

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori Umum

Dalam bagian ini, penulis akan memberikan penjelasan berkenaan dengan teori-teori umum yang saling berkaitan dengan berjalannya penelitian ini, penjelasan yang dimaksud bersifat penjelasan secara umum.

2.1.1 Sistem

Sistem (Mulia, 2020), ialah suatu satuan entitas yang terbentuk dari dua atau lebih bagian atau sistem yang kecil yang saling berkaitan dan berhubungan untuk menggapai suatu tujuan. Dalam (Romindo & Hondro, 2018) juga berpendapat bahwa sistem adalah sekelompok komponen yang memiliki hubungan menerima *input* dan menghasilkan *output* dalam suatu kegiatan transformasi yang terstruktur dengan maksud yang jelas dan saling bekerja sama untuk menggapai suatu tujuan. Jadi, elemen atau komponen yang saling tergantung dan berhubungan satu dan lainnya untuk menggapai suatu tujuan tertentu ialah definisi dari sistem (Koswoyo & Mauludin, 2021).

Berdasarkan teori yang sudah diuraikan sebelumnya maka peneliti menyimpulkan bahwasanya sistem adalah sekumpulan elemen, entitas ataupun komponen yang terbentuk dari beberapa komponen yang saling berhubungan dan bekerja sama untuk menggapai sebuah tujuan.

Suatu sistem bisa dikatakan sebuah sistem apabila memiliki karakteristik atau sifat-sifat tertentu sebagai berikut (Paryanta, Sutariyani, & Susilowati, 2017):

1. *Components* (Komponen Sistem)
2. *Boundary* (Batasan Sistem)
3. *Environment* (Lingkungan Luar Sistem)
4. *Interface* (Penghubung Sistem)
5. *Input* (Masukan Sistem)
6. *Output* (Keluaran Sistem)
7. *Process* (Pengolah Sistem)
8. *Objective* (Sasaran Sistem)

2.1.2 Informasi

Informasi didefinisikan sebagai suatu data yang telah diproses menjadi wujud yang lebih bermakna bagi sang penerima (Koswoyo & Mauludin, 2021). Hasil pengolahan data tersebut menjadi informasi dapat dijadikan sebagai suatu dasar dalam mengambil keputusan yang konsekuensinya dapat dirasakan langsung atau tidak langsung dikemudian hari (Elisa, Azwanti, & Simanjuntak, 2020).

Menurut (Tukino & Amrizal, 2020), kualitas informasi dipengaruhi dari tiga faktor berupa akurat, tepat pada waktunya dan relevan. Suatu informasi bisa tidak diperlukan jika tidak ada pilihan atau keputusan, karena nilai informasi berhubungan erat dengan keputusan (Ayu & Fitri, 2019).

Dari penjelasan yang diberikan di atas, maka peneliti menyimpulkan bahwa pengertian dari informasi adalah data yang telah melalui proses pengolahan yang menghasilkan bentuk yang memiliki arti bagi penerimanya.

2.1.3 Sistem Informasi

Sistem informasi merupakan gabungan dari bagian yang saling bekerja sama dalam menyatukan, memproses, menyimpan dan mendelegasikan informasi yang berkaitan untuk membantu pengendalian, koordinasi dan pengambilan suatu keputusan (Romindo & Hondro, 2018). Sistem informasi berfungsi untuk mencapai suatu target atau tujuan yang biasanya mencakup beberapa komponen seperti manusia, komputer, teknologi informasi dan biasanya terdapat sesuatu yang akan diproses berupa data menjadi informasi (Batubara, Raihan, Tanjung, Fadlurohman, & Can, 2022).

Dapat dikatakan bahwa sistem informasi merupakan sebuah sistem dalam organisasi yang mendukung operasi pengolahan transaksi harian dan kegiatan lainnya dari suatu organisasi yang menghasilkan laporan-laporan yang diperlukan (Koswoyo & Mauludin, 2021).

Berdasarkan uraian diatas, peneliti menyimpulkan bahwa sistem informasi didefinisikan sebagai sekumpulan sistem yang saling berhubungan yang bekerja sama untuk menggabungkan, memproses, mengendalikan, menyimpan, dan mendelegasikan informasi untuk menghasilkan laporan yang bersangkutan.

2.1.4 SDLC (*Software Development Life Cycle*)

SDLC (Software Development Life Cycle) digunakan untuk menghasilkan *software* (perangkat lunak) dengan proses pengembangan, apapun bisa dilakukan dengan cepat dan murah (Purwanto, Wijaya, Timotius, & Trisno, 2022).

SDLC (Software Development Life Cycle) biasanya memiliki tahapan terstruktur dalam pengembangan sistem, mulai dari analisis sistem sampai dengan

proses perancangan sistem yang dilakukan oleh pemrograman sistem (Lesmana & Silalahi, 2022). Metodologi yang terdapat dalam *SDLC (Software Development Life Cycle)* berupa (Ayu & Fitri, 2019) :

1. *Waterfall*
2. *Prototyping*
3. *Rapid Application Development (RAD)*
4. *Agile Software Development*

Tahapan pengembangan yang terdapat dalam *SDLC (Software Development Life Cycle)* dapat membantu perusahaan dalam menghasilkan *software* atau perangkat lunak yang teruji dan siap pakai (Purwanto et al., 2022). Dalam penelitian yang dilakukan oleh (Setiawan, Syaifullah, & Putra, 2021) mengungkapkan bahwa tahapan umum dalam *SDLC (Software Development Life Cycle)* yaitu :

1. *Planning*

Tahapan awal dalam pengembangan sistem yaitu perencanaan (*planning*). Dalam tahapan ini pengembang sistem melakukan perencanaan (*planning*) terhadap sistem yang akan disesuaikan berdasarkan kebutuhan dari pengguna sistem.

2. *Requirement Analysis*

Pada tahap ini, pengembang sistem melakukan analisis yang dibutuhkan mengenai kebutuhan-kebutuhan pengguna terhadap sistem yang nantinya akan dikembangkan, serta melakukan pengembangan sistem berdasarkan hasil analisa tersebut.

3. *Design*

Tahapan ini merupakan proses dimana pengembang sistem mengubah dan merealisasikan kebutuhan dari pengguna sistem menjadi sebuah rancangan desain sistem yang dapat memenuhi fungsi-fungsi yang dibutuhkan pengguna.

4. *Development*

Dalam tahapan ini, pengembang sistem mengubah hasil rancangan desain sistem ke dalam bahasa pemrograman untuk dijadikan sebuah sistem informasi yang dapat memenuhi kebutuhan-kebutuhan pengguna sistem.

5. *Testing & Integration*

Dalam tahapan ini, melibatkan pengembang dan pengguna sistem untuk menguji sistem, memperbaiki kesalahan, dan memastikan bahwa sistem telah memenuhi kebutuhan pengguna.

6. *Implementation*

Dalam fase ini, *developer* akan menginterasikan sistem yang telah dikembangkan ke dalam ekosistem lingkungan kerja dari pengguna sistem.

7. *Maintenance*

Maintenance merupakan tahapan akhir dari *SDLC* (*Software Development Life Cycle*). Dimana dalam tahapan ini dilakukan pemeliharaan terhadap sistem yang telah dikembangkan.

Dalam penelitian ini penulis akan menggunakan metodologi pengembangan *agile* dengan model *scrum*. Dimana penjelasan secara umum mengenai metodologi *agile* akan penulis uraikan di sub bab berikut.

2.1.5 Agile

Metode *agile* merupakan salah satunya metode dalam *SDLC* (*Software Development Life Cycle*) yang tahapan pengembangannya lebih berpusat pada orang dan hubungan antar satu sama lain, fungsi *software*, koalisi pelanggan dan pergantian daripada metode, alat, kontrak dan rencana (Hayat, Rehman, Arif, Wahab, & Abbas, 2019).

Metode *agile* memberikan tahapan perencanaan yang fleksibel, pengembangan evolusioner, pengiriman awal, peningkatan yang berkelanjutan dan perubahan persyaratan yang dapat di sesuaikan setiap saat selama proses pengembangan sistem (Mallidi & Sharma, 2021).

Scrum merupakan salah satu *framework* (kerangka kerja) dalam metode *agile* karena *scrum* berfokus pada manajemen proyek sehari-hari dan merupakan *framework* (kerangka kerja) *agile* yang paling mendekati metode manajemen proyek pada *agile* (Hayat et al., 2019). *Scrum* memberikan fleksibilitas untuk mengendalikan dan mengelola persyaratan serta pengembangan perangkat lunak karena *scrum* menggunakan model basis iteratif dan inkremental yang membangun *software* dengan mekanisme yang telah di tentukan, seperti suatu modul perangkat lunak dapat berkembang dalam potongan-potongan kecil secara iteratif (Hayat et al., 2019). Beberapa tahapan yang biasanya dilakukan dalam metode pengembangan *agile* model *scrum* adalah sebagai berikut (Nikolaus & Octaviano, 2022) :

1. *Product Backlog*

Dalam tahap ini, akan ditentukan prioritas dalam mengerjakan *sprint*.

2. *Sprint Planning Meeting*

Dalam tahap ini, seluruh anggota tim berkumpul untuk mengidentifikasi tugas dari masing-masing anggota, proses ini sangat penting untuk dilakukan sebelum menjalankan *sprint*.

3. *Daily Stand Up Meeting*

Dalam tahap ini, tugas-tugas pekerjaan yang dilakukan anggota tim akan dievaluasi beserta kendala yang dihadapi. Proses berjalan bersamaan dengan proses *sprint*.

4. *Sprint Review*

Setiap selesainya satu *sprint*, setiap anggota tim akan mendemonstrasikan tugas yang telah diselesaikan selama periode tersebut. Kegiatan ini dilakukan setiap selesainya satu periode *sprint*.

5. *Sprint Retrospective*





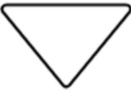
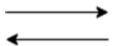
Tahap ini dilakukan pada saat setiap sesi *sprint* berakhir. Dalam tahap ini semua anggota tim bisa memberikan pendapat dan evaluasi yang berkaitan dengan kinerja selama menjalankan metode *scrum*.

2.1.6 Aliran Sistem Informasi

Aliran sistem informasi tersusun atas kumpulan-kumpulan simbol yang akan dilaksanakan oleh sistem atau program yang merupakan bagian yang menunjukkan arah logika dari data yang akan diproses dari awal sampai akhir dari sebuah sistem atau program (Sitompul & Arnomo, 2022). Dengan adanya aliran sistem informasi, dapat dilihat apakah sebuah sistem atau program masih layak untuk digunakan atau tidak layak untuk digunakan, masih dilakukan secara manual ataupun sudah terkomputerisasi, dengan kata lain aliran sistem informasi berfungsi untuk

mengetahui permasalahan yang dimiliki oleh suatu sistem atau program (Sukrianto & Oktarina, 2020). Beberapa simbol yang dipakai dalam aliran sistem informasi dijelaskan pada Tabel 2.1.

Tabel 2.1 Simbol aliran sistem informasi

Gambar	Nama	Keterangan
	<i>Process</i>	<i>Process</i> menunjukkan proses yang dilakukan oleh komputer.
	<i>Manual Operation</i>	<i>Manual Operation</i> menunjukkan proses yang dijalankan secara manual.
	<i>Document</i>	<i>Document</i> menunjukkan dokumen atau data yang digunakan sebagai <i>input</i> atau <i>output</i> .
	<i>Database</i>	<i>Database</i> menunjukkan <i>input</i> atau <i>output</i> yang menggunakan database.
	<i>Storage</i>	<i>Storage</i> menunjukkan dokumen yang diarsip dan disimpan secara manual.
	<i>Flow Line</i>	<i>Flow Line</i> menunjukkan arah aliran dari proses.


2.1.7 UML (*Unified Modeling Language*)

UML (Unified Modeling Language) diperkenalkan pertama kalinya pada tahun 1997, yang merupakan bahasa pemodelan perangkat lunak yang saat ini telah dikembangkan menjadi bahasa pemodelan yang baku (*de facto*) dalam sebuah pengembangan *software* atau perangkat lunak yang berorientasi objek (Kurniawan, 2018). *UML (Unified Modeling Language)* tidak hanya sebagai sebuah metodologi yang digunakan dalam pengembangan sistem yang berorientasi pada objek, tetapi juga digunakan untuk melakukan dokumentasi, spesifikasi dan membangun sebuah perangkat lunak sebagai pendukung dalam pengembangan sistem (Koswoyo & Mauludin, 2021). Beberapa jenis diagram *UML (Unified Modeling Language)* yang sering digunakan sebagai berikut :


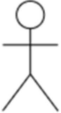

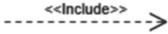



1. *Use case* diagram

Use case diagram ialah diagram yang memberikan gambaran pemodelan untuk *behavior* (kelakuan) dari sistem yang akan dibuat (Aini & Pratama, 2022). Beberapa simbol yang digunakan dalam *use case* diagram dijelaskan pada Tabel 2.2 di bawah.



Tabel 2.2 Simbol *use case* diagram

Simbol	Nama	Deskripsi
	<i>Use case</i>	<i>Use case</i> menyatakan tujuan dari interaksi <i>user</i> dengan sistem yang dapat dicapai.

Tabel 2.2 Lanjutan

	<i>Association</i>	<i>Association</i> menyatakan hubungan antara <i>actor</i> dengan <i>use case</i> untuk mencapai sebuah <i>use case</i> .
	<i>Actor</i>	<i>Actor</i> menyatakan seluruh orang atau entitas yang melakukan interaksi dengan sistem.
	<i>System</i>	<i>System</i> menyatakan batasan dan ruang lingkungnya melalui sistem.
	<i>Include</i>	<i>Include</i> menyatakan pemakaian fungsi yang terdapat dalam <i>use case</i> lainnya.
	<i>Extend</i>	<i>Extend</i> menyatakan perluasan batasan <i>use case</i> .
	<i>Dependency</i>	<i>Dependency</i> menyatakan pemakaian fungsi yang terdapat dalam <i>use case</i> lainnya secara opsional.
	<i>Generalization</i>	<i>Generalization</i> menyatakan persamaan yang ada antara <i>actor</i> dan <i>use case</i> .

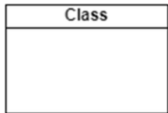

Tabel 2.2 Lanjutan

	<i>Realization</i>	<i>Realization</i> menyatakan hubungan yang ada antara spesifikasi dan implementasinya.
	<i>Collaboration</i>	<i>Collaboration</i> menyatakan struktur elemen yang saling bekerja sama, yang melakukan tugas masing-masing, dan bekerja sama untuk melakukan beberapa tugas yang diinginkan.








2. Class diagram

Class diagram menggambarkan deskripsi kelas dan struktur, paket dan objek serta hubungannya antara satu dan lain, seperti *inheritance* (pewarisan) *association* (asosiasi) dll (Sovia, Yanto, & Nursam, 2020). Beberapa simbol yang digunakan dalam *class* diagram dijelaskan dilihat pada Tabel 2.3 di bawah.


Tabel 2.3 Simbol *class* diagram

Simbol	Nama	Deskripsi
	<i>Class</i>	<i>Class</i> menyatakan kumpulan objek yang masing-masing mempunyai struktur, perilaku, dan relasi yang sama dengan objek <i>class</i> lain.
	<i>Generalization</i>	<i>Generalization</i> menyatakan hubungan dimana objek anak (<i>descendant</i>) memiliki karakteristik yang mirip dengan objek yang ada di atas objek induk (<i>ancestor</i>).

Tabel 2.3 Lanjutan

	<i>Realization</i>	<i>Realization</i> menunjukkan hubungan yang ada pada <i>interface</i> dan <i>class</i> atau komponen yang melaksanakannya.
	<i>Association</i>	<i>Association</i> menunjukkan hubungan antara dua <i>class</i> .
	<i>Aggregation</i>	<i>Aggregation</i> menyatakan asosiasi dengan hubungan antara keseluruhan dan bagian-bagiannya, relasi ketika suatu <i>class</i> adalah entitas tertentu yang menyertakan entitas lain sebagai komponen.
	<i>Composition</i>	<i>Composition</i> merupakan varian <i>aggregation</i> yang kuat dalam kasus di mana komponen tidak dapat di pisahkan dari ensemble.
	<i>Dependency</i>	<i>Dependency</i> menyatakan hubungan antara entitas atau elemen yang <i>independent</i> akan mempengaruhi entitas atau elemen yang <i>dependent</i>
	<i>N-ary Association</i>	<i>N-ary association</i> menyatakan <i>aggregation</i> yang berjumlah dua atau lebih.
	<i>Collaboration</i>	<i>Collaboration</i> menyatakan urutan kegiatan yang ditunjukkan oleh sistem yang menghasilkan sebuah hasil yang dapat diukur bagi suatu <i>actor</i> .


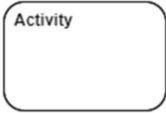

Tabel 2.3 Lanjutan

	<i>Package</i>	<i>Package</i> menyatakan pengelompokan <i>class</i> dan <i>package</i> lainnya.
---	----------------	--





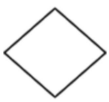
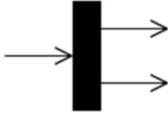
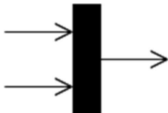
3. *Activity* diagram

Activity diagram menerangkan tentang langkah atau tahapan kerja pada sebuah proyek yang sedang dikembangkan. Langkah atau tahapan kerja ini bisa terdiri dari menu-menu atau proses bisnis yang terdapat dalam proyek sistem tersebut (Krismanto, Adhitama, & Prasetyo, 2022). Beberapa simbol yang digunakan *activity* diagram dijelaskan pada Tabel 2.4.

Tabel 2.4 Simbol *activity* diagram

Simbol	Nama	Deskripsi
	<i>Initial Node</i>	<i>Initial node</i> menyatakan sebuah titik awal atau kondisi awal dari sebuah aktivitas.
	<i>Activity</i>	<i>Activity</i> menyatakan sebuah proses aktivitas.
	<i>Action</i>	<i>Action</i> menyatakan sub-area yang dapat dilakukan dari sebuah aktivitas.


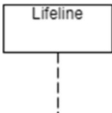
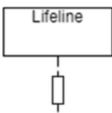

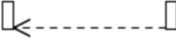
Tabel 2.4 Lanjutan

	<i>Control Flow</i>	<i>Control flow</i> menyatakan suatu aliran dari sebuah aktivitas ke aktivitas lainnya.
	<i>Object Flow</i>	<i>Object flow</i> menyatakan suatu aliran dari sebuah objek ke objek lainnya.
	<i>Activity Final Node</i>	<i>Activity final node</i> menyatakan sebuah titik akhir atau kondisi akhir dari sebuah aktivitas.
	<i>Flow Final Node</i>	<i>Flow final node</i> menyatakan akhir dari sebuah aliran aktivitas tunggal.
	<i>Decision Node</i>	<i>Decision node</i> menyatakan percabangan kondisional dari aktivitas yang memiliki satu <i>input</i> dan beberapa <i>output</i> .
	<i>Fork Node</i>	<i>Fork node</i> menyatakan satu aliran t yang dapat bercabang menjadi banyak aliran paralel.
	<i>Join Node</i>	<i>Join node</i> menyatakan dua buah <i>input</i> yang digabungkan menjadi satu buah <i>output</i> .



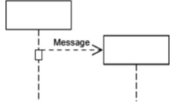
4. *Sequence* diagram

Sequence diagram menggambarkan kelakuan antar objek dalam *use case diagram* dengan mendeskripsikan *lifeline* (waktu hidup) objek dan *message* (pesan) yang dikirimkan dan diterima sesama objek (Julianto & Setiawan, 2019). Beberapa simbol yang digunakan dalam *sequence* diagram dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol *sequence* diagram

Simbol	Nama	Deskripsi
	<i>Actor</i>	<i>Actor</i> menyatakan entitas yang berinteraksi dengan sistem atau yang berada di luar sistem.
	<i>Lifeline</i>	<i>Lifeline</i> menyatakan entitas atau peserta individu dalam suatu interaksi.
	<i>Activation</i>	<i>Activation</i> menyatakan jangka waktu yang dilalui suatu elemen dalam melakukan sebuah operasi.
	<i>Call Message</i>	<i>Call message</i> merupakan jenis <i>message</i> yang menyatakan permintaan operasi pada <i>lifeline</i> .
	<i>Return Message</i>	<i>Return message</i> merupakan jenis <i>message</i> yang menyatakan penyampaian kembali informasi ke <i>caller</i> dari pesan koresponden sebelumnya.

Tabel 2.5 Lanjutan

	<p><i>Self Message</i></p>	<p><i>Self message</i> merupakan jenis <i>message</i> yang menyatakan permintaan <i>message</i> dari <i>lifeline</i> yang sama.</p>
	<p><i>Recursive Message</i></p>	<p><i>Recursive message</i> merupakan jenis <i>message</i> yang menyatakan permintaan <i>message</i> dari <i>lifeline</i> yang sama. Tujuannya menunjuk ke aktivasi di atas lokasi aktivasi tempat <i>message</i> itu dipanggil.</p>
	<p><i>Create Message</i></p>	<p><i>Create message</i> merupakan jenis <i>message</i> yang menyatakan instantiasi dari (target) <i>lifeline</i>.</p>

2.1.8 Database

Database atau basis data terdiri atas dua kata, yaitu basis dan data. Basis memiliki arti sebagai markas atau gudang, tempat berkumpul. Sedangkan data merupakan representasi fakta dalam dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkap dalam bentuk angka, huruf, simbol, teks, gambar, suara atau kombinasinya (Mulyanto & Khasanah, 2018).

Dengan kata lain *database* atau basis data ialah kumpulan data dan deskripsi yang berhubungan secara logis, yang dipakai untuk memperoleh informasi dan pengetahuan, yang secara fisik disimpan dalam bentuk kumpulan *bit* yang direkam dengan basis *track* dalam suatu perangkat penyimpanan (Dalimunthe, 2020). Data-

data tersebut tersusun atas kumpulan *bit* yang membentuk *field*, kumpulan *field* yang membentuk *record*, kumpulan *record* yang membentuk tabel dan kumpulan tabel yang membentuk *database* atau basis data (Dalimunthe, 2020).

Jadi *database* atau basis data bisa didefinisikan sebagai kumpulan dari data yang saling berhubungan (*interrelated data*) yang disimpan secara bersamaan dalam suatu media penyimpanan, tanpa mengatap satu sama lain atau tidak perlu suatu redundansi data (*controlled redundancy*) (Ayu & Fitri, 2019).

2.2 Tinjauan Teori Khusus

Dalam bagian ini, penulis akan memberikan penjelasan tentang teori-teori khusus yang saling berkaitan dengan penelitian ini, penjelasan yang dimaksud berupa pengertian secara umum.

2.2.1 Proyek

Proyek adalah sebuah aktifitas usaha yang kompleks dan sifatnya tidak rutin, memiliki memiliki batas waktu, dana, sumber daya, serta memiliki spesifikasi tersendiri atas produk yang akan dihasilkan (Nurhidayat, Arianto, & Bhirawa, 2021).

Dengan kata lain, proyek dapat diartikan sebagai serangkaian kegiatan yang memiliki waktu dimensi yang terbatas, dengan alokasi sumber dana dan sumber daya tertentu, yang berguna untuk mewujudkan suatu gagasan serta memiliki tujuan tertentu setelah gagasan tersebut dinyatakan layak untuk dijalankan (Sana Ose, Mochtar, & Tohir, 2020).

2.2.2 Pemesanan

Pemesanan dalam arti umum menurut Edwin dan Chris dalam (Satri & Thahjo Seabtion, 2019) adalah suatu perjanjian yang terjadi antara dua pihak atau lebih yang meliputi perjanjian pemesanan produk barang ataupun jasa.

Pemesanan barang atau jasa ini biasanya diikuti oleh transaksi finansial berupa pelunasan pembayaran akan produk ataupun jasa yang diinginkan sebagai suatu indikator atas rencana atau keinginan untuk memiliki atau membeli barang dan jasa tersebut (Utama et al., 2019).

Dalam arti lain pemesanan adalah proses, pembuatan, cara memesan barang atau jasa kepada pihak lain (Handrianto & Sanjaya, 2020). Jadi pemesanan adalah suatu adalah suatu kegiatan yang dilakukan oleh pelanggan sebelum membeli suatu barang atau jasa (Julianto & Setiawan, 2019). Kelanjutan dari pemesanan barang atau jasa adalah pengiriman barang atau jasa agar dapat sampai dengan selamat ke pelanggan yang memesan (Satri & Thahjo Seabtion, 2019).

2.2.3 Pelanggan

Setiap individu yang memakai barang, jasa atau keduanya yang tersedia dalam lingkungan masyarakat baik untuk kepentingan pribadi, keluarga ataupun orang lain dan tidak untuk diperdagangkan disebut sebagai pelanggan. Pelanggan sendiri dapat diartikan sebagai sekelompok ataupun seorang individu yang membeli barang atau jasa dengan mempertimbangkan beberapa faktor seperti *cost*, *quality*, *service* dan lain sebagainya berdasarkan keputusan dari pembeli (Sudaria, Putra, & Novembrianto, 2021).

Jadi, pelanggan dapat diartikan sebagai kelompok atau individu yang berdasarkan keputusan sendiri melakukan pertimbangan atas harga, kualitas dan pelayanan, membeli dan menggunakan barang atau jasa dari orang atau perusahaan lain (Lestari & Santoso, 2022).

2.2.4 Sistem Informasi Proyek

Sistem informasi proyek adalah suatu sistem yang dipakai dalam manajemen proyek untuk menyampaikan informasi, yang digunakan untuk melakukan perbandingan *baseline* dengan proyek aktual dari setiap kegiatan, mengelola material dan mencatat laporan, yang biasanya untuk menggantikan pekerjaan manajemen proyek yang selama ini dilakukan secara manual (Sanjaya & Andry, 2019). Berdasarkan penelitian (Praudian & Ramadhani, 2021), sistem informasi proyek ditekankan pada tiga faktor, yaitu manusia, masalah dan proses, dimana dalam pekerjaan sistem informasi faktor manusia merupakan peran yang penting dalam menentukan suksesnya suatu manajemen proyek yang dinyatakan dalam model kematangan kemampuan manajemen manusia (*a people management capability maturity model/ PM-CMM*) yang bertujuan dalam meningkatkan kesiapan organisasi suatu *software* sistem informasi dalam menyelesaikan masalah dengan melakukan kegiatan menerima, memilih, kinerja manajemen, pelatihan, kompensasi, pengembangan karier, organisasi dan rancangan kerja dan juga pengembangan tim.

2.2.5 Sistem Pemesanan

Sistem pemesanan merupakan suatu sistem dalam perusahaan yang menawarkan suatu barang atau jasa yang sifatnya belum jadi atau tidak berwujud,

dan baru dapat terlihat hasilnya setelah melalui tahapan produksi sesuai dengan pemesanan yang dilakukan oleh pelanggan. Sebuah perusahaan mestinya memiliki suatu sistem pemesanan yang baik untuk mewujudkan kepuasan pelanggan (Saputri, Oktavia, Ramdhani, & Suherman, 2019).

Berdasarkan penelitian (C. Chandra & Susilo, 2021) dalam penelitiannya yang berjudul Perancangan Sistem Pemesanan Online Jasa Print Out Berbasis Android Menggunakan Metode Scrum, menyatakan bahwa tujuan dari sistem pemesanan yaitu untuk meningkatkan pelayanan bagi pelanggan, meminimalisir investasi pada stok, sebagai perencanaan akan kapasitas serta persediaan.

2.2.6 Sistem Informasi Pemesanan

Secara umum, sistem informasi pemesanan adalah sebuah sistem informasi yang digunakan di suatu perusahaan dalam usahanya, untuk membantu dan memudahkan pencarian mengenai informasi-informasi pemesanan sebuah produk atau jasa (Heriansyah & Triawan, 2020).

Sistem informasi pemesanan adalah sistem informasi yang dapat mengendalikan arus transaksi pemesanan, dimulai dari proses perencanaan suatu barang atau jasa sebelum diproses dan diolah oleh sistem terstruktur yang kemudian dilanjutkan dengan transaksi finansial berupa pembayaran tunai terhadap barang atau jasa yang dipesan (Utama et al., 2019).

Sistem informasi pemesanan juga dapat dijadikan sebagai suatu media yang menjadi sarana dalam menyediakan informasi mengenai produk, jasa serta promosi-promosi yang ada (Heriansyah & Triawan, 2020).

2.2.7 Visual Studio Code

Visual studio code adalah sebuah *IDE (Integrated Development Environment)*, yaitu editor perangkat lunak dengan ukuran yang kecil namun kuat. *Visual studio code* memiliki dukungan bawaan untuk bahasa pemrograman seperti *JavaScript*, *Scripts*, *Node.js*, dan berbagai ekstensi lainnya, yang tersedia dalam berbagai bahasa seperti *C++*, *C#*, *Python* dan *PHP* (Noviantoro, Silviana, Fitriani, & Permatasari, 2022).

2.2.8 PHP

PHP pada mulanya dikembangkan oleh Rasmus Lerdorf pada tahun 1995, yang digunakan sebagai bahasa pemrograman umum *server-side scripting* yang dirancang untuk perancangan *website* (Irianto, Anshori, & Mas'udi, 2020). *PHP* sendiri merupakan sebuah akronim rekursif untuk *PHP Hypertext Preprocessor*, dan merupakan sebuah perangkat lunak *open source* dengan bahasa pemrograman *server-side scripting*, yaitu dimana semua *script* dari *PHP* diproses dan dijalankan seluruhnya oleh *server* (H. D. Chandra & Amrizal, 2023). *PHP* secara luas banyak dipakai untuk pengembangan dan pembuatan sebuah *website* dan bisa dijalankan bersamaan dengan *HTML* (Sasono, 2020).

2.2.9 XAMPP

XAMPP merupakan *software gratis (freeware)* yang dirancang untuk dapat digunakan diberbagai macam sistem operasi dan berfungsi sebagai *Localhost* atau *server* yang berdiri sendiri (*stand-alone*) (Dalimunthe, 2020). *XAMPP* merupakan paket program web lengkap yang sering digunakan untuk media pembelajaran pemograman web, khususnya *PHP* dan *MySQL* yang berfungsi sebagai *server* yang

berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan menggunakan bahasa pemrograman *PHP* (Anggraini, Pasha, Damayanti, & Setiawan, 2020).

2.2.10 MySQL

MySQL adalah *software* atau perangkat lunak sistem manajemen *database* atau basis data yang sangat populer digunakan untuk pembangunan aplikasi web yang menggunakan *database* atau basis data sebagai sumber dan pengelola datanya, terutama dikalangan pemrograman web (Batubara et al., 2022). *MySQL* sebagai *software* atau perangkat lunak sistem manajemen *database* atau basis data ini bersifat *open source* dan bertujuan untuk menangani sistem manajemen *database* atau basis data dan sistem manajemen *database* atau basis data relasional (*Relational Database Management System*) (Nasution & Sari, 2020).

Jadi *MySQL* merupakan *software* atau perangkat lunak sistem manajemen *database* atau basis data *SQL* atau *DBMS* yang *mutithread* dan *multiuser* (Fernando & Harman, 2023).

2.2.11 PhpMyAdmin

PhpMyAdmin merupakan suatu aplikasi *open source* berbasis web yang dibuat dengan menggunakan bahasa pemrograman *PHP*, yang berguna untuk mengakses *database* atau basis data *MySQL* (Mulyanto & Khasanah, 2018). *PhpMyAdmin* ini berbasis *GUI* (*Graphical User Interface*) yang berguna untuk mengelola *database* atau basis data *MySQL* (Maryani, Ishaq, & Mulyadi, 2018). Dengan menggunakan *PhpMyAdmin*, akan sangat memudahkan dan mempercepat

kinerja dalam mengelola *database* atau basis data *MySQL* (Mulyanto & Khasanah, 2018).

2.2.12 Javascript

Pada tahun 1995 *Javascript*, yang pada awalnya diberi nama *LiveScript* yang pertama kali diperkenalkan oleh Netscape, memiliki fungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2 (Sahi, 2020). *Javascript* ini merupakan *script* program berbasis klien (*client based*) yang dijalankan oleh *browser* sehingga dapat membuat halaman web melakukan aktifitas dan tugas tambahan yang tidak dapat dilakukan oleh *script HTML* biasa (Noviantoro et al., 2022). Sepanjang sejarah internet, *javascript* merupakan bahasa *script* pertama yang dikembangkan untuk web, yang memberikan kemampuan tambahan terhadap bahasa *HTML* dengan memberikan izin untuk mengeksekusi perintah di sisi *user* pada *browser* (Sahi, 2020).