

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Pengertian Sistem

Sistem adalah adalah suatu jaringan kerja dari prosedur-prosedur yang saling berkaitan, dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai tujuan tertentu. Umumnya sistem terdiri atas tiga komponen utama *software, hardware, dan brainware* ketiga komponen ini saling berkaitan satu sama lain [5].

Software tidak memiliki bentuk fisik yang dapat disentuh atau dilihat, tetapi merupakan serangkaian instruksi atau prosedur yang digunakan untuk mengoperasikan komputer dan melakukan pemrosesan data sesuai dengan aturan yang telah ditentukan sebelumnya. *Hardware* adalah bagian dari komputer yang bisa dilihat dan disentuh secara langsung oleh manusia. Ini mencakup perangkat elektronik dan mekanik yang merupakan unsur penyusun komputer. *Hardware* dapat menjalankan tugas-tugas berdasarkan instruksi atau perintah yang telah ditentukan, yang biasanya disebut sebagai *instruction set*. *Brainware* adalah unsur yang berasal dari akal manusia. Apabila tidak ada manusia, komputer tidak bisa digunakan. *Brainware* juga berarti orang-orang yang bekerja secara langsung dengan menggunakan komputer sebagai alat bantu, ataupun orang-orang yang tidak bekerja secara langsung menggunakan komputer, tetapi menerima hasil kerja dari komputer yang berbentuk laporan [6].

Sistem memiliki karakteristik yaitu [5]:

1. Komponen sistem (*Component*)

Suatu sistem terdiri dari beberapa bagian komponen yang saling berinteraksi atau bekerja sama, artinya membentuk menjadi satu kesatuan. Komponen sistem terdiri dari subsistem atau bagian sistem.

2. Batasan sistem (*Boundary*)

Sistem adalah area yang memisahkan suatu sistem dengan sistem lainnya dengan menggunakan lingkungan eksternal. Keterbatasan sistem ini memungkinkan satu sistem dianggap menjadi suatu kesatuan. Batasan suatu sistem memperlihatkan ruang lingkup (*scope*) menurut sistem tersebut.

3. Lingkungan luar sistem (*Environment*)

Segala yang di luar batas sistem mempengaruhi operasi sistem. Lingkungan luar dapat bermanfaat bagi sistem jika dipertahankan dan dapat juga merugikan sistem tersebut.

4. Penghubung sistem (*Interface*)

Penghubung sistem adalah merupakan media koneksi dari satu subsistem ke subsistem lainnya. Melalui penghubung ini memungkinkan sumber daya mengalir dari subsistem ke subsistem lainnya. *Output* subsistem tersebut akan diteruskan ke subsistem lainnya melalui sebuah *link* atau penghubung.

5. Masukkan Sistem (*Input*)

Input adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa *maintenance input* atau pemeliharaan dan *signal input*. Contohnya adalah *maintenance input* digunakan untuk mengoperasikan komputer sedangkan data adalah sinyal *input* yang diubah menjadi informasi.

6. Keluaran sistem (*Output*)

Output dari sistem adalah hasil dari energi yang diproses dan diklasifikasikan menjadi *output* yang bermanfaat, contoh pada komputer adalah menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah sistem

Suatu sistem menjadi bagian pemrosesan yang mengubah *input* menjadi *output*. Sistem produksi akan mengubah bahan baku menjadi produk jadi dan sistem akuntansi mengubah data menjadi laporan keuangan.

8. Sasaran sistem

Suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (*objective*). Tujuan sistem menentukan *input* yang dibutuhkan oleh sistem dan *output* yang dihasilkan oleh sistem.

2.1.2 Pengertian informasi

Informasi adalah data yang telah diproses menjadi bentuk yang lebih berguna dan bermakna bagi penerimanya disebut informasi. Sumber informasi berasal dari data, yaitu informasi yang menggambarkan kejadian-kejadian dan kesatuan nyata dalam kehidupan. Fungsi utama informasi adalah menambah pengetahuan. Informasi yang disampaikan kepada pemakai mungkin merupakan hasil data yang sudah diolah untuk menanggapi seleksi kepada pengambil keputusan menjadi sebuah keputusan [7].

Karakteristik informasi yang baik, yaitu sebagai berikut [6]:

1. *Information must be pertinent.* Artinya, informasi harus berhubungan. Pernyataan informasi harus berhubungan dengan urusan dan masalah yang penting bagi penerima informasi (orang yang membutuhkan informasi).
2. *Information must be accurate.* Informasi harus bebas dari kesalahan dan tidak memiliki bias atau menyesatkan. Informasi yang dihasilkan harus mencerminkan maksudnya. Keakuratan informasi bergantung pada keadaan.
3. *Information must be timely.* Informasi harus ada ketika dibutuhkan. Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang telah usang tidak akan mempunyai nilai karena informasi merupakan landasan dalam pengambilan keputusan.
4. Relevan. Artinya, informasi mempunyai manfaat untuk pemakainya.

2.1.3 Sistem Informasi

Sistem informasi adalah sekumpulan komponen yang saling berhubungan yang mengumpulkan (mengambil), memproses, menyimpan, dan berbagi informasi untuk mendukung pengambilan keputusan dan pengendalian dalam suatu organisasi. Selain itu, sistem informasi membantu manajer menyelidiki masalah, memvisualisasikan masalah yang kompleks, dan mengembangkan produk baru. Sebuah sistem informasi dapat digambarkan sebagai satu set proses formal di mana informasi dikumpulkan [8].

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*Building block*) yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, yaitu [7]:

1. Blok masukan (*Input block*)

Input menyatakan data yang dimasukkan ke sistem informasi. *Input* meliputi cara dan alat yang digunakan untuk mengumpulkan data, seperti dokumen dasar.

2. Blok model (*Model block*)

Blok ini mengkombinasikan prosedur, logika, dan metode matematik untuk memproses data *input* dan data yang disimpan dalam basis data dengan cara yang telah ditentukan, sehingga menghasilkan *output* yang diinginkan.

3. Blok keluaran (*Output block*)

Output dari sistem informasi adalah informasi berkualitas dan dokumentasi yang berguna bagi semua tingkatan manajemen dan semua pengguna sistem.

4. Blok teknologi (*Technologi block*)

Teknologi bertujuan untuk memproses informasi melalui berbagai tahapan seperti menerima *input*, melakukan pemrosesan, menyimpan dan mengambil data, menghasilkan dan mengirimkan *output* dan membantu kontrol diri secara keseluruhan.

Teknologi terdiri dari unsur utama :

- a. Teknisi (*Humanware* atau *Brainware*)
- b. Perangkat lunak (*Software*)
- c. Perangkat keras (*Hardware*)

5. Blok basis data (*Database block*)

Ini adalah kumpulan data yang saling berkaitan satu dengan lainnya yang disimpan pada perangkat keras komputer dan diolah dengan bantuan perangkat lunak.

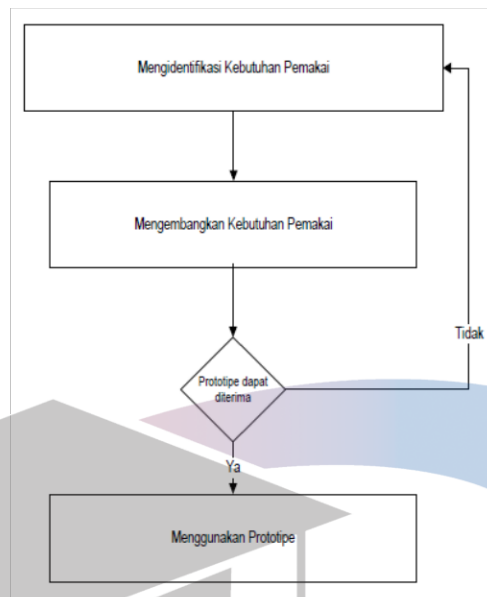
6. Blok kendali (*Control block*)

Ada banyak hal yang dapat menimbulkan kerusakan pada sistem informasi, seperti bencana alam, kebakaran, temperatur tinggi, air, debu, tindakan curang, masalah pada sistem itu sendiri, kesalahan efisiensi, sabotase dan lain sebagainya. Beberapa tindakan pencegahan perlu diterapkan untuk memastikan bahwa kerusakan pada sistem dapat dicegah atau dapat segera diperbaiki jika sudah terlanjur terjadi.

2.2 Prototyping

Prototype berfungsi sebagai sarana untuk meningkatkan persyaratan perangkat lunak untuk pengembangan, dengan iterasi yang dibuat saat *prototype* disesuaikan untuk memenuhi kebutuhan pelanggan dan membantu pengembang mendapatkan pemahaman yang lebih jelas tentang apa yang harus diselesaikan. Dalam skenario yang ideal, *prototype* dapat berfungsi sebagai alat untuk menentukan kebutuhan perangkat lunak [9]. Namun, terdapat kelemahan pada pembuatan *prototype* yaitu keputusan penting dapat dibuat terlalu dini sebelum memahami masalah bisnis atau teknologi informasi secara menyeluruh. Oleh karena itu, penting untuk menggunakan teknik pemodelan yang cermat dan mencari fakta sebelum membuat *prototype*. Apabila *prototype* dibuat dengan baik, maka dapat menjadi alat yang sangat berharga [10].

2.2.1 Tahapan Metode *Prototyping*



Gambar 2. 1 Tahapan Metode *Prototyping*

Berikut ini adalah penjelasan dari gambar tahapan - tahapan metode *prototype* [11]:

1. Mengidentifikasi kebutuhan dan analisis sistem

Tahapan pertama yang digunakan dalam penelitian ini adalah pengumpulan kebutuhan dan analisis sistem. Pada tahapan ini dilakukan identifikasi kebutuhan sistem dan garis besar dari sistem yang akan dibuat. Dalam mengembangkan sistem informasi tugas akhir ini, diperlukan biodata mahasiswa dan judul dari tugas akhir.

2. Mengembangkan kebutuhan dan Pembentukan *prototype*

Dalam tahap ini, dilakukan pembentukan *prototype* berdasarkan rancangan pemodelan yang telah dilakukan sebelumnya.

3. Evaluasi *prototype*

Pada tahapan ini, dilakukan evaluasi terhadap *prototype* yang disesuaikan dengan kebutuhan. Jika belum sesuai dengan kebutuhan, maka dapat melakukan tahap selanjutnya yaitu melakukan perubahan *prototype*.

4. Perubahan *prototype*

Tahapan ini dilakukan untuk menyempurnakan *prototype* yang dibangun agar menghasilkan *prototype* yang sesuai dengan kebutuhan.

5. Penggunaan *prototype*

Tahap terakhir dari metode penelitian ini adalah penggunaan sistem. Pada tahap ini sistem yang telah dievaluasi siap untuk digunakan.

Prototype Evaluation Form					
Observer Name	Michael Cervris			Date	1/06/2010
System or Project Name	Cloud Computing Data Center		Company or Location		Aquarius Water Filters
Program Name or Number	Prev. Maint.	Version		1	
	User 1	User 2	User 3	User 4	
User Name	Andy H.	Pam H.			
Period Observed	1/06/2010	1/06/2010			
User Reactions	Generally favorable, got excited about project.	Excellent!			
User Suggestions	Add the date when maintenance was performed.	Place a form number on top for reference. Place word WEEKLY in title.			
Innovations					
Revision Plans	Modify on 1/08/2010. Review with Andy and Pam.				

Gambar 2. 2 *Prototype Evaluation Form*

Peran pengguna dalam pembuatan *prototype* dapat diringkas dalam dua kata: keterlibatan yang jujur. Tanpa pengguna keterlibatan ada sedikit alasan untuk membuat *prototype*. Perilaku yang tepat diperlukan untuk berinteraksi dengan *prototype* dapat bervariasi, tetapi jelas bahwa pengguna sangat penting dalam proses pembuatan *prototype*. Menyadari pentingnya pengguna untuk keberhasilan proses, anggota analisis sistem tim harus mendorong dan menerima masukan dan menjaga dari penolakan alami mereka sendiri untuk mengubah *prototype*.

Ada tiga cara utama pengguna dapat membantu dalam pembuatan *prototype* [12] :

1. Bereksperimen dengan *prototype*.
2. Memberikan reaksi terbuka terhadap *prototype*.
3. Menyarankan penambahan atau penghapusan dari *prototype*.

Aspek ketiga dari peran pengguna dalam pembuatan *prototype* adalah kesediaan mereka untuk menyarankan penambahan atau penghapusan dari fitur yang dicoba. Peran analis adalah untuk mendapatkan saran tersebut dengan meyakinkan pengguna bahwa umpan balik yang mereka berikan ditanggapi dengan serius, dengan mengamati pengguna saat mereka berinteraksi sistem, dan dengan melakukan wawancara singkat dan spesifik

dengan pengguna mengenai pengalaman mereka dengan *prototype*. Meskipun pengguna akan diminta untuk mengartikulasikan saran dan inovasi untuk *prototype*, pada akhirnya adalah tanggung jawab analis untuk menimbang umpan balik ini dan menerjemahkannya ke dalam perubahan yang dapat diterapkan jika diperlukan. Untuk memudahkan proses *prototyping*, analis harus jelas mengkomunikasikan tujuan pembuatan *prototype* kepada pengguna, bersama dengan gagasan bahwa pembuatan *prototype* itu berharga hanya ketika pengguna terlibat secara bermakna.

2.2.2 Unified Modeling language

Unified Modeling Language atau UML di definisikan sebagai bahasa pemodelan tujuan umum standar di bidang rekayasa perangkat lunak berorientasi objek [13]. UML adalah alat yang digunakan untuk menentukan dan memvisualisasikan, memodifikasi, membangun, dan mendokumentasikan artefak dari sistem kompleks atau intensif perangkat lunak, menghasilkan artefak yang berorientasi objek pada saat dikembangkan [14].

Berikut jenis - jenis diagram pada UML (*Unified Modeling Language*) yaitu [15]:

1. *Use case Diagram*
2. *Activity diagram*
3. *Sequence Diagram*
4. *Class Diagram*


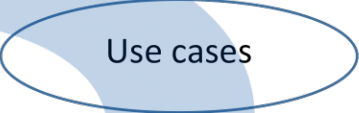



2.3 Teknik Perancangan sistem


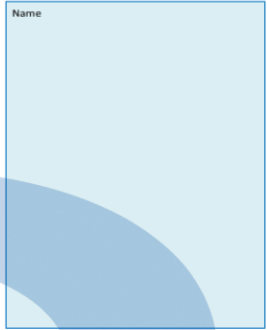
2.3.1 Use case Diagram

Use case menggambarkan *external view* dari sistem yang akan dibuat modelnya. Model *use case* dapat dijabarkan dalam diagram *use case*, penting untuk dicatat bahwa diagram tidak sepenuhnya mencakup model karena lebih komprehensif [16]. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [17].

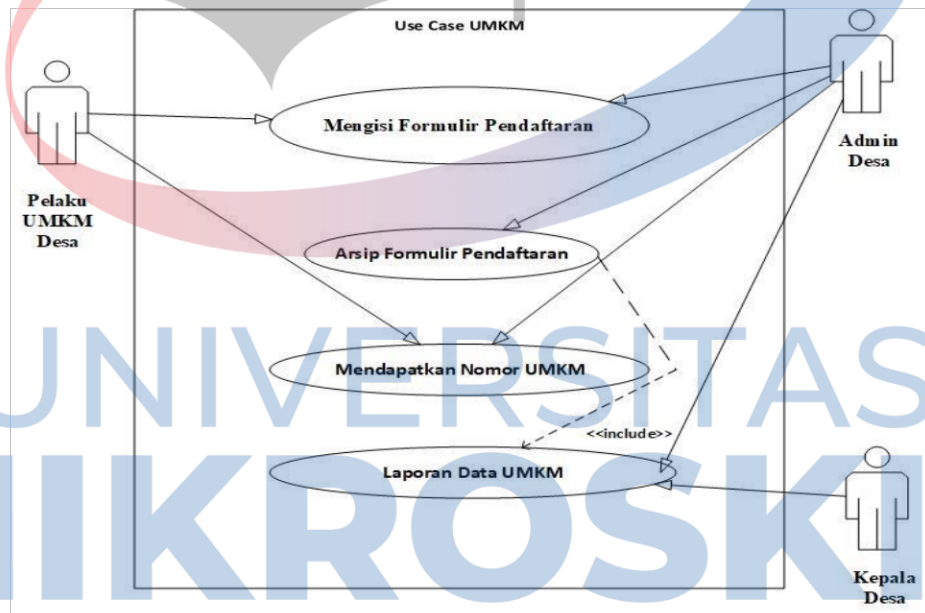
Berikut adalah penjelasan simbol pada *use case diagram* [18]:

Tabel 2. 1 Simbol *Use Case Diagram*

Notasi	Fungsi	Shape
<i>Actor</i>	Seseorang atau sekelompok orang di luar sistem yang berperan dalam satu atau lebih interaksi dengan sistem, itu mewakili dari mana informasi berasal dan ke mana ia pergi.	
<i>Use Case</i>	Menunjukkan semua fungsi sistem.	
Generalisasi	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum-khusus).	
<i>Association/Asosiasi</i>	Menunjukkan hubungan antara <i>use case</i> dan <i>actor</i> mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.	
Asosiasi Berarah	Merupakan hubungan antara <i>use case</i> dan <i>actor</i> mengindikasikan bila <i>actor</i> berinteraksi pasif dengan sistem.	
<i>Include</i>	<i>Include</i> merupakan relasi tambahan sebuah <i>use case</i> dimana memerlukan <i>use case</i> lain untuk menjalankian fungsinya.	

<p><i>Extends</i></p>	<p><i>Extend</i> merupakan relasi tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambah kan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.</p>	
<p><i>System Boundary Box</i></p>	<p>Merupakan batasan sistem dalam <i>use case diagram</i>, fungsinya menggambarkan sebuah kotak yang melingkupi semua <i>use case</i>, namun <i>actor</i> tetap berada diluar.</p>	

Berikut adalah contoh *Use Case diagram* [19]:








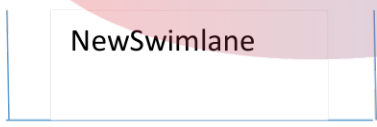


Gambar 2. 3 *Use Case Diagram UMKM*

2.3.2 *Activity Diagram*

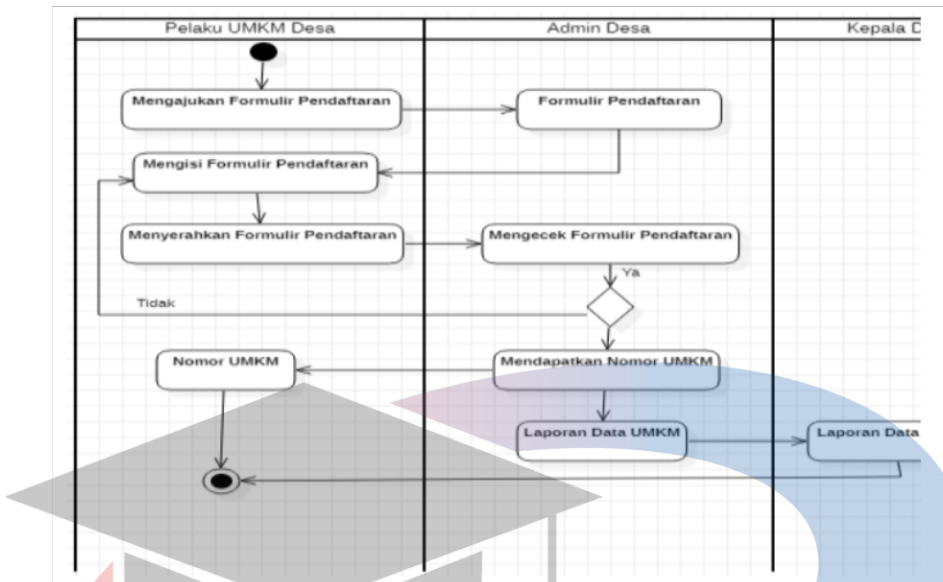
Activity diagram menunjukkan urutan aktivitas yang terjadi dalam suatu proses, yang terdiri dari aktivitas linier dan paralel, serta pilihan yang dibuat. Biasanya, diagram aktivitas dikembangkan untuk kasus penggunaan individu dan dapat menggambarkan berbagai kemungkinan skenario[12] .

Activity Diagram terdiri dari beberapa simbol, yaitu sebagai berikut [18]:

Tabel 2. 2 Simbol *Activity Diagram*

No	Simbol	Keterangan	Definisi
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain.
2.		<i>Action</i>	<i>State</i> dari suatu sistem yang mencerminkan eksekusi dari suatu aksi.
3.		<i>Initial Node/Start point</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Activity Final Node/End point</i>	Bagaimana objek dibentuk dan dihancurkan maksudnya sebuah <i>Activity diagram</i> memiliki sebuah status akhir.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
7.		<i>State Tansision</i>	Aliran dan aktifitas.
8.		<i>Decision/Per cabang</i>	Cabang keluaran dari <i>conditon</i> dapat lebih dari dua, tetapi biasanya sebagian besar hanya berisi dua keluaran biner.

Berikut adalah contoh *Activity diagram* [19]:



Gambar 2. 4 *Activity Diagram* UMKM


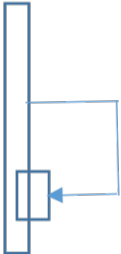
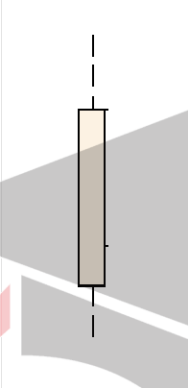
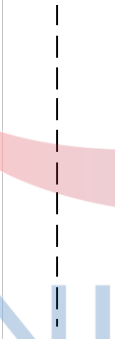


2.3.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar *class* selama durasi tertentu dan digunakan sebagai model dinamis dari *use case*. *Sequence diagram* mendokumentasikan kasus penggunaan secara visual dengan menampilkan pesan, kelas pesan, dan waktu [10].

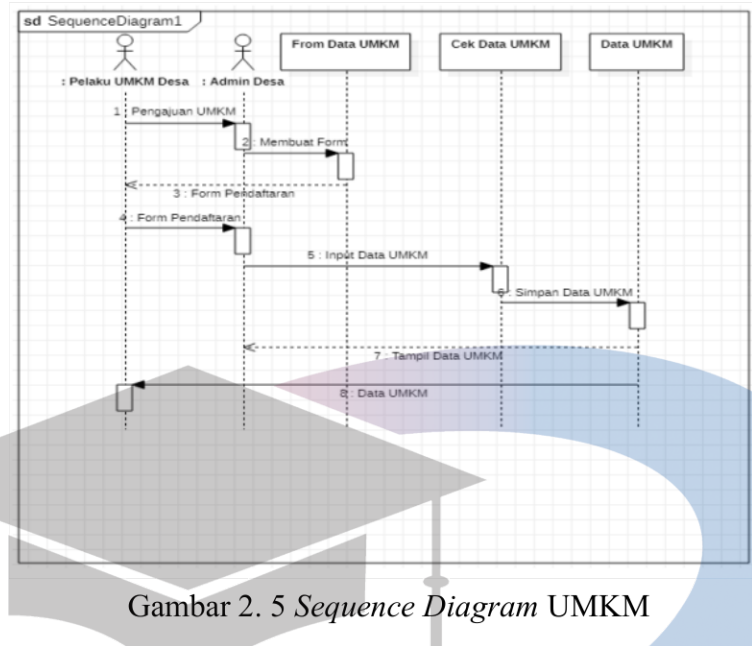
Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu [18]:

Tabel 2. 3 Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.

	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>
	<p><i>Actor</i> adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.</p>
	<p><i>Object</i>, merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan secara horizontal untuk menyatakan objek yang berinteraksi.</p>

Berikut adalah contoh *Sequence diagram* [19]:



Gambar 2. 5 *Sequence Diagram* UMKM

2.3.4 *Class Diagram*


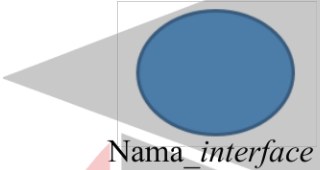



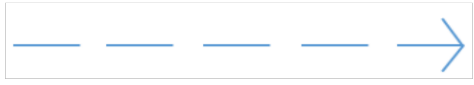
Model logis, yang dikenal sebagai *Class diagram*, berkembang menjadi model fisik dan akhirnya berubah menjadi sistem informasi fungsional. Dalam analisis terstruktur, entitas, penyimpanan data, dan proses diubah menjadi kode program dan struktur data. Demikian pula, diagram kelas maju ke dalam komponen sistem seperti modul kode dan objek data [10].


Class diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut [15].

UNIVERSITAS
MIKROSKIL

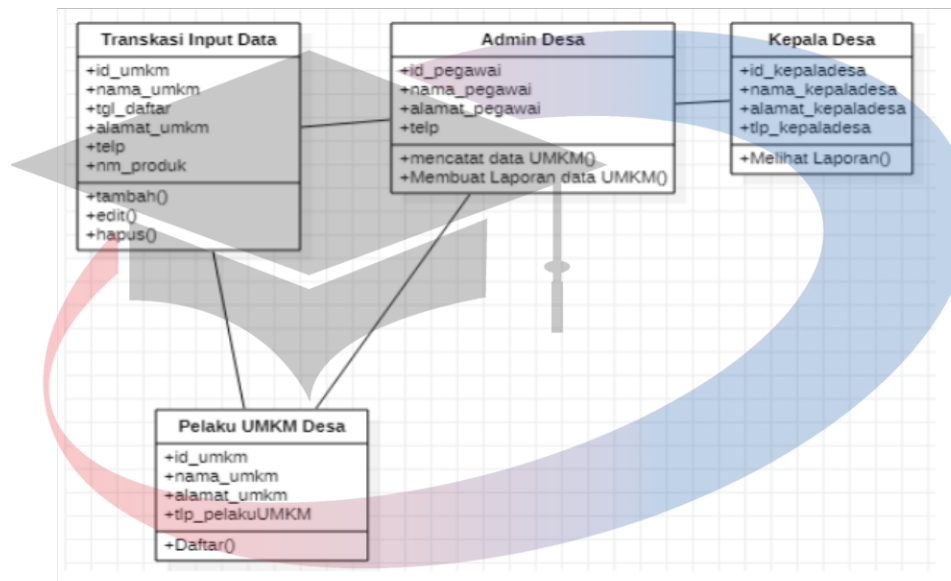
Simbol-simbol yang digunakan dalam *class diagram* yaitu [17]:

Tabel 2. 4 Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Antarmuka/<i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi/<i>Asociation</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi Berarah / <i>Directed Asociation</i></p> 	Kelas antar kelas dengan kelas yang satu digunakan oleh kelas yang lain, asosisasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan/<i>Depedency</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.

<p>Agregasi/Aggregation</p> 	<p>Relasi antar kelas dengan makna semua-bagian (<i>whole part</i>).</p>
---	--

Berikut adalah contoh Class *diagram* [19]:



Gambar 2. 6 Class Diagram UMKM

2.4 Basis data

Dalam dunia komputer, basis data merujuk pada kegiatan sistem program komputer yang digunakan untuk berbagai aplikasi. Untuk menyimpan data, basis data memerlukan media simpan komputer yang terorganisir dengan baik, serta perawatan data yang baik dalam fungsi manajemen sistem. Secara keseluruhan, basis data merupakan pengetahuan tentang cara mengorganisir data, dan *database* menjadi salah satu komponen penting dalam sistem informasi. Sistem basis data (*database system*) adalah penerapan *database* dalam sistem informasi [7].

Istilah yang terdapat dalam basis data, yaitu [20]:

1. Entitas

Entitas adalah objek yang dapat dibedakan dalam dunia nyata kumpulan dari entitas disebut dengan himpunan entitas contoh entitas yaitu :

- a. Entitas berupa objek secara fisik: rumah, kendaraan, peralatan
- b. Entitas objek secara konsep: pekerjaan, perusahaan, rencana.

2. Atribut/*Field*

Merupakan karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan *detail* tentang *entity* atau *relationship* tersebut. Berikut beberapa jenis atribut yang sering dipakai, yaitu:

- a. Atribut kunci (*Key*): atribut kunci merupakan atribut utama yang digunakan untuk menentukan data. Pada umumnya, data dari atribut *Key* berbentuk numerik. Contohnya nomor KTP, SIM, MPWP.
- b. Atribut sederhana : atribut sederhana adalah atribut yang tidak dapat dibagi lagi yang tidak dapat diuraikan lebih lanjut menjadi atribut lainnya. Contoh entitas mahasiswa mempunyai atribut sederhana berupa NIM dan nama mahasiswa.
- c. Atribut komposit : atribut yang dapat didekomposisi lebih lanjut menjadi sub-atribut yang lebih kecil, masing-masing memiliki signifikansi yang berbeda. Contoh : entitas mahasiswa mempunyai atribut alamat. Alamat disini dapat dipecah menjadi sub atribut seperti nama_kota, kode_pos.
- d. Atribut bernilai tunggal : yaitu setiap baris hanya dapat berisi satu nilai untuk atribut. Contoh : entitas mahasiswa mempunyai atribut NPM, nama, alamat isi data dari atribut ini hanya boleh diisi dengan satu data. Setiap mahasiswa hanya memiliki satu NPM, satu nama, satu alamat.
- e. Atribut bernilai jamak (*Multivalue*) : yaitu atribut dapat memiliki beberapa nilai untuk setiap baris. Contoh : entitas mahasiswa mempunyai atribut hobi, isi data dari atribut ini boleh lebih dari satu data. Mahasiswa rusdi memiliki NPM 13402021 beralamat di Jalan Garuda 32 Medan memiliki hobi (olah raga, nyanyi, masak dan nonton tv)
- f. Atribut harus bernilai atau *not null*: yaitu mengacu pada atribut wajib yang membutuhkan nilai data untuk setiap baris. Biasanya, atribut seperti itu disertakan dalam desain tabel sedemikian rupa sehingga kesalahan akan dihasilkan jika nilai wajib

tidak diberikan saat mengisi data untuk baris tersebut. Contoh : entitas mahasiswa mempunyai atribut NPM dan Nama_Mahasiswa yang harus diisi datanya, sebab jika tidak diisi akan terjadi kekacauan dalam basis data.

- g. Atribut bernilai null : yaitu atribut yang belum tentu memiliki nilai data untuk setiap baris. Contoh : entitas mahasiswa mempunyai atribut alamat, hobi, nama_pacar yang boleh untuk tidak diisi tetapi kalau diisi akan lebih baik,
- h. Atribut turunan (*Derived*) : yaitu atribut yang nilainya diperoleh melalui beberapa bentuk pemrosesan atau dapat dihitung berdasarkan nilai atribut terkait lainnya. Contoh : entitas mahasiswa mempunyai atribut “IPK” yang diperoleh dari pengolahan atribut nilai pada tabel (entitas nilai) dengan kode NIM mahasiswa yang sama dan diproses sehingga menghasilkan IPK untuk mahasiswa yang bersangkutan.

3. *Record*

Record adalah beberapa unsur data yang dikelompokkan secara logis, misalnya yang berkaitan dengan objek data tertentu. Misalnya ketiga unsur data nama mahasiswa, nomor induk mahasiswa, matakuliah yang diambil digabungkan, maka terbentuklah satu *record*. *Record* ini dokumen berisi sekumpulan komponen informasi yang menggambarkan karakteristik suatu atribut dari entitas (*entity*) mahasiswa. *Record* dapat diorganisasikan dengan cara :

- a. *Record* sebuah catatan dapat dikenali dengan memanfaatkan satu atau banyak petunjuk, yang juga disebut sebagai kunci. Kunci adalah informasi tertentu dalam catatan yang digunakan untuk mengidentifikasi catatan itu dalam file. Misalnya *record* tentang mahasiswa dapat diidentifikasi melalui nomor induk mahasiswa.
- b. Suatu *record* dapat meng-enkapsulasi semua komponen data yang signifikan atau dapat dikategorikan ke dalam *master record* dan *trailer record*. *Master record* mungkin termasuk rincian seperti nama siswa, alamat, tanggal lahir, dan pekerjaan orang tua mereka, antara lain. Sedangkan *trailer record* mungkin berisi daftar nilai mata kuliah yang sudah ditempuh, IPK, KRS, daftar mata kuliah yang belum ditempuh dan sebagainya.

Tujuan efektivitas dari sebuah basis data adalah sebagai berikut [12]:

1. Memastikan data dapat dibagikan di antara pengguna untuk berbagai aplikasi.
2. Mempertahankan data tetap akurat dan konsisten.
3. Memastikan semua data yang dibutuhkan untuk aplikasi saat ini dan kedepannya akan selalu tersedia.
4. Memungkinkan basis data untuk berkembang seiring dengan peningkatan kebutuhan pengguna.
5. Memungkinkan pengguna untuk membangun pandangan pribadi mengenai data tanpa memperhatikan cara data disimpan secara fisik.

Basis data relasional merupakan suatu cara mengatur data secara fisik kedalam memori melalui penggunaan basis data relasional. *Database* ini menggunakan tabel dengan baris dan kolom untuk mewakili *file* data dalam format dua dimensi [21]. Beberapa istilah yang terdapat dalam basis data relasional, yaitu [21]:

1. Relasi, adalah sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris.
2. Atribut, adalah kolom pada sebuah relasi.
3. Tupel, adalah baris pada sebuah relasi.
4. *Domain*, adalah kumpulan nilai yang valid untuk satu atau lebih atribut.
5. Derajat (*Degree*), adalah jumlah atribut dalam sebuah relasi (jumlah *field*).
6. *Cardinality*, yaitu jumlah tupel dalam sebuah relasi (jumlah *record*).

Model relasional juga terdapat *relational key* yang digunakan untuk mengurangi waktu yang diperlukan untuk menelusuri semua data, berikut adalah jenis-jenis *relational key* dalam model relasional [21]:

1. *Super Key*, adalah sebuah atau sekelompok atribut yang membedakan satu baris yang lain dalam tabel relasional.
2. *Candidate Key*, adalah kemungkinan suatu hubungan memiliki banyak kunci, dan masing-masing kunci ini dikenal sebagai *candidate key*.
3. *Primary Key*, adalah *candidate key* yang dipilih sebagai pengenal yang berbeda untuk sebuah tabel relasi.
4. *Alternate Key*, adalah suatu *schema* relasi dapat memiliki atribut yang menunjuk ke *primary key* dari relasi lain. Atribut ini disebut *foreign key*.

2.5 Usaha Mikro, Kecil, dan Menengah (UMKM)

Usaha Mikro Kecil dan Menengah (UMKM) merupakan jenis usaha yang dapat membuka lapangan kerja yang luas dan memberikan manfaat ekonomi yang luas bagi masyarakat. Peran UMKM sangat penting dalam mencapai tujuan pemerataan pendapatan dan pertumbuhan ekonomi masyarakat, serta berkontribusi untuk mencapai stabilitas nasional [22].

Definisi UMKM (Usaha Mikro, Kecil, Menengah) Berdasarkan Undang-Undang No. 20 tahun 2008 tentang UMKM ada beberapa kriteria yang dipergunakan untuk mendefinisikan pengertian dan kriteria Usaha Mikro, Kecil dan Menengah. Pengertian-pengertian UMKM tersebut adalah [23]:

1. Usaha Mikro

Kriteria kelompok Usaha Mikro adalah usaha produktif milik orang perorangan atau badan usaha perorangan yang memenuhi kriteria Usaha Mikro sebagaimana diatur dalam Undang-Undang ini:

- a. Memiliki kekayaan bersih paling banyak Rp. 50.000.000,00 (lima puluh juta rupiah) tidak termasuk tanah dan bangunan tempat usaha; atau
- b. Memiliki hasil penjualan tahunan paling banyak Rp. 300.000.000,00 (tiga ratus juta rupiah).

2. Usaha Kecil

Kriteria 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 sebagaimana dimaksud dalam Undang-Undang ini:

- a. 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) tidak termasuk tanah dan bangunan tempat usaha; atau
- b. Memiliki hasil penjualan tahunan lebih dari Rp.300.000.000 (tiga ratus juta rupiah) sampai dengan paling banyak Rp. 2.500.000.000 (dua milyar lima ratus juta rupiah).

3. Usaha Menengah

Kriteria Usaha Menengah adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perseorangan 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 sebagaimana diatur dalam Undang-Undang ini:

- a. Memiliki kekayaan bersih lebih dari Rp. 500.000.000 (lima ratus juta rupiah) sampai dengan paling banyak Rp. 10.000.000.000 (sepuluh milyar rupiah), atau
- b. Memiliki hasil penjualan tahunan lebih dari Rp. 2.500.000.000 (dua milyar lima ratus juta rupiah) sampai dengan paling banyak Rp. 50.000.000.000 (lima puluh milyar rupiah).

Kriteria usaha mikro menurut Pasal 6 Undang-Undang Republik Indonesia No. 20 tahun 2008 menyebutkan bahwa kriteria usaha mikro adalah sebagai berikut [24]:

- a. Memiliki kekayaan bersih paling banyak Rp50.000.000,00 (lima puluh juta rupiah) tidak termasuk tanah dan bangunan tempat usaha; atau
- b. Memiliki hasil penjualan tahunan paling banyak Rp300.000.000,00 (tiga ratus juta rupiah).

Ciri-Ciri Usaha Mikro

- a. Jenis usaha/komoditi usahanya tidak selalu tetap, sewaktu-waktu dapat berganti.
- b. Tempat usahanya tidak selalu menetap, sewaktu-waktu dapat pindah tempat.
- c. Belum melakukan administrasi keuangan yang sederhana sekalipun, dan tidak memisahkan keuangan keluarga dengan keuangan usahanya.
- d. Sumber daya manusianya (pengusahanya) belum memiliki jiwa wirausaha yang memadai.
- e. Umumnya belum pernah mengakses kepada perbankan, namun sebagian dari mereka sudah mengakses ke lembaga keuangan non bank (bank titil dan semacamnya).
- f. Umumnya tidak memiliki izin usaha atau persyaratan legalitas lainnya termasuk NPWP [24].

Dari beberapa pengertian diatas dapat disimpulkan bahwa UMKM merupakan suatu bentuk usaha ekonomi produktif yang dilakukan oleh orang perseorangan atau pun badan usaha perorangan yang mewajibkan iuran kepada negara dan harus dibayarkan oleh wajib pajak berdasarkan undang-undang yang telah ditetapkan dengan tidak mendapatkan imbalan secara langsung, tetapi digunakan untuk pengeluaran-pengeluaran negara dan pembangunan nasional yang dinikmati secara bersama-sama.

2.6 Pencatatan Penjualan UMKM

Pencatatan merupakan pengumpulan data melalui cara mencatat yang dapat menghasilkan informasi yang ter-integrasi, dapat diartikan sebagai pencatatan [25]. Fungsi pencatatan, artinya semua transaksi yang terjadi berdasarkan bukti dokumen yang ada harus dicatat seluruhnya [26].

Pencatatan transaksi adalah langkah awal untuk mencatat berbagai perubahan posisi keuangan pada sebuah UMKM yang dilakukan secara kronologis, dengan metode-metode tertentu sehingga hasil pencatatan dapat di lihat [27]. Beberapa cakupan mengenai pencatatan dalam tugas akhir ini adalah [27]:

1. Pencatatan kas masuk: yaitu suatu kegiatan pencatatan transaksi keuangan yang berkaitan dengan penerimaan uang atau kas dari berbagai sumber ke dalam entitas bisnis atau organisasi seperti UMKM. Contoh pencatatan kas masuk antara lain adalah penerimaan uang dari penjualan barang atau jasa, penerimaan uang dari peminjaman, penerimaan uang dari pelanggan yang membayar tagihan, dan penerimaan uang dari sumber lain seperti investasi atau sumbangan.
2. Pencatatan kas keluar: yaitu pengeluaran uang atau kas yang dikeluarkan oleh suatu UMKM dalam menjalankan kegiatan usahanya. Contoh pencatatan kas keluar antara lain adalah pembayaran gaji karyawan, pembayaran tagihan listrik, air, telepon, sewa dan pembayaran utang.
3. Pencatatan penjualan: UMKM harus mencatat semua penjualan yang dilakukan, termasuk jumlah barang atau jasa yang terjual, harga jual, tanggal penjualan, dan identitas pembeli.
4. Pencatatan pembelian: UMKM juga harus mencatat semua pembelian barang dan jasa yang dilakukan. Hal ini dapat membantu dalam mengawasi biaya operasional, serta memudahkan dalam menghitung keuntungan dan rugi.
5. Pencatatan utang: yaitu pencatatan transaksi keuangan yang berkaitan dengan uang yang harus dibayar oleh suatu entitas bisnis atau organisasi lebih tepatnya kepada *supplier*. Dalam pencatatan utang, dicatat jumlah utang, tanggal jatuh tempo pembayaran, dan pihak yang akan menerima pembayaran. Pencatatan utang sangat penting untuk menghindari terjadinya kesalahan dalam melakukan pembayaran dan menghindari denda atau sanksi lainnya yang mungkin timbul akibat keterlambatan pembayaran.