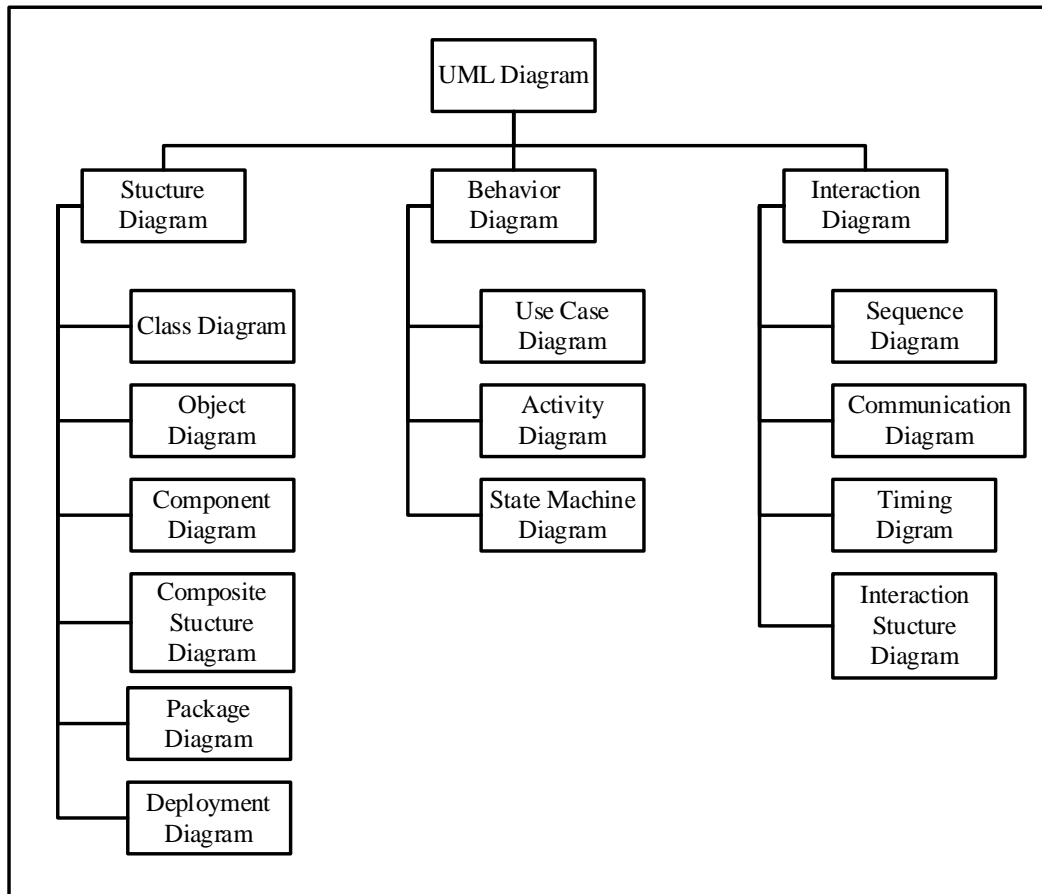
Menu yang digunakan untuk memastikan apakah ada aplikasi lain yang menggunakan port XAMPP pada perangkat yang digunakan.

2.2.10 Unified Modelling Language (UML)

Menurut (Rajendra, 2022), UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software*. UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk *factor-faktor scalability, robustness, security*, dan sebagainya. Adapun tujuan dan fungsi perlu adanya UML yaitu sebagai berikut:

1. Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
1. Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.
2. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
3. Tidak hanya menggambarkan model sistem *software* saja, namun dapat memodelkan sistem berorientasi objek.
4. Mempermudah pengguna untuk membaca suatu sistem.
5. Berguna sebagai *blueprint*, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa *coding* suatu program.

Dalam penelitian ini UML yang digunakan penulis untuk perancangan sistem adalah *Use case* diagram, *Activity* diagram, dan *Class* diagram. UML diagram dapat dilihat pada gambar 2.1.



Gambar 2.1 UML Diagram

2.2.10.1 *Structure* Diagram

Structure diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan (AS dan Shalahuddin, 2016). *Structure* diagram terdiri dari *Class* diagram, *object* diagram, *component* diagram, *composite structure* diagram, *package* diagram, dan *deployment* diagram.

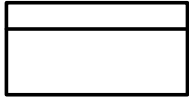





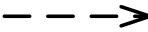
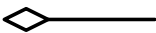
1. *Class* Diagram

Class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem *Class*, atributnya, metode, dan hubungan antar objek (AS dan Shalahuddin, 2016). *Class* diagram disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen. Berbagai komponen tersebut dapat mewakili *Class* yang akan diprogram, objek utama, atau interaksi antara *Class* dan objek. *Class* sendiri merupakan istilah yang mendeskripsikan sekelompok objek yang semuanya memiliki peran serupa dalam sistem. Sekelompok objek ini terdiri atas fitur struktural yang mendefinisikan apa yang diketahui *Class* dan fitur operasional yang mendefinisikan apa yang bisa dilakukan oleh *Class*. Fungsi utama dari *Class* diagram adalah menggambarkan struktur sebuah sistem pemrograman. Meski demikian, terdapat beberapa fungsi lainnya dari *Class* diagram.

1. Menunjukkan struktur statis pengklasifikasi dalam suatu sistem.
2. Memberikan notasi dasar untuk diagram struktur lain yang ditentukan oleh UML.
3. Dapat digunakan business analyst untuk membuat model sistem dari perspektif bisnis.

Bagian bagian dalam *Class* Diagram dapat dilihat pada tabel 2.2.

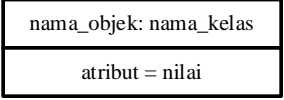

Tabel 2.2 *Class* Diagram

Gambar	Nama	Keterangan
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendant</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
	<i>Association</i>	Penghubung antara objek satu dengan objek yang lainnya
	<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah biasanya juga disertai dengan multiplicity.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri
	<i>Aggregation</i>	Relasi antar <i>class</i> dengan makna semua bagian (<i>whole-part</i>)

2. Object Diagram

Object diagram merupakan penggambaran stuktur sistem dari segi penamaan objek dan jalannya objek dalam sistem (AS dan Shalahuddin, 2016). Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggung jawabkan. Diagram objek juga berfungsi untuk mendefinisikan contoh nilai atau isi dari atribut tiap kelas. Simbol-simbol *object* diagram serta penjelasannya dapat dilihat pada tabel 2.3.

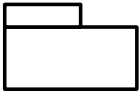
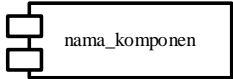



Tabel 2.3 Object Diagram

Simbol	Keterangan
<p>Objek</p> 	Objek dari kelas yang berjalan saat sistem dijalankan
<p>Link</p> 	Relasi antar objek

3. Component Diagram

Component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem (AS dan Shalahuddin, 2016). *Component* Diagram fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Simbol-simbol *component* diagram serta penjelasannya dapat dilihat pada tabel 2.4.

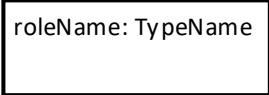


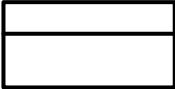
Tabel 2.4 Component Diagram

Simbol	Keterangan
<p>Package</p> 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen
<p>Komponen</p> 	Komponen Sistem
<p>Kebergantungan / <i>dependency</i></p> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
<p>Link</p> 	Relasi antar komponen
<p>Anarmuka/ <i>interface</i></p> 	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen

4. Composite Structure Diagram

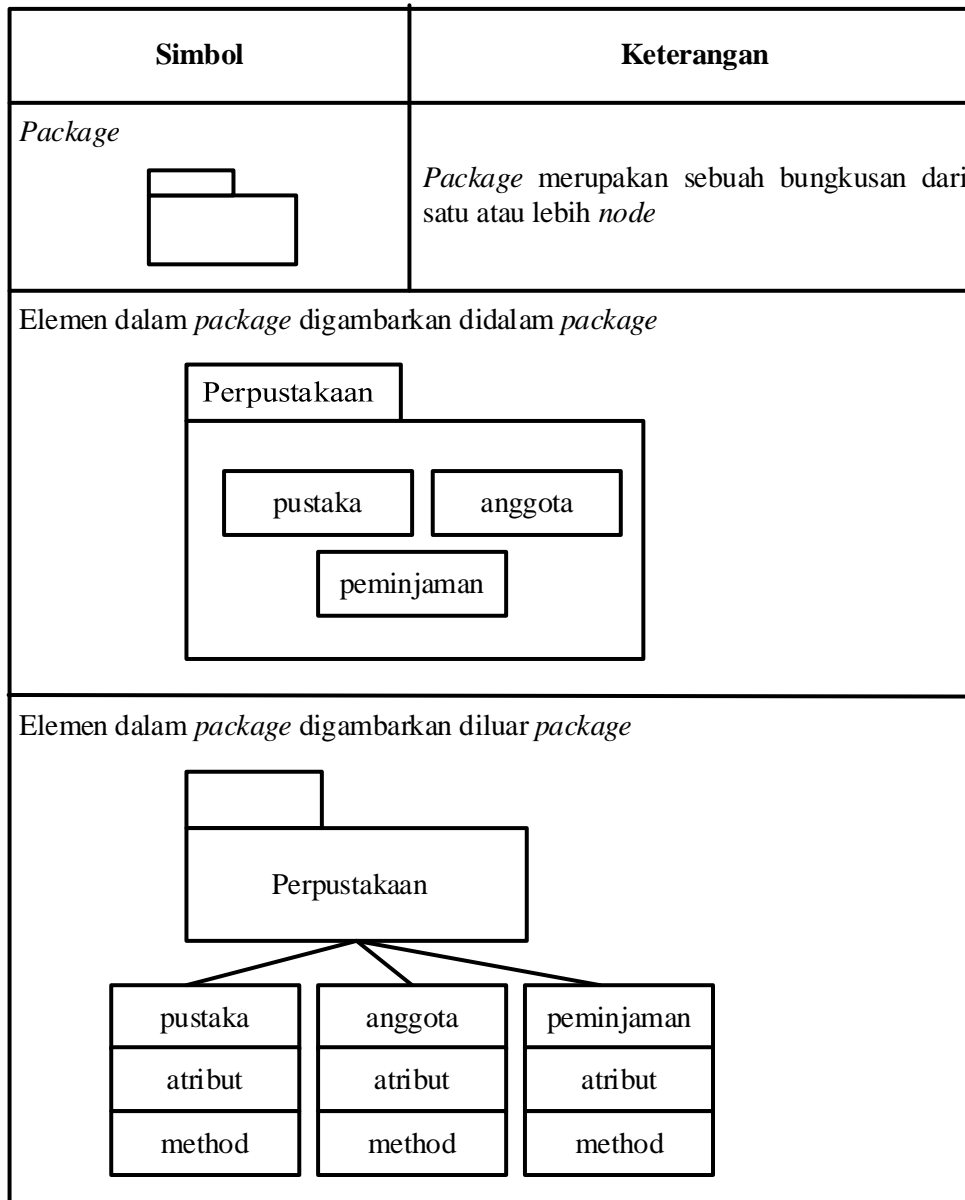
Composite Structure Diagram digunakan untuk menggambarkan stuktur dari bagian bagian yang saling terhubung maupun mendeskripsikan stuktur pada saat berjalan (*runtime*) dari *instance* yang saling terhubung (AS dan Shalahuddin, 2016). Dapat menggambarkan stuktur didalam kelas atau kolaborasi. Contoh penggunaan diagram ini misalnya untuk menggambarkan deskripsi dari setiap bagian mesin yang saling terkait untuk menjalankan fungsi mesin tersebut, menggambarkan aliran *router* pada jaringan computer, dan lain lain. *Composite structure* diagram dapat dilihat pada tabel 2.5.

Tabel 2.5 *Composite Structure Diagram*

Simbol	Keterangan
<p><i>Property</i></p>  <p>roleName: TypeName</p>	<p><i>Property</i> adalah satu set dari suatu <i>instance</i></p> <p>roleName: peran / nama /identitas dari property (opsional)</p> <p>TypeName: tipe kelas dari property (harus ada)</p>
<p><i>Connector</i></p>  <p>namaConnector: TypeConnector</p>	<p><i>Connector</i> adalah cara komunikasi dari 2 buah <i>instance</i></p> <p>connName: nama <i>connector</i> (opsional)</p> <p>ConnType: tipe <i>connector</i> (opsional)</p>
<p><i>Port</i></p>  <p>portName: EntityName(n)</p>	<p><i>Port</i> adalah cara yang digunakan dalam diagram <i>composite structure</i> tanpa menampilkan detail internal dari suatu sistem</p>
<p><i>Class</i></p> 	<p>Kelas; Jika yang akan dijabarkan srukturnya adalah sebuah kelas</p>

5. Package Diagram

Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML (AS dan Shalahuddin, 2016). Hampir semua diagram dalam UML dapat dikelompokkan menggunakan *package* diagram. *Package* diagram dapat dilihat pada gambar 2.2.



Gambar 2.2 *Package* Diagram


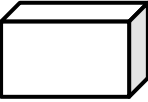


6. Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi (AS dan Shalahuddin, 2016). *Deployment* diagram juga dapat digunakan untuk memodelkan hal-hal berikut:

- a. Sistem tambahan
- b. Sistem *client/server*
- c. Sistem terdistribusi murni
- d. Rekayasa ulang aplikasi

Simbol-simbol *deployment* diagram serta penjelasannya dapat dilihat pada tabel 2.6.

Tabel 2.6 *Deployment* Diagram

Simbol	Keterangan
<i>Package</i> 	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih <i>node</i>
<i>Node</i> 	Biasanya mengacu pada hardware, software yang tidak dibuat sendiri, jika didalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan / <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
<i>Link</i> 	Relasi antar <i>node</i>




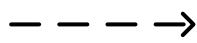

2.2.10.2 Behavior Diagram

Behavior diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem (AS dan Shalahuddin, 2016). *Behavior* diagram terdiri dari *Use case* diagram, *Activity* diagram, dan *State machine* diagram.

1. Use case Diagram

Use case diagram merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem (AS dan Shalahuddin, 2016). *Use case* diagram bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* diagram juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan bisa juga mempresentasikan sebuah interaksi aktor dengan sistem. Komponen tersebut kemudian menjelaskan komunikasi antara aktor, dengan sistem yang ada. Dengan demikian, *use case* dapat dipresentasikan dengan urutan yang sederhana, dan akan mudah dipahami oleh para konsumen. Manfaat dari *use case* sendiri adalah untuk memudahkan komunikasi dengan menggunakan *domain expert* dan juga *end User*, memberikan kepastian pemahaman yang pas tentang requirement atau juga kebutuhan sebuah sistem. Bagian bagian pada *Use case* diagram dapat dilihat pada tabel 2.7.

Tabel 2.7 Use case Diagram

Simbol	Keterangan
	<p><u>Aktor</u> Mewakili peran Orang, Sistem yang lain, atau alat yang lain ketika berkomunikasi dengan <i>Use Case</i></p>
	<p><u>Use Case</u> Abstraksi dan Interaksi antara sistem dan aktor</p>
	<p><u>Association</u> Abstraksi dari penghubung antara aktor dengan <i>Use Case</i></p>
	<p><u>Generalisasi</u> Menunjukkan Spesialisasi Aktor untuk dapat berpartisipasi dengan <i>Use Case</i></p>
<p><<include>>→ </p>	<p><u>Include/Uses</u> Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya</p>
<p><<extend>>→</p>	<p><u>Extend</u> Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi</p>

2. Activity Diagram






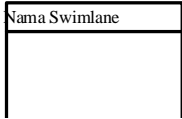
Activity Diagram adalah penggambaran workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (AS dan Shalahuddin, 2016). *Activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.

Activity diagram juga banyak digunakan untuk mendefinisikan hal hal berikut:

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

Simbol-simbol yang terdapat pada *Activity* diagram dapat dilihat pada tabel 2.8.




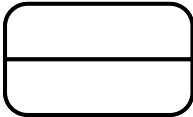
Tabel 2.8 *Activity* Diagram

Simbol	Keterangan
<u>Status Awal</u> 	Status Awal dari sebuah aktivitas sistem
<u>Aktivitas</u> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
<u>Percabangan/ Decision</u> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
<u>Penggabungan/ Join</u> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
<u>Status Akhir</u> 	Status akhir yang dilakukan oleh sistem
<u>Swimlane</u> 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi

3. *State machine* Diagram

State machine diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek (AS dan Shalahuddin, 2016). Jika diagram sekuen digunakan untuk interaksi antar objek maka *State machine* diagram digunakan untuk interaksi didalam sebuah objek. Perubahan tersebut digambarkan dalam suatu graf berarah. *State machine* diagram cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem. Simbol-simbol *State machine* diagram serta penjelasannya dapat dilihat pada tabel 2.9.

Tabel 2.9 *State machine* Diagram

Simbol	Keterangan
<i>Start/Status Awal (Initial State)</i> 	<i>Start</i> atau <i>initial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup
<i>End / status akhir (Final State)</i> 	<i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem
<i>Event</i> 	<i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin
<i>State</i> 	<i>State</i> atau status adalah keadaan sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada event tertentu yang memicu perubahan tersebut.

2.2.10.3 Interaction Diagram


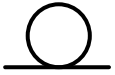
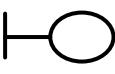



Interaction diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem (AS dan Shalahuddin, 2016). Macam macam diagram pada *Interaction* diagram yaitu *Sequence* diagram, *Communication* diagram, *Timing* diagram, *Interaction overview* diagram.

1. *Sequence* Diagram

Sequence Diagram merupakan salah satu yang menjelaskan bagaimana suatu operasi itu dilakukan; *overview* (pesan) apa yang dikirim dan kapan pelaksanaannya (AS dan Shalahuddin, 2016). Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. *Sequence* Diagram menampilkan interaksi antar objek dalam dua dimensi. Dimensi vertikal adalah poros waktu, dimana waktu berjalan ke arah bawah. Sedangkan dimensi horizontal

merepresentasikan objek-objek individual. Tiap objek (termasuk *actor*) tersebut mempunyai waktu aktif yang direpresentasikan dengan kolom vertikal yang disebut dengan *lifeline*. Pesan (*overview*) direpresentasikan sebagai panah dari satu *lifeline* ke *lifeline* yang lain. *Overview* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *overview* akan dipetakan menjadi operasi/metoda dari *Class*. *Sequence Diagram* ini *Sequence Diagram* mendeskripsikan bagaimana entitas dalam sistem berinteraksi, termasuk pesan yang digunakan saat interaksi. Semua pesan dideskripsikan dalam urutan dari eksekusi. Bagian bagian dalam *Sequence Diagram* dijelaskan dalam gambar 2.10.

Tabel 2.10 *Sequence Diagram*

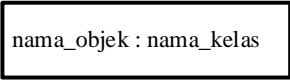


Gambar	Nama	Keterangan
	<i>Aktor</i>	Menggambarkan Orang yang sedang berinteraksi dengan Sistem
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari form
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
	<i>A Focus of Control & A Lifeline</i>	Menggambarkan tempat mulai dan berakhirnya sebuah message
	<i>A Message</i>	Menggambarkan pengiriman sebuah pesan

2. Communication Diagram

Communication diagram menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan (AS dan Shalahuddin, 2016). Diagram ini merepresentasikan informasi yang diperoleh dari *Class* diagram, *Sequence* diagram, dan *use case* diagram untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem. *Communication* diagram mengelompokkan *overview* pada kumpulan diagram *Sequence* menjadi sebuah diagram. Dalam diagram komunikasi yang dituliskan adalah operasi/metode yang dijalankan antara objek yang satu dan objek yang lainnya secara keseluruhan, oleh karena itu dapat diambil dari jalannya interaksi pada semua diagram sekuen. Penomoran metode dapat dilakukan

berdasarkan urutan dijalankannya metode/operasi diantara objek yang satu dengan objek lainnya atau obejk itu sendiri. *Communication* diagram dapat dilihat pada tabel 2.11.

Tabel 2.11 *Communication* Diagram

Simbol	Keterangan
Objek 	Objek yang melakukan interaksi pesan
Link 	Relasi antar-objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri
Arah pesan/Stimulus 	Arah pesan yang terjadi, jika pada suatu link ada dua arah pesan yang berbesa maka arah juga digambarkan dua arah pada dua sisi link

3. *Timing* Diagram

Timing Diagram merupakan diagram yang fokus pada penggambaran terkait batasan waktu (AS dan Shalahuddin, 2016). Diagram ini digunakan untuk menggambarkan tingkah laku sistem dalam periode waktu tertentu. *Timing* diagram biasanya digunakan untuk mendeskripsikan operasi alat digital karena penggambaran secara visual akan lebih mudah dipahami daripada dengan kata-kata.

4. *Interaction Overview* Diagram

Interaction Overview Diagram mirip dengan *Activity* diagram yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas (AS dan Shalahuddin, 2016). Diagram ini adalah bentuk aktivitas diagram yang setiap titik merepresentasikan diagram interaksi. Interaksi diagram dapat meliputi *Sequence* diagram, *Communication* diagram, *Interaction overview* diagram, dan *Timing* diagram.

2.2.11 Prototype

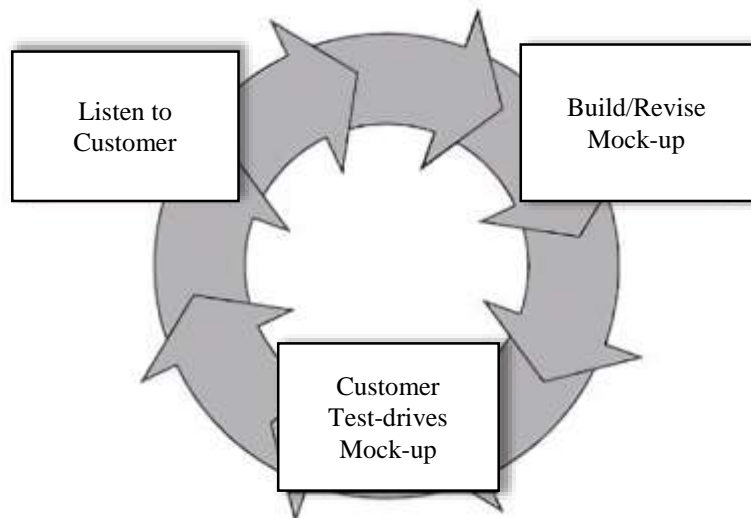
Prototype merupakan salah satu metode pengembangan perangkat lunak yang menggunakan pendekatan untuk membuat rancangan dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh calon pengguna/klien (Maya, 2020). Dengan metode prototyping ini pengembang dan klien dapat saling berinteraksi selama proses pembuatan *prototype* sistem. Terkadang sering terjadi, klien hanya mendefinisikan secara umum apa yang dikehendaki tanpa menyebutkan proses masukan (*input*) dan keluaran (*output*) dari sistem yang akan dibuat. Untuk mengatasi ketidakselarasan tersebut maka harus dibutuhkan kerjasama yang baik di antara keduanya, sehingga pengembang akan mengetahui dengan benar apa yang dibutuhkan klien. Dengan demikian nantinya akan menghasilkan sebuah rancangan sistem yang interaktif sesuai dengan kebutuhan. Metode *prototype* dapat dilihat pada gambar 2.3.

Gambar 2.3 Metode *Prototype*

Terdapat 3 siklus dalam metode *prototype* yaitu sebagai berikut:

1. *Listen to Customer* (Mendengarkan Pelanggan)

Pada tahap ini merupakan identifikasi kebutuhan *User*, proses ini dilakukan agar penulis dapat memperoleh informasi mengenai permasalahan yang terjadi oleh klien. Data yang



diperoleh dari permasalahan tersebut yang nantinya menjadi acuan untuk dilakukan proses pencarian solusi dan pengembangan pada tahap selanjutnya.

2. *Build and Revise Mock-up* (Membangun dan Memperbaiki *Prototype*)

Setelah kebutuhan sistem terkumpul, maka akan dilakukan proses perancangan *prototype* pada sistem yang diusulkan oleh *User*, yang mana tahap-tahapannya sebagai berikut:

- a. Perancangan proses-proses yang akan terjadi di dalam sistem, seperti, *input* (masukan), *output* (keluaran) dari sistem yang telah diusulkan.
- b. Perancangan UML (*Unified Modelling Language*), hal ini dilakukan untuk menspesifikasikan sistem tentang apa yang diperlukan dan bagaimana sistem tersebut direalisasikan.
- c. Perancangan *Interface* (antarmuka) dan fitur yang dibutuhkan oleh klien *User*.

3. *Customer Test Drives Mock-up* (Pengujian *Prototype*)

Pada tahap ini akan dilakukan pengujian terhadap *prototype* sistem yang telah dibuat, serta mengevaluasi apakah *prototype* sistem yang sudah dibuat sudah sesuai dengan yang diharapkan. Apabila dari hasil pengujian *prototype* tersebut belum memenuhi kebutuhan klien *User*, maka pengembang akan melakukan proses perbaikan ulang *prototype* sampai *prototype* tersebut menjadi sistem yang final dan benar-benar diterima atau sesuai dengan keinginan *User*.

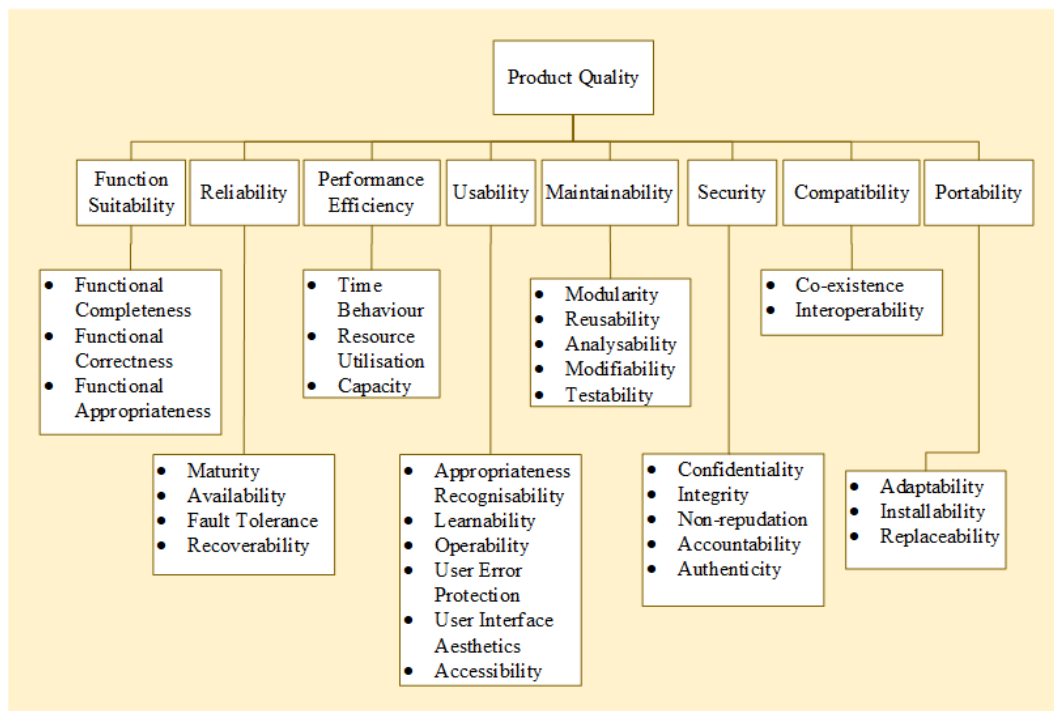
2.2.12 ISO/IEC 25010

Menurut (Mulyawan *et al.*, 2021), ISO/IEC 25010 merupakan pedoman yang digunakan untuk melakukan evaluasi perangkat lunak yang dikeluarkan oleh *Canadian Standards Association* pada tahun 2011. ISO/IEC 25010 adalah model baru dari seri ISO/IEC 250n yang merupakan pengembangan dari versi ISO/IEC 9126. Model ISO/IEC 25010 memiliki karakteristik

kualitas yang dapat dipertimbangkan ketika melakukan evaluasi perangkat lunak. Terdapat 2 model yang dapat digunakan untuk mengukur kualitas sistem yaitu *quality in use model*, dan *software product quality model*.

Gambar 2.4 ISO/IEC 25010

ISO/IEC 25010 memiliki 8 karakteristik sebagai berikut:



1. *Function Suitability*

Sejauh mana perangkat lunak mampu menyediakan fungsi yang memenuhi kebutuhan yang dapat digunakan dalam kondisi tertentu. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu:

- a. *Function completeness*, yaitu sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik.
- b. *Function Correctness*, yaitu sejauh mana produk atau sistem menyediakan hasil yang benar sesuai kebutuhan.
- c. *Function Appropriateness*, yaitu sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

2. *Compatibility*

Sejauh mana sebuah produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen dan/atau menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan *environment* perangkat lunak yang sama.

Karakteristik ini dibagi menjadi 2 karakteristik yaitu:

- a. *Co-existence*, yaitu sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien sementara berbagi sumber daya dengan produk atau sistem yang lain tanpa merugikan produk atau sistem tersebut.
- b. *Interoperability*, yaitu sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.

3. *Usability*

Sejauh mana sebuah produk atau sistem dapat digunakan oleh *User* tertentu untuk mencapai tujuan dengan efektif, efisien, dan kepuasan tertentu dalam konteks penggunaan.

Karakteristik ini terbagi menjadi beberapa karakteristik, yaitu:

- a. *Appropriateness Recognizability*, yaitu sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.

- b. *Learnability*, yaitu sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari resiko, dan kepuasan dalam konteks tertentu.
- c. *Operability*, yaitu sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
- d. *User Error Protection*, yaitu sejauh mana produk atau sistem melindungi pengguna terhadap membuat kesalahan.
- e. *User Interface Aesthetics*, yaitu sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.
- f. *Accessibility*, yaitu sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.

4. *Reliability*

Sejauh mana sebuah sistem, produk atau komponen dapat menjalankan fungsi tertentu dalam kondisi tertentu selama jangka waktu yang ditentukan. Karakteristik ini terbagi menjadi beberapa subkarakteristik, yaitu:

- a. *Maturity*, yaitu sejauh mana produk atau sistem mampu memenuhi kebutuhan secara handal di bawah keadaan normal.
- b. *Availability*, yaitu sejauh mana produk atau sistem siap beroperasi dan dapat diakses saat perlu digunakan.
- c. *Fault Tolerance*, yaitu sejauh mana produk atau sistem tetap berjalan sebagaimana yang dimaksud meskipun terjadi kesalahan pada perangkat keras atau perangkat lunak.

- d. *Recoverability*, yaitu sejauh mana produk atau sistem mampu dapat memulihkan data yang terkena dampak secara langsung dan menata ulang kondisi sistem seperti yang diinginkan ketika terjadi gangguan.

5. *Security*

Sejauh mana sebuah produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan level otorisasi yang dimiliki.

Karakteristik ini terbagi menjadi beberapa karakteristik, yaitu:

- a. *Confidentiality*, yaitu sejauh mana produk atau perangkat lunak memastikan data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
- b. *Integrity*, yaitu sejauh mana produk atau perangkat lunak mampu mencegah akses yang tidak sah untuk memodifikasi data.
- c. *Non-repudiation*, yaitu sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan terhadap peristiwa atau tindakan tersebut.
- d. *Accountability*, yaitu sejauh mana tindakan dari suatu entitas dapat ditelusuri secara unik untuk entitas.
- e. *Authenticity*, yaitu sejauh mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim.

6. *Portability*

Sejauh mana keefektifan dan efisiensi sebuah sistem, produk atau komponen dapat dipindahkan dari suatu perangkat keras, perangkat lunak atau digunakan pada lingkungan yang berbeda. Karakteristik ini dibagi menjadi beberapa karakteristik, yaitu:

- a. *Adaptability*, yaitu sejauh mana produk atau sistem dapat secara efektif dan efisien disesuaikan pada perangkat lunak, perangkat keras, dan lingkungan yang berbeda

- b. *Installability*, yaitu sejauh mana produk atau sistem dapat berhasil dipasang atau dihapus dalam lingkungan tertentu.
- c. *Replaceability*, sejauh mana produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama pada lingkungan yang sama.

7. *Performance Efficiency*

Kinerja relative terhadap sumber daya yang digunakan dalam kondisi tertentu.

Karakteristik ini terbagi menjadi beberapa subkarakteristik, yaitu:

- a. *Time Behaviour*, yaitu sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- b. *Resource Utilization*, yaitu sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- c. *Capacity*, yaitu sejauh mana batas maksimum parameter, produk, atau sistem dapat memenuhi persyaratan.

8. *Maintainability*

Sejauh mana keefektifan dan efisiensi dari sebuah produk atau sistem dapat dirawat.

Karakteristik ini dibagi menjadi beberapa karakteristik, yaitu:

- a. *Modularity*, yaitu sejauh mana sistem terdiri dari komponen terpisah sehingga perubahan atau modifikasi pada salah satu komponen tersebut memiliki dampak yang kecil terhadap komponen yang lain.
- b. *Reusability*, yaitu sejauh mana asset dapat digunakan lebih oleh satu sistem atau digunakan untuk membangun asset lain.
- c. *Analyzability*, yaitu tingkat efektivitas dan efisiensi untuk mengkaji dampak perubahan pada satu atau lebih bagian-bagian produk atau sistem, untuk mendiagnosis

kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan dirubah.

- d. *Modifiability*, yaitu sejauh mana produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menurunkan kualitas produk yang ada.
- e. *Testability*, yaitu tingkat efektivitas dan efisiensi untuk membentuk kriteria uji dari produk, sistem, atau komponen dan uji dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.