

BAB II

TINJAUAN PUSTAKA

2.1 UMKM

Menurut Prasetyo Bella Ramadhanu (2021), Usaha Mikro, Kecil, dan Menengah (UKM) dalam Undang-Undang Nomor 20 Tahun 2008 adalah usaha produktif yang dijalankan oleh perorangan atau badan usaha perorangan yang memenuhi kriteria tertentu sesuai dengan ketentuan yang diatur dalam undang-undang tersebut.

2.2 Manajemen

Menurut Divangga Revansa Arya Pradhana (2022), manajemen adalah proses yang melibatkan perencanaan, pengorganisasian, penggerakan, dan pengendalian untuk mencapai tujuan yang telah ditetapkan dengan memanfaatkan sumber daya manusia dan sumber daya lainnya secara efektif dan efisien.

2.3 Manajemen Keuangan

Menurut Hielmi Sulaeman (2023), manajemen keuangan adalah serangkaian aktivitas yang mencakup perolehan, pembiayaan, dan pengelolaan aset untuk mencapai tujuan tertentu. Dalam kehidupan sehari-hari, pengelolaan keuangan menjadi hal penting bagi setiap individu untuk menyeimbangkan pemasukan dan pengeluaran, memenuhi kebutuhan hidup, serta menghindari kesulitan finansial. Oleh karena itu, pengelolaan keuangan yang baik dan optimal sangat diperlukan agar seseorang dapat mencapai tujuan finansialnya dan terhindar dari kegagalan ekonomi.

2.4 Pusat Layanan Usaha Terpadu (PLUT)

Menurut Moch Waris Laksana Agung (2022), Pusat Layanan Usaha Terpadu (PLUT) adalah lembaga yang menyediakan layanan pendampingan non-finansial secara menyeluruh dan terintegrasi bagi pelaku Koperasi dan Usaha Mikro, Kecil, dan Menengah (KUMKM) untuk meningkatkan daya saing melalui bimbingan produksi, pemasaran, akses pembiayaan, serta pengembangan kapasitas kewirausahaan, teknis, manajerial, dan kelembagaan.

2.5 Aplikasi

Menurut Albert Budiyo (2023), aplikasi adalah perangkat lunak yang berfungsi sebagai antarmuka (*front-end*) dalam suatu sistem untuk mengolah data menjadi informasi yang berguna bagi pengguna dan sistem terkait. Secara umum, aplikasi merupakan program siap pakai yang dirancang untuk menjalankan fungsi tertentu, membantu menyelesaikan masalah melalui teknik pemrosesan data yang sesuai dengan kebutuhan atau harapan pengguna.

2.6 Visual Studio Code

Menurut Agustini (2019), *Visual Studio Code* adalah editor kode sumber buatan *Microsoft* yang tersedia untuk *Windows*, *Linux*, dan *macOS*. Editor ini mendukung berbagai fitur seperti *debugging*, kontrol *Git* dan *GitHub*, penyorotan sintaksis, penyelesaian kode cerdas, *snippet*, dan *refactoring* kode. *Visual Studio Code* juga sangat dapat disesuaikan, memungkinkan pengguna mengubah tema, pintasan *keyboard*, pengaturan, serta menginstal ekstensi untuk menambah fungsionalitas.

2.7 Xampp

Menurut A. Taqwa Martadinata (2020), XAMPP adalah sebuah paket *software* lengkap yang terdiri dari beberapa aplikasi seperti *Apache*, *MySQL*, *PHPMyAdmin*, *PHP*, *Perl*, *FileZilla*, dan lainnya. XAMPP digunakan untuk memudahkan proses pengembangan dan pengujian aplikasi *web* secara lokal tanpa perlu konfigurasi *server* secara manual.

2.8 PHP

Menurut Happy Anita Margaretha (2020), PHP adalah bahasa pemrograman skrip yang berjalan di sisi *server* (*server-side*) dan dirancang khusus untuk pengembangan *web*, meskipun juga dapat digunakan sebagai bahasa pemrograman umum. Dikembangkan oleh Rasmus Lerdorf pada tahun 1995, PHP awalnya merupakan singkatan dari *Personal Home Page*, yang digunakan untuk membuat situs *web* pribadi. Saat ini, PHP dikelola oleh The PHP *Group* dan telah berkembang menjadi bahasa pemrograman *web* yang kuat, digunakan mulai dari pembuatan halaman web sederhana hingga situs *web* berskala besar. PHP diproses di *server*, berbeda dengan bahasa seperti *JavaScript* yang diproses di sisi klien (*browser*). PHP dirilis di bawah lisensi *PHP License*, yang berbeda dari lisensi *GNU GPL* yang umum pada proyek-proyek *open source*.

2.9 MySQL

Menurut Tumini (2021), *MySQL* adalah sistem manajemen basis data (*database*) yang digunakan sebagai media penyimpanan data dan mendukung integrasi dengan *script PHP*. *MySQL* menggunakan bahasa *query SQL* (*Structured Query Language*) yang sederhana dan memiliki kemiripan dalam penggunaan

escape character dengan *PHP*. Selain kemudahan penggunaannya, *MySQL* dikenal sebagai salah satu *database* tercepat dan paling populer dalam pengembangan aplikasi *web*.



2.10 Database


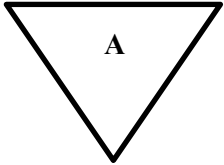
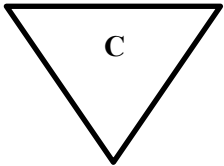
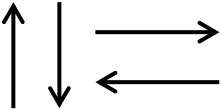

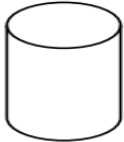
Menurut Maisila Cendani (2023), Basis Data (*Database*) adalah komponen yang berada di belakang layar, sering disebut dengan *back-end*, sedangkan *user interface* adalah komponen yang digunakan untuk menampilkan data ke hadapan *user* dan untuk memudahkan *user* dalam memanipulasi data yang terdapat di dalam *database*. *User Interface* sering disebut sebagai *front-end*.

2.11 Aliran Sistem Informasi

Menurut Wahyu Pudyawardana (2023), Aliran sistem informasi adalah bagan alir yang menunjukkan arus program dan formulir, termasuk tembusannya. Aliran ini berguna untuk mengidentifikasi permasalahan dalam sistem, serta membantu menilai apakah sistem tersebut masih layak digunakan, masih manual atau sudah terkomputerisasi. Jika sistem dianggap tidak layak, maka perlu dilakukan perubahan dalam pengolahan data agar dapat menghasilkan informasi yang cepat, akurat, dan mendukung pengambilan keputusan yang lebih baik.

Tabel 2.1 Simbol – simbol pada Aliran Sistem Informasi

Simbol	Nama	Keterangan
	Simbol proses komputerisasi	Menggambarkan proses yang dilakukan secara komputerisasi
	Simbol dokumen	Menggambarkan dokumen masukan dan keluaran system

	Simbol proses manual	Menggambarkan proses manual
	Simbol proses pengarsipan	Menggambarkan <i>file</i> yang diarsipkan menurut alfabet
	Simbol pengarsipan	Menggambarkan <i>file</i> yang diarsipkan menurut kronologi
	Simbol garis alir	Menggambarkan aliran proses dan dokumen
	Penghubung	Digunakan untuk menghubungkan sambungan aliran.
	Basis Data	Untuk media penyimpanan secara terkomputerisasi.

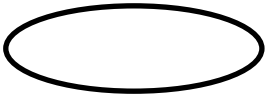
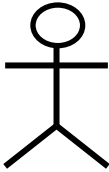

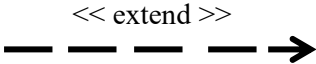
2.12 Unified Modeling Language (UML)


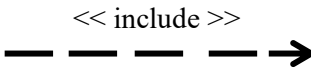
Menurut Muhamad Khoirul Fahmi (2024), *Unified Modeling Language* (UML) merupakan sebuah bahasa pemodelan yang digunakan untuk menggambarkan sistem atau perangkat lunak dengan paradigma berorientasi objek.

2.12.1 Use Case Diagram

Menurut Fitriya Nur Saidah (2024), *use case diagram* adalah *diagram* yang digunakan untuk mengidentifikasi kebutuhan sistem serta memahami bagaimana sistem seharusnya bekerja. *Diagram* ini menggambarkan interaksi antara pengguna (aktor) dan sistem melalui serangkaian skenario (*use case*) yang merepresentasikan fungsi atau layanan yang harus disediakan oleh sistem.

Tabel 2.2 Simbol – simbol pada *use case diagram*



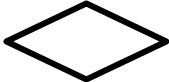


No	Simbol	Keterangan
	<i>Use Case</i> 	Fungsional yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau <i>actor</i> : biasanya dinyatakan dengan menggunakan kata kerja diawal <i>frase name use case</i> .
	<i>Actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri. Jadi walaupun simbol <i>actor</i> adalah gambar orang, biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama <i>actor</i> .
	Asosiasi 	Komunikasi antar <i>actor</i> dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> atau <i>usecase</i> memiliki interaksi dengan <i>actor</i> .
	<i>Extend</i> 	Perilaku <i>usecase</i> memperluas perilaku <i>usecase</i> yang lain.

	<p>Generalisasi</p> 	<p>Hubungan generalisasi dan spesialisasi antara dua buah <i>usecase</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya</p>
	<p>Include</p> 	<p>Perilaku <i>usecase</i> merupakan bagian dari <i>usecase</i> yang lain</p>

2.12.2 Activity Diagram

Menurut Sukarno Bahat Nauli (2024), diagram aktivitas (*Activity Diagram*) digunakan untuk menggambarkan aliran kerja (*workflow*) atau serangkaian aktivitas dalam sebuah sistem atau proses bisnis. Diagram ini sering digunakan untuk menunjukkan bagaimana suatu proses dijalankan secara berurutan, termasuk keputusan, interaksi, dan hasil dari berbagai aktivitas yang terjadi dalam sistem atau proses tersebut.


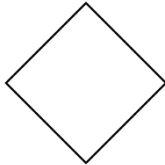
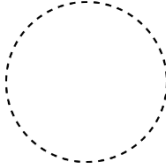
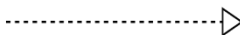
Tabel 2.3 Simbol – simbol pada *activity diagram*

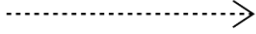

No	Gambar	Nama	Keterangan
1		<i>Initial Node</i>	Mengambarkan titik awal aktivitas.
2		<i>Activity</i>	Merupakan gambar aktivitas yang ada pada sistem.
3		<i>Decision</i>	Merupakan pilihan untuk pengambilan suatu keputusan.
4		<i>Join</i>	Penggabungan dimana lebih dari aktifitas.
5		<i>Activity Final Node</i>	Mengambarkan titik akhir aktifitas.

2.12.3 Class Diagram

Menurut Fauzul Muna (2023), *class diagram* digunakan untuk menjelaskan atribut, proses (metode), dan hubungan antar kelas serta menggambarkan struktur statis dari kelas dalam suatu sistem. *Diagram* ini merupakan salah satu *diagram* yang umum digunakan untuk membantu memahami struktur kelas dalam pengembangan sistem.

Tabel 2.4 Simbol – simbol pada *class diagram*

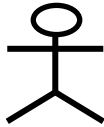

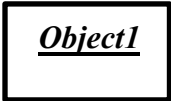

No	Gambar	Nama	Keterangan			
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendene</i>) berbagi perilaku dan struktur data dari objek yang ada diatas nya objek induk (<i>ancestor</i>).			
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.			
3	<table border="1"><tr><td>Nama Kelas</td></tr><tr><td>+atribut</td></tr><tr><td>-operasi()</td></tr></table>	Nama Kelas	+atribut	-operasi()	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
Nama Kelas						
+atribut						
-operasi()						
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .			
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.			

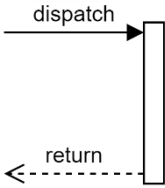
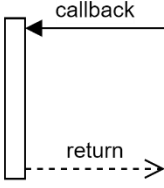
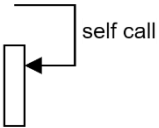
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Association</i>	Apa yang menghubungkan objek satu dan lainnya.

2.12.4 Sequence Diagram

Menurut Vivi Afifah (2021), *sequence diagram* digunakan untuk menggambarkan perilaku objek dalam suatu *use case*, dengan menunjukkan waktu hidup objek serta pesan yang dikirim dan diterima oleh objek tersebut. Jumlah *diagram* yang dibuat disesuaikan dengan jumlah *use case* yang memiliki proses masing-masing.

Tabel 2.5 Simbol – simbol pada *sequence diagram*

No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Mengambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3		Objek	Antar muka yang sering berinteraksi
4		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berintegrasi, semua yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang akan dilakukan di dalamnya.

5	 <pre> sequenceDiagram participant S participant O S->>O: dispatch O-->>S: return </pre>	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
6	 <pre> sequenceDiagram participant S participant O S->>O: callback O-->>S: return </pre>	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi.
7	 <pre> sequenceDiagram participant O O->>O: self call </pre>	<i>Self Message</i>	Spesifikasi dari komunikasi untuk menunjukan kegiatan memuat proses informasi pada aktifitas sendiri.