

## BAB II TINJAUAN PUSTAKA

### 2.1 Teori Dasar

#### 2.1.1 Manajemen Pengetahuan

Manajemen Pengetahuan (*Knowledge Management*) merupakan kegiatan/aktivitas yang dilakukan oleh suatu organisasi untuk mencari, mengidentifikasi, menjelaskan, mengembangkan, dan menyebarkan pengetahuan yang dapat digunakan kembali untuk dipelajari dan dimanfaatkan oleh Sumber Daya Manusia (SDM) organisasi tersebut. Hal ini dapat dimanfaatkan untuk meningkatkan kualitas dan kinerja dalam mencapai tujuan organisasi (Hendrawan, 2019). Manajemen pengetahuan dapat dibagi menjadi beberapa komponen, yaitu *Technology*, *People*, dan *Process*. Berikut ini adalah penjelasan dari tiga komponen tersebut:

##### a. *People* (Orang)

Seorang individu yang terdapat pada suatu organisasi dapat menyimpan dan menyebarkan pengetahuan yang menjadi salah satu faktor suksesnya dalam penerapan manajemen pengetahuan. Pengetahuan yang dimiliki oleh SDM di dalam organisasi dapat dijadikan *strategic knowledge* yang akan berdampak pada keberhasilan organisasi dalam mencapai target atau tujuan (Hendrawan, 2019).

##### b. *Process* (Proses)

Proses dalam manajemen pengetahuan merupakan upaya membantu dalam mengubah *tacit* menjadi *explicit knowledge* (externalisasi pengetahuan). Proses memastikan bahwa manajemen pengetahuan diterapkan atau diimplementasikan dengan baik di organisasi tersebut (Hendrawan, 2019). Terdapat 6 proses inti dalam manajemen pengetahuan, yakni sebagai berikut:

1. *Knowledge Identification* (Identifikasi Pengetahuan) merupakan proses mengidentifikasi pengetahuan yang dapat berupa *tacit* atau *explicit knowledge* (Hendrawan, 2019).

2. *Knowledge Acquisition* (Pengadaan Pengetahuan) merupakan proses untuk menambah atau mengadakan pengetahuan yang dapat diperoleh dari organisasi lain atau memanggil ahli pada bidang yang diperlukan (Hendrawan, 2019).

3. *Knowledge Development* (Pengembangan Pengetahuan) merupakan proses yang berfokus pada pengembangan pengetahuan seperti ide atau produk baru, kompetensi SDM, peningkatan efisiensi, dan lain – lain (Hendrawan, 2019).

4. *Knowledge Distribution* (Distribusi Pengetahuan) merupakan proses untuk mendistribusikan pengetahuan di dalam ruang lingkup internal suatu organisasi yang dapat dimanfaatkan untuk mengembangkan SDM melalui *knowledge transfer* antara individu – individu di dalam organisasi (Hendrawan, 2019). Salah satu media perantara yang dapat digunakan untuk distribusi pengetahuan *tacit* dan *explicit* adalah secara tertulis dengan bentuk yaitu artikel. Wadah distribusi pengetahuan tersebut dapat dimanfaatkan oleh suatu organisasi untuk menyebarkan informasi/pengetahuan penting karena dapat digunakan dalam eksternalisasi pengetahuan (Mengubah *Tacit Knowledge* menjadi *Explicit Knowledge*) (Sawan, 2021).

5. *Knowledge Utilization* (Utilisasi Pengetahuan) merupakan proses untuk memastikan bahwa pengetahuan disimpan dengan baik dan dapat digunakan kembali untuk mendukung proses bisnis di dalam organisasi (Hendrawan, 2019).

6. *Knowledge Retention* (Penyusutan Pengetahuan) merupakan proses yang digunakan untuk menyusutkan, memilih, menyimpan, dan memperbaharui pengetahuan yang sangat bernilai bagi

organisasi tersebut untuk menghadapi potensi masalah tertentu (Hendrawan, 2019).

c. *Technology* (Teknologi)

Teknologi memiliki peran penting dalam berjalannya manajemen pengetahuan dan membutuhkan individu yang memiliki kemampuan dalam mengelola dan menggunakan teknologi tersebut. Hal ini dikarenakan teknologi dan alat yang digunakan dapat memfasilitasi seluruh proses – proses yang terdapat pada siklus manajemen pengetahuan. Teknologi yang dapat menjadi alat dalam manajemen pengetahuan sering disebut dengan *Knowledge Management Systems* (KMS) (Hendrawan, 2019).

Pengetahuan yang terdapat dalam manajemen pengetahuan terdiri dari 2 jenis, yaitu pengetahuan tacit dan eksplisit:

a. *Tacit Knowledge*

Pengetahuan tacit merupakan pengetahuan yang hanya diketahui oleh beberapa individu yang sulit untuk didokumentasikan dan disebarkan kepada individu lainnya. Pada umumnya, pengetahuan tacit didapatkan dari pengalaman personal dan tidak disajikan dalam bentuk tulisan (Hendrawan, 2019).

b. *Explicit Knowledge*

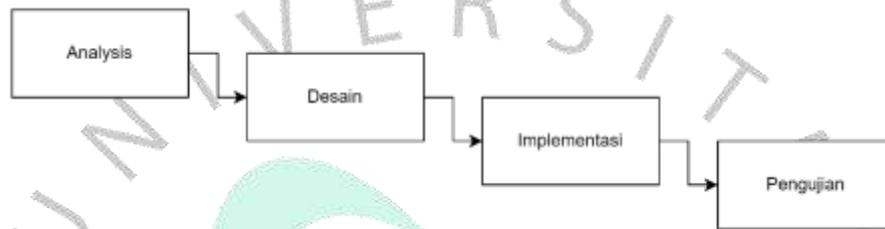
Pengetahuan eksplisit merupakan pengetahuan yang dapat didokumentasikan dan menggunakan bahasa formal agar dapat mudah dipahami dan disebarkan di dalam organisasi (Hendrawan, 2019).

### 2.1.2 Software Development Life Cycle (SDLC)

SDLC merupakan tahapan – tahapan yang terdapat pada proses pengembangan, perubahan, atau pembaruan sistem perangkat lunak (*software*). Penggunaan metodologi pengembangan sistem SDLC dapat menggunakan model – model pendekatan seperti *Waterfall*, *Prototyping*, RAD, iteratif, dan

spiral (Rosa & Shalahuddin, 2018). Dalam penelitian ini, penulis menerapkan model *Waterfall* untuk merancang dan mengembangkan aplikasi KMS berbasis web.

Model Waterfall adalah pemodelan sekuensial linier yang berbentuk seperti air terjun. Alur yang digunakan dalam menggunakan pendekatan ini terurut dari tahapan analisis, desain, implementasi, dan pengujian (Rosa & Shalahuddin, 2018).



**Gambar 2.1 Model Waterfall**

Sumber: Rosa & Shalahuddin, 2018

Berikut adalah penjelasan tahapan – tahapan yang terdapat pada model *Waterfall*:

a. *Requirement Analysis*

Pada tahapan ini, data kebutuhan sistem dikumpulkan untuk mengetahui kebutuhan, standar, dan spesifikasi yang sesuai dengan keinginan pengguna. Hasil dari pengumpulan data akan didokumentasikan dalam bentuk dokumen spesifikasi kebutuhan sistem (Rosa & Shalahuddin, 2018).

b. *System Design*

Hasil dokumentasi kebutuhan sistem kebutuhan akan digunakan sebagai acuan dalam mendesain sistem. Pada tahapan ini, desain antarmuka, struktur data, dan rencana implementasi sistem disusun untuk digunakan pada tahapan selanjutnya (Rosa & Shalahuddin, 2018).

c. *System Implementation*

Pada tahapan implementasi, desain sistem diterjemahkan ke dalam pemrograman aplikasi perangkat lunak. Hasil dari implementasi

menyesuaikan dengan desain sistem yang dibuat pada tahapan sebelumnya (Rosa & Shalahuddin, 2018).

d. *System Testing*

Setelah tahapan implementasi, aplikasi perangkat lunak akan diuji bagian – bagian dari sistem untuk memastikan bahwa tidak terjadi kesalahan (*error*) dan sesuai dengan kebutuhan pengguna (Rosa & Shalahuddin, 2018).

### 2.1.3 Oriented Analysis and Design (OOAD)

*Object Oriented Analysis* (OOA) adalah suatu metode atau pendekatan yang dapat digunakan pada proses analisis kebutuhan sistem yang akan dikembangkan dengan pendekatan berbasis objek. Sedangkan *Object Oriented Design* (OOD) adalah tahapan untuk menerjemahkan hasil analisis ke dalam desain sistem yang menggunakan konsep berorientasi objek. Kedua tahapan tersebut memiliki keterkaitan karena terjadi secara berulang - ulang yang batasannya hampir kasat mata pada saat berjalannya pengembangan aplikasi. Oleh karena itu, Seringkali kedua tahapan tersebut disingkat menjadi OOAD (*Object Oriented Analysis and Design*). Pemodelan yang dapat digunakan dalam OOAD adalah *Unified Modelling Language* (UML) yang dapat digunakan sebagai dokumentasi pengembangan aplikasi (Rosa & Shalahuddin, 2018).

UML merupakan bahasa perancangan sistem terstandarisasi yang diperuntukkan untuk membuat model perangkat lunak sehingga dapat dimengerti oleh banyak pengembang. UML dapat digunakan sebagai bahasa pemodelan visual dalam pengembangan perangkat lunak dengan konsep berorientasi objek. (Rosa & Shalahuddin, 2018). Pada penelitian ini, penulis menggunakan 4 diagram dari UML, yaitu *Use Case* dan *Activity Diagram* (*Behaviour*), *Sequence Diagram* (*Interaction*), serta *Class Diagram* (*Structure*):

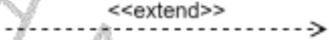
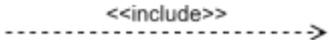
a. *Use Case Diagram*

Diagram *use case* merupakan pemodelan yang diperuntukkan untuk memvisualisasikan *system behaviour* pada perangkat lunak yang ingin

dikembangkan. Diagram ini digunakan untuk menggambarkan interaksi antara sistem dengan aktor yang terlibat/berkaitan. Penggambaran interaksi tersebut dapat digunakan untuk mendefinisikan fungsi atau fitur yang terdapat pada sistem tersebut (Rosa & Shalahuddin, 2018). Menurut Dennis (2015), berikut ini adalah notasi yang digunakan pada penggunaan *use case diagram*:

**Tabel 2.1 Simbol Use Case Diagram**

Sumber: Dennis, 2015

Simbol	Deskripsi Simbol
	<p>Use Case: Merupakan gambaran fungsi yang terdapat pada suatu sistem.</p>
	<p>Aktor: Merupakan individu, perangkat, atau sistem yang terlibat dalam interaksi yang terjadi di dalam sistem.</p>
	<p>Asosiasi: Merupakan hubungan antara <i>use case</i> dan aktor yang berinteraksi satu dengan lainnya.</p>
	<p><i>Extend</i>: Merupakan relasi antara <i>use case</i> jika dapat berdiri sendiri tanpa <i>use case</i> tambahan tersebut (Opsional).</p>
	<p><i>Include</i>: Merupakan relasi antara <i>use case</i> yang memiliki arti <i>use case</i> tersebut melibatkan fungsi <i>use case</i> yang ditunjukkan.</p>
	<p>Generalisasi: Merupakan hubungan antara <i>use case</i> yang</p>

	memiliki arti suatu <i>use case</i> lebih umum dari yang lainnya.
--	---

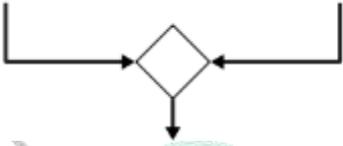
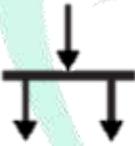
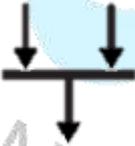
### b. Activity Diagram

Diagram aktivitas merupakan pemodelan visual yang menggambarkan *workflow* dari suatu sistem. *Activity diagram* digunakan untuk mendeskripsikan proses bisnis dengan penekanan aktivitas yang dilakukan oleh sistem tersebut (Rosa & Shalahuddin, 2018). Menurut Dennis (2015), berikut ini adalah simbol yang digunakan pada penggunaan *activity diagram*:

**Tabel 2.2 Simbol Activity Diagram**

Sumber: Dennis, 2015

Simbol	Deskripsi Simbol
	<i>Start</i> : Simbol yang menandakan awal dari aktivitas yang terjadi pada fungsi sistem tersebut.
	Aktivitas: Simbol yang mewakili Aktivitas yang dilakukan oleh sistem.
	<i>Decision Node</i> : Simbol yang menandakan percabangan <i>path</i> untuk memilih opsi aktivitas yang telah ditentukan.
	<i>End</i> : Merupakan status berakhirnya aktivitas pada sistem tersebut.

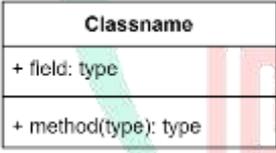
	<p><i>Swimlane</i>: Merupakan pemisah untuk bagian, individu, atau objek yang terlibat dalam aktivitas sistem tersebut.</p>
	<p><i>Merge Node</i>: notasi yang digunakan untuk menyatukan kembali percabangan pada <i>decision node</i>.</p>
	<p><i>Fork Node</i>: notasi yang digunakan untuk memisahkan beberapa aktivitas paralel.</p>
	<p><i>Join Node</i>: Notasi yang digunakan untuk menggabungkan kembali aktivitas yang bersifat paralel.</p>
 <p>Sumber: Seidl, 2014</p>	<p><i>Call Behaviour</i>: Merupakan notasi yang dapat digunakan untuk memanggil aktivitas lainnya untuk mengeksekusi proses-proses di dalamnya (Seidl, 2014).</p>

c. *Class Diagram*

Diagram ini merupakan gambaran struktural sistem yang mendefinisikan kelas-kelas yang ada di dalamnya. Setiap kelas terdiri dari atribut, yang merupakan variabel di dalam kelas, dan operasi, yang merupakan fungsi-fungsi yang dimiliki oleh kelas tersebut. Dalam pengembangan aplikasi KMS ini, penulis menggunakan struktur kelas yang diambil dari basis data atau model sistem MVC (Model, View, dan Controller). Informasi yang tersimpan dalam sistem basis data diwakili sebagai kelas (Rosa & Shalahuddin, 2018). Menurut Dennis (2015), berikut ini adalah simbol yang digunakan dalam menyusun *class diagram*:

Tabel 2.3 Simbol Class Diagram

Sumber: Dennis, 2015

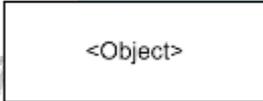
Simbol	Deskripsi Simbol
	Kelas: Merupakan representasi kelas sistem yang menerima dan menyimpan informasi.
	Asosiasi: Simbol yang menandakan hubungan antara 2 kelas atau lebih.
	Generalisasi: Merupakan penanda hubungan antara kelas yang bersifat unik.
	Agregasi: Simbol yang menandakan hubungan antara kelas yang bersifat <i>logical</i> (dapat berdiri sendiri).
	Komposisi: Simbol yang menandakan hubungan antara kelas yang bersifat <i>physical</i> (tidak dapat berdiri sendiri).

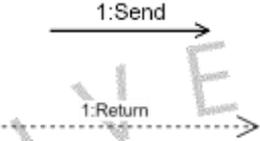
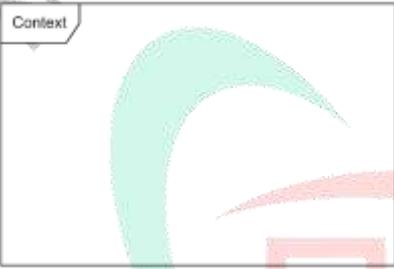
#### d. Sequence Diagram

Diagram sekuen adalah pemodelan visual untuk menggambarkan perilaku, kegiatan, keterkaitan, atau kelakuan suatu objek pada *use case* sistem tersebut. Jumlah dari diagram sekuen mengikuti banyaknya *use case* yang telah didefinisikan (Rosa & Shalahuddin, 2018). Menurut Dennis (2015), berikut ini adalah notasi yang terdapat dalam penyusunan *Sequence diagram*:

Tabel 2.4 Simbol Sequence Diagram

Sumber: Dennis, 2015

Simbol	Deskripsi Simbol
 Aktor	<i>Actor</i> : Merupakan individu, perangkat, atau sistem yang terlibat dalam interaksi berupa mengirim dan menerima pesan yang terjadi pada aktivitas sistem tersebut.
 <Object>	<i>Object</i> : Merupakan simbol yang menandakan suatu objek yang terlibat dalam mengirim dan menerima pesan.
 Object	<i>Lifeline</i> : Merupakan garis yang digunakan untuk merepresentasikan kehidupan suatu objek dalam berjalannya suatu sekuen.

	<p><i>Execution Occurrence:</i></p> <p>Merupakan simbol yang menyatakan bahwa suatu objek sedang mengirim atau menerima pesan dari objek lainnya.</p>
	<p><i>Message:</i> Merupakan simbol untuk mewakili proses perpindahan informasi dari suatu objek ke objek lainnya.</p>
	<p><i>Frame:</i> Merupakan visualisasi dari konteks pada <i>sequence diagram</i> yang dibuat.</p>

#### 2.1.4 Object Oriented Programming (OOP)

OOP merupakan strategi dan metode yang dapat digunakan dalam pengembangan sistem perangkat lunak yang menjadikan entitas atau fungsi bagian dari suatu objek. Metode OOP meliputi analisis, implementasi, dan pengujian berbasis objek (Rosa & Shalahuddin, 2018). Berikut ini adalah konsep dasar yang terdapat pada metode OOP:

##### a. *Class*

Merupakan kumpulan objek yang memiliki karakteristik yang sama. Suatu kelas memiliki atribut, operasi, arti dan relasi yang dapat diwariskan kepada kelas lainnya (Rosa & Shalahuddin, 2018).

##### b. *Object*

Merupakan sesuatu entitas pada suatu sistem yang dapat memiliki operasi dan menyimpan informasi yang dapat diterapkan untuk fungsi – fungsi yang terdapat pada sistem tersebut (Rosa & Shalahuddin, 2018).

c. *Method*

Merupakan fungsi atau operasi pada suatu kelas dan dapat diimplementasikan lebih dari satu *method* dalam satu kelas. *Method* digunakan untuk memanipulasi objek tersebut atau menjalankan fungsi/ fitur tertentu (Rosa & Shalahuddin, 2018).

d. *Attribute*

Merupakan variabel yang terdapat pada suatu kelas. Atribut dapat dinyatakan dalam bentuk nilai yang dimiliki oleh objek dalam suatu kelas objek (Rosa & Shalahuddin, 2018).

e. *Encapsulation*

Merupakan konsep untuk membungkus atribut dan method yang dimiliki oleh suatu objek agar cara kerja sistem tidak dapat diketahui oleh objek yang tidak diberikan akses (disembunyikan) (Rosa & Shalahuddin, 2018).

f. *Inheritance*

Merupakan konsep yang memungkinkan pewarisan dari sebagian atau keseluruhan suatu objek kepada objek lainnya (Rosa & Shalahuddin, 2018).

g. *Reusability*

Merupakan konsep pemanfaatan kembali suatu objek yang sudah dideklarasikan/didefinisikan untuk menyelesaikan masalah lainnya (Rosa & Shalahuddin, 2018).

h. *Polymorphism*

Merupakan konsep penggunaan objek dalam menjalankan fungsi yang berbeda – beda agar dapat menghemat jumlah baris *source code* aplikasi (Rosa & Shalahuddin, 2018).

### 2.1.5 Metode Penelitian Kualitatif

Metode penelitian kualitatif merupakan pendekatan atau metode yang menggunakan data deskriptif, baik itu dalam bentuk tertulis maupun lisan, untuk mengamati fenomena yang dialami oleh individu, keyakinan, atau dinamika sosial masyarakat. Pendekatan kualitatif dimulai dari asumsi-asumsi yang dapat diangkat atau dikaitkan dalam suatu penelitian. Data-data tersebut dikumpulkan dengan berbagai macam teknik yang akan menjawab pertanyaan-pertanyaan sebagai perantara untuk memahami lebih dalam fenomena, pandangan, persepsi, serta individu yang terpengaruh dan berpengaruh dalam lingkungan subjek penelitian. Semakin dalam hasil pengumpulan data, maka akan meningkatkan kualitas penelitian tersebut (Ardyan, 2023).

Proses pengumpulan data dilakukan untuk mengumpulkan data penelitian yang diperlukan untuk analisis data pada tahap berikutnya (Ardyan, 2023). Terdapat beberapa metode pengumpulan data dalam penelitian kualitatif, yaitu:

#### a. Wawancara

Percakapan antara dua orang atau lebih, baik secara formal maupun informal, yang bertujuan untuk memperoleh informasi yang relevan dengan konteksnya, dapat disebut sebagai wawancara. Metode pengumpulan data ini dapat menambah wawasan peneliti dalam memahami kondisi lapangan yang dialami oleh lawan bicara atau narasumber. Maka dari itu, pemilihan responden/narasumber penting agar mendapatkan data yang cocok dengan tujuan penelitian. Hal ini dapat dilakukan melalui evaluasi pekerjaan, jabatan, dan kegiatan-kegiatan yang diikuti oleh narasumber (Subakti, 2023).

Pertanyaan yang diajukan selama wawancara menggunakan pendekatan 5W1H, yaitu *What* (Apa yang terjadi), *When* (Kapan peristiwa tersebut terjadi), *Why* (Mengapa hal itu terjadi), *Who* (Siapa yang terlibat), *Where* (Dimana kejadian tersebut terjadi), dan *How* (Bagaimana peristiwa itu terjadi) (Subakti, 2023). Terdapat beberapa jenis wawancara, yakni sebagai berikut:

- i. Wawancara Terstruktur: Merupakan wawancara yang dilakukan Ketika peneliti sudah tahu tipe informasi yang

akan didapatkan dan memiliki pertanyaan yang sama antara responden/narasumber (Subakti, 2023).

- ii. Wawancara Semi Terstruktur: Merupakan wawancara yang lebih fleksibel daripada yang terstruktur dikarenakan pewawancara dapat meminta ide atau pendapat dari responden secara bebas (Subakti, 2023).
- iii. Wawancara Tidak Terstruktur: Merupakan wawancara bebas yang dilakukan tanpa mengikuti prosedur terstruktur dalam pelaksanaannya. Hasil wawancara ini hanya mencakup hal yang tidak diketahui secara pasti oleh peneliti. Oleh karena itu, responden wawancara ini perlu menceritakan lebih panjang dan detail agar mendapatkan hasil yang diinginkan (Subakti, 2023).

b. Observasi

Merupakan kegiatan pengamatan objek secara langsung di lapangan agar mendapatkan fakta-fakta detail peristiwa yang sesuai dengan kebutuhan data penelitian (Subakti, 2023). Teknik observasi dapat dibagi menjadi 2, yaitu:

- i. Observasi yang direncanakan dan Terkontrol: Merupakan observasi yang menggunakan daftar aspek-aspek yang perlu diamati pada saat proses pengamatan dilakukan (Subakti, 2023).
- ii. Observasi Tidak Terencana: Merupakan observasi yang dilakukan tanpa mengetahui apa yang harus diamati dari objek penelitian beserta lingkungannya (Subakti, 2023).

### 2.1.6 Data, Informasi, dan Pengetahuan

Data merupakan istilah untuk sebuah fakta yang dapat disimpan di media penyimpanan. Selain fakta, data dapat dikatakan sebagai sebuah fakta berupa bahan mentah atau deskripsi (objek atau aktivitas) yang belum diolah dan tidak dapat dipahami oleh pengguna. Agar dapat dipahami oleh pengguna, data diolah melalui proses tertentu agar dapat disajikan ke dalam bentuk yang

bermakna/berarti yang sering disebut dengan istilah informasi. Contoh dari data yang dapat dilihat dan digunakan dalam kehidupan sehari-hari adalah data penjualan, stok, harga, nama barang, dan lain – lain (Kadir, 2020).

Kd_Barang	Nama Barang	Stok	Harga
Brg001	Lotion	2	Rp5.000
Brg002	Parfume	5	Rp10.000

**Gambar 2.2 Contoh Data Penjualan**

Sumber: Kadir, 2020

Informasi adalah data yang telah melalui serangkaian proses yang dapat menghasilkan arti. Proses tersebut dilakukan melalui interpretasi, peringkasan, dan penyajian yang mudah dipahami oleh manusia seperti grafik. Hasil pemrosesan data berupa informasi dapat membantu pengguna membuat keputusan saat ini dan di masa depan. (Kadir, 2020).

Pengetahuan merupakan pemahaman terhadap suatu informasi yang berkaitan dengan suatu objek, lingkungan atau keadaan sekitar. Pengetahuan bisa didapatkan dari suatu pembelajaran, pelatihan atau pengalaman masing-masing individu. Dengan adanya pengetahuan yang relevan, keputusan yang sesuai dapat diambil sesuai dengan kondisi atau masalah yang dihadapi (Hadiprakoso, 2021).

### **2.1.7 Basis Data dan Database Management Systems (DBMS)**

Basis data adalah koleksi data terstruktur yang saling terhubung yang disimpan dalam media penyimpanan tertentu, memungkinkan pengguna untuk mengakses kembali data tersebut (Rachmadi, 2020). Basis data disusun dan diorganisasikan ke dalam bentuk suatu skema tertentu yang dapat dimanipulasi menggunakan suatu perangkat lunak. Data yang direkam di dalam basis data tersebut adalah kumpulan fakta yang dapat berbentuk huruf, angka, simbol, suara, gambar, dokumen, dan kombinasi dari objek - objek tersebut (Ihksan, 2023). Untuk melakukan pengelolaan terhadap data yang disimpan, terdapat

beberapa perintah, *syntax*, atau operasi yang dapat dilakukan terhadap data yang disimpan di dalam *database*, antara lain:

- a. *Create*: Merupakan operasi basis data yang digunakan untuk membuat tabel baru yang berisikan kolom, nilai kunci, tipe data, batasan, dan lain-lain (Ihksan, 2023).
- b. *Update*: Merupakan operasi basis data yang diperuntukkan untuk mengubah atau memperbaharui baris data yang sebelumnya telah dimasukkan (Ihksan, 2023).
- c. *Delete*: Adalah operasi basis data untuk menghapus data yang terdapat pada suatu tabel (Ihksan, 2023).
- d. *Select*: Adalah operasi basis data untuk memilih atau mengambil data dari tabel yang sesuai dengan pernyataan atau parameter yang dimasukkan oleh pengguna (Ihksan, 2023).
- e. *Join*: Merupakan operasi basis data yang digunakan untuk menggabungkan kolom data dari dua tabel atau lebih sesuai dengan relasi yang dimiliki menggunakan nilai kunci (Ihksan, 2023).
- f. *Insert*: Merupakan operasi basis data yang memiliki fungsi untuk menambahkan data baru ke dalam tabel (Ihksan, 2023).
- g. *Order By*: adalah operasi dalam basis data yang mengurutkan data berdasarkan nilai dalam sebuah kolom. Pengguna dapat mengatur pengurutan ini secara naik (*ascending*) atau turun (*descending*), sesuai dengan kebutuhan mereka (Ihksan, 2023).

Basis data memiliki 3 konsep penting dalam penerapannya, yaitu Entitas, *data type* dan variabel. Dalam suatu basis data, entitas merupakan suatu objek nyata yang direpresentasikan dengan sebuah tabel yang memiliki atribut. Dalam sebuah tabel, terdapat *data type* yang menggambarkan bentuk atau jenis nilai yang disimpan. Data tersebut memiliki batasan atau aturan yang memiliki keterkaitan dengan nilai yang disimpan (Contoh: VARCHAR, DATE, INTEGER, dan lain-lain). Untuk dapat menggunakan nilai-nilai yang terdapat pada basis data ke dalam kode pemrograman (*query*), variabel digunakan untuk menyimpan perintah atau pernyataan untuk menghasilkan *output* yang sesuai

dengan parameter pengguna (Romadloni, 2024). Terdapat beberapa jenis basis data yang mengimplementasikan konsep-konsep tersebut, antara lain:

a. Basis Data Hierarkis

Jenis basis data ini memiliki *record* dengan bentuk pohon yang terdiri dari induk dan anak. Basis data ini cocok digunakan untuk menyimpan struktur organisasi (Romadloni, 2024).

b. Basis Data Relational

Merupakan jenis basis data yang disimpan dengan menggunakan tabel yang memiliki hubungan satu dengan lainnya. Hubungan antara tabel direpresentasikan dengan menggunakan *primary* dan *foreign key* (Romadloni, 2024).

c. Basis Data Jaringan

Jenis basis data ini memiliki struktur yang mirip dengan jenis hierarkis yaitu memiliki induk dan anak. Perbedaannya pada basis data ini ialah kemungkinan adanya beberapa induk yang saling menghubungkan antara *record* atau penyimpanan data (Romadloni, 2024).

d. Basis Data Objek

Merupakan jenis basis data yang menggabungkan konsep *object-oriented programming*. Data yang disimpan disusun ke dalam sebuah objek yang memiliki atribut dan metode (Romadloni, 2024).

e. Basis Data Dokumen

Jenis basis data ini menyimpan data ke dalam bentuk dokumen (Contoh: JSON) untuk memungkinkan struktur data yang kompleks dan fleksibel (Romadloni, 2024).

DBMS merupakan sistem perangkat lunak yang diperuntukkan untuk mengelola, menyimpan, dan menyajikan data kepada pengguna (Rosa & Shalahuddin, 2018). DBMS dapat memberikan keuntungan bagi penggunaannya, seperti mempercepat dan mempermudah akses data, menghemat ruang penyimpanan, menyimpan data dengan jumlah yang besar, mencegah hilangnya data penting, dan memperkuat keamanan (Ningsih, 2022). Berikut ini adalah fungsi - fungsi yang terdapat pada DBMS:

- a. *Define*: Memungkinkan pengguna untuk mendefinisikan tipe dan struktur data yang disimpan.
- b. *Construct*: Mengkonfigurasi data yang tersimpan di dalam beberapa media penyimpanan yang dikontrol oleh DBMS tersebut.
- c. *Manipulate*: Memungkinkan pengguna untuk mengambil, mengubah, dan membuat laporan dari data yang disimpan di dalam *database*.

Terdapat beberapa komponen yang memiliki keterkaitan dengan DBMS, yaitu:

- a. *Hardware*: Merupakan perangkat atau alat yang digunakan untuk media instalasi perangkat lunak seperti komputer, *workstation*, *server*, laptop, dan lain-lain yang terhubung dengan DBMS (Hadiprakoso, 2021).
- b. *Software*: Merupakan perangkat atau alat yang digunakan untuk menjalankan sistem basis data seperti *operating system*, DBMS, dan program lainnya yang memiliki hubungan fungsionalitas (Hadiprakoso, 2021).
- c. *Data*: Merupakan komponen penting dari sebuah basis data yang berisikan fakta-fakta yang diperlukan sebagai bahan mentah dalam penyajian informasi kepada pengguna (Hadiprakoso, 2021).
- d. *Prosedur*: Merupakan suatu aturan yang mengatur proses penggunaan sistem basis data. Prosedur perlu diperhatikan oleh setiap organisasi yang menerapkan konsep basis data agar dapat mengikuti standar yang telah ditetapkan (Hadiprakoso, 2021).
- e. *Manusia*: Merupakan komponen yang mencakup sistem basis data secara keseluruhan karena memiliki fungsi sebagai pengguna dan pengontrol sistem (Hadiprakoso, 2021).

Untuk menjalankan fungsi – fungsi yang terdapat pada DBMS, pengguna dapat menggunakan Bahasa SQL (*Structured Query Language*). Bahasa tersebut adalah bahasa perintah basis data yang telah distandarisasi oleh ANSI (*American National Standards Institute*), digunakan untuk mengambil, memperbarui, menambahkan, dan menghapus data yang disimpan dalam *database*. Bahasa ini juga merupakan bahasa yang digunakan dalam salah satu jenis DBMS yaitu RDBMS (*Relational Database Management*) (Fitri, 2020).

Basis data relational sendiri adalah suatu konsep yang memungkinkan suatu tabel kumpulan data memiliki hubungan satu sama lain dengan menggunakan *primary* dan *foreign key* (Ihksan, 2023). Berikut ini adalah contoh dari tabel yang memiliki relasi dalam basis data relasional:

a. Tabel Konsumen

Customer_num	Customer_name	Address
1	A	Delhi
2	B	Mumbai

Gambar 2.3 Contoh Tabel Pelanggan

Sumber: Vaidya, 2021

b. Tabel Penjualan

Order_num	Customer_num	Amount
001	1	12340
002	2	9740

Gambar 2.4 Contoh Tabel Penjualan

Sumber: Vaidya, 2021

c. Relasi Tabel Berdasarkan Nilai Kunci

Order_num	Customer_num	Amount	Customer_name	Address
001	1	12340	A	Delhi
002	2	9740	B	Mumbai

Gambar 2.5 Contoh Relasi Tabel Pelanggan dengan Penjualan

Sumber: Vaidya, 2021

Terlihat pada **Gambar 2.3, 2.4, dan 2.5**, dengan adanya pengulangan data pada kolom “Customer\_num” data dari kedua tabel dapat digabungkan berdasarkan nilai *primary* dan *foreign key*. Terdapat sebuah persyaratan untuk membuat atau mempertahankan relasi yaitu nilai dari *primary key* tidak boleh

kosong dan harus unik agar dapat direferensikan pada tabel lainnya (Vaidya, 2021).

Basis data relational memiliki konsep-konsep fundamental yang diterapkan untuk membantu dalam merancang *database*, antara lain:

- a. Entitas: Merupakan suatu pemodelan objek yang memiliki atribut. Contoh dari entitas adalah Mahasiswa yang memiliki atribut nim, program studi, umur, dan nama (Vaidya, 2021).
- b. Kardinalitas: Merupakan definisi dari relasi antara entitas yang memiliki tiga jenis, yaitu *one-to-one*, *one-to-many*, dan *many-to-many* (Vaidya, 2021).
- c. Kunci: Ada beberapa jenis kunci dalam basis data, termasuk *candidate key* (kunci yang unik yang digunakan untuk mengidentifikasi baris dalam sebuah tabel, di mana hanya satu yang dapat dijadikan sebagai kunci utama), *primary key* (kunci yang digunakan untuk menyatakan bahwa baris dalam sebuah tabel adalah unik), dan *foreign key* (kunci yang mereferensikan nilai dari *primary key* dari tabel lain untuk membangun relasi) (Vaidya, 2021).

### 2.1.8 MySQL

MySQL adalah salah satu aplikasi yang dapat dimanfaatkan oleh pengguna untuk mengelola basis data dan bersifat *open-source* yang memiliki 2 tipe lisensi yaitu *Freeware* dan *Shareware*. Dalam suatu database pada MySQL, dapat terdiri dari beberapa tabel (Memiliki baris dan kolom) yang memiliki hubungan satu dengan lainnya. Maka dari itu, MySQL merupakan perangkat lunak DBMS dengan jenis RDBMS (Fitri, 2020).

MySQL menggunakan bahasa query yang terdiri dari dua bagian utama yang digunakan dalam penelitian ini, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML) (Yanto, 2016).

#### a. *Data Definition Language* (DDL)

DDL adalah serangkaian perintah atau bahasa SQL yang digunakan untuk membuat, menghapus, dan mengubah struktur objek-objek dalam basis data

(database, tabel, function, index, dan view) di dalam *database*. DDL memiliki 3 skema utama yaitu: *CREATE*, *DROP*, dan *ALTER*.

b. *Data Manipulation Language* (DML)

DML merupakan serangkaian perintah atau bahasa SQL yang digunakan untuk mengakses atau mengubah data terdapat di dalam *database*. Beberapa skema dalam penggunaan DML mencakup *INSERT*, *SELECT*, *UPDATE*, dan *DELETE*.

### 2.1.9 Pemrograman Web

Pemrograman web terdiri dari 2 sisi, yaitu *client* dan *server sided programming*. Pemrograman pada sisi *client* adalah program yang dijalankan dari sisi pengguna yang mengandung *script* dan dijalankan di dalam *web browser* yang digunakan untuk mengakses web tersebut. Contoh dari pemrograman dari sisi *client* adalah HTML dan Javascript. Sedangkan, pemrograman dari sisi *server* adalah program yang dijalankan di dalam *web server*. *Output* pengolahan data dari program tersebut akan diberikan kepada pengguna melewati *web browser* tanpa memperlihatkan *source code* aplikasi tersebut (Agusriandi, 2018). Pada tahapan implementasi aplikasi pada penelitian ini, bahasa pemrograman *web* yang diimplementasikan pada pengembangan aplikasi KMS adalah sebagai berikut:

a. *Hyper Text Markup Language* (HTML)

Merupakan bahasa deklaratif berupa text (*plain text file*) untuk menyatakan arti khusus pada dokumen web. Versi HTML yang paling terbaru adalah HTML 5. Karakteristik dari HTML adalah antara lain memiliki ekstensi *file* “.html” serta menggunakan *tag* yang berpasangan pada *tag* awal” <tag>” dan akhir “</tag>” (Agusriandi, 2018).

b. *Cascading Style Sheet* (CSS)

CSS adalah styling language yang digunakan untuk memberikan atribut dekorasi pada tampilan web dengan mudah (Agusriandi, 2018). CSS dapat ditulis dengan beberapa cara pada dokumen HTML, antara lain:

1. *Inline*: CSS yang ditulis di dalam *tag* HTML.

2. *Embedded*: CSS yang ditulis pada bagian *head* di dalam dokumen HTML.

3. *Linked*: CSS yang ditulis pada *file* atau halaman yang berbeda dan terpisah dari dokumen HTML.

c. Javascript

Javascript merupakan bahasa pemrograman skrip yang dapat digunakan dan ditempatkan di dalam dokumen HTML untuk *client sided programming*. Karakteristik Javascript yang digunakan di dalam dokumen HTML adalah dapat ditempatkan di bagian *head* atau *body*. Bahasa ini dapat dimanfaatkan untuk memvalidasi *input form* yang terdapat pada dokumen HTML sebelum dikirim ke *web server* (Agusriandi, 2018).

d. PHP

PHP adalah bahasa pemrograman *web* yang dapat digunakan untuk mengembangkan aplikasi pada bagian *server sided programming*. Untuk menggunakan PHP pada dokumen HTML, *file* disimpan dalam bentuk ekstensi PHP yaitu (.php). PHP dapat dimanfaatkan oleh penulis agar dapat membuat web statis memiliki fungsi – fungsi yang diperlukan oleh sistem tersebut seperti form, pengelolaan data, *login*, dan fungsi sistem lainnya (Agusriandi, 2018).

### 2.2.0 Framework

*Framework* adalah kumpulan dari fungsi, kelas, dan fitur yang siap digunakan oleh pengembang (Adri, 2018). *Framework* menyediakan beberapa fitur-fitur yang dapat digunakan oleh pengembang untuk mengembangkan suatu aplikasi (*Multiple Database*, *ORM*, *Pagination*, dan lain-lain), memudahkan perawatan sistem, dan mempercepat tahapan implementasi sistem (Supono, 2018). Hal ini dapat dimudahkan dengan adanya penerapan konsep Model, View, dan Control (MVC) untuk menjaga struktur kode (Arifin, 2024). Konsep ini memisahkan struktur kode aplikasi menjadi 3, yaitu:

a. *Model*

*Model* adalah komponen yang bertugas untuk melakukan interaksi dan komunikasi dengan *database*.

b. **View:**

*View* adalah komponen yang digunakan untuk menampilkan fungsi sistem yang dapat dilihat oleh pengguna.

c. **Control:**

*Control* adalah komponen yang berfungsi untuk menghubungkan antara komponen *view* dan *model*. Respon sistem terhadap *input* dari pengguna ditangani oleh komponen ini yang nantinya akan menghasilkan *output* sesuai dengan kode program.

## 2.2 Tinjauan Studi

Tinjauan Studi yang digunakan untuk mendukung penelitian dalam perancangan dan pengembangan aplikasi KMS oleh penulis adalah sebagai berikut:

1. Tinjauan Studi berupa jurnal yang berjudul “KMS (Knowledge Management System) Obat Ibu Hamil Berbasis Android” ini bertujuan untuk merancang dan membangun *Knowledge Management System* (KMS) berbasis Android untuk memudahkan akses informasi tentang obat-obatan yang aman untuk ibu hamil. Penulis dari penelitian ini adalah Halimah Tus Sadiyah pada Jurnal Teknologi Informasi dan Ilmu Komputer Vol. 8 No.2. Penulis jurnal mengangkat permasalahan pada pencarian dan pemilihan obat yang aman dan sesuai dengan ibu hamil yang terstandarisasi oleh US FDA (*United States Food and Drug Administration*) yang memerlukan akurasi dan ketepatan yang tinggi untuk menjamin keamanan. Maka dari itu, penulis merancang dan membangun sistem yang memuat pengetahuan tentang obat yang digunakan oleh ibu hamil. Penelitian ini menggunakan metode KMSLC yang melalui beberapa tahap, yaitu penangkapan pengetahuan, membuat *blueprint* KMS, dan validasi KMS. Pengembangan KMS ini didasarkan pada evaluasi dari infrastruktur lab inovasi sekolah vokasi Universitas Pakuan yang menghasilkan data dan informasi mengenai pengetahuan apa saja yang dibutuhkan melalui wawancara dan observasi obat – obat yang dijual. Hasil dari penelitian menunjukkan bahwa KMS yang dibangun dapat membantu ibu hamil dan karyawan dalam memilih

obat yang aman untuk dikonsumsi selama kehamilan dan menurunkan resiko konsumen terhadap penyakit yang berbahaya. Kesimpulan dan hasil dari penelitian ini dapat dijadikan sebagai pendukung dan bahan referensi untuk membuat rancangan dan membangun aplikasi KMS yang dapat membantu pengelolaan pengetahuan di dalam UKM.

2. Tinjauan studi berupa jurnal yang berjudul “Rancang Bangun Sistem Informasi Pilkades Kabupaten Pelalawan Berbasis Web (Studi Kasus: Dinas Kominfo Kab.Pelalawan)” ini ditulis oleh Elvi Wulandari pada Indonesian Journal of Informatic Research and Software Engineering (IRSE) volume 2 No. 1 pada tahun 2022. Penulis membahas tentang bagaimana Kantor Dinas KOMINFO Kabupaten Pelalawan membutuhkan sistem yang dapat memfasilitasi pengelolaan data pada saat berlangsungnya pemilihan kepala desa (PILKADA). Maka dari itu, penulis memiliki tujuan untuk mengembangkan aplikasi dengan menggunakan pendekatan *Object Oriented Design & Analysis* (OOAD). Temuan dan hasil dari penelitian yang telah dilakukan oleh penulis jurnal ini adalah aplikasi yang memudahkan instansi pemerintahan dalam mengelola data yang diperlukan dan meningkatkan kinerja karyawan Kantor Dinas tersebut. Hal ini dapat dijadikan referensi oleh dalam mengembangkan sistem perangkat lunak dengan menggunakan pendekatan OOAD.
3. Tinjauan studi berupa jurnal yang berjudul “Rancang Bangun Aplikasi Knowledge Management Berbasis Web” ini membahas tentang pengembangan aplikasi knowledge management berbasis web yang dibangun menggunakan bahasa pemrograman PHP dan *database* MySQL yang ditulis oleh Yance Sonatha (Jurnal Inovasi Vokasional dan Teknologi Vol.18 No.2). Jurnal ini membahas tentang permasalahan pengetahuan yang keluar bersama suatu individu yang resign dari perusahaan tersebut (*knowledge walkout*). Karena hal tersebut, diperlukan adanya pelatihan tambahan yang memakan waktu dan biaya yang berdampak besar bagi perusahaan. Maka dari itu, perlu adanya sistem yang dapat melakukan pengelolaan pada *knowledge* yang dimiliki oleh karyawan dan mudah

diakses ketika dibutuhkan di masa yang akan datang. Hasil dari penelitian ini adalah pengembangan aplikasi untuk mengelola pengetahuan dan informasi pada perusahaan PT Gamatechno agar dapat dimanfaatkan secara optimal. Dalam pengembangan aplikasi ini, penulis menggunakan model *Waterfall* untuk pengembangan aplikasi yang terstruktur dan jelas tahapan dalam perancangan dan pembangunan. Aplikasi ini memiliki beberapa fitur, seperti manajemen pengetahuan, pencarian pengetahuan, dan berbagi pengetahuan di dalam ruang lingkup perusahaan PT Gamatechno. Penulis juga melakukan evaluasi terhadap aplikasi yang dikembangkan untuk memastikan bahwa aplikasi tidak mengalami penyimpangan pada saat digunakan oleh pengguna dari segi fungsi dan teknis. Berdasarkan hasil penelitian ini, pengembangan aplikasi *knowledge management systems* dengan metode *Waterfall* terbukti dapat mewujudkan dan menghasilkan produk sistem informasi yang sesuai standar dan kriteria pengguna. Karena hal tersebut, penulis memperoleh pemahaman lebih mendalam mengenai penggunaan model *Waterfall* dalam pembuatan aplikasi KMS.

4. Tinjauan studi berupa jurnal yang memiliki judul “Knowledge Management System Berbasis Web Tentang Budidaya Hidroponik Untuk Mendukung Smart Society” ini ditulis oleh Brilian Sidhatama pada Jurnal Pengembangan Sistem Informasi dan Informatika Vol.1, No.3 menjelaskan pengembangan *knowledge management systems* dengan metode KMSLC. Berdasarkan temuan dari penelitian ini, KMS yang telah dibuat dapat dimanfaatkan untuk menyimpan pengetahuan *tacit* dan *explicit* dengan baik dan sesuai kebutuhan organisasi yang berkaitan. Oleh karena itu, penjelasan dari hasil penelitian ini dapat dijadikan referensi untuk menyelesaikan masalah terkait dokumentasi dan penyimpanan *tacit knowledge*.
5. Tinjauan studi berupa jurnal yang berjudul “Arsitektur Knowledge Management System Bagi Unit Kegiatan Mahasiswa (UKM) Dan Organisasi Mahasiswa (ORMA) Di STMIK Amikom Purwokerto” ini ditulis oleh Puji Ratwiyanti pada jurnal Pro Bisnis Vol. 11 No. 2. Penulis membahas tentang mengenai pentingnya pengetahuan agar organisasi bisa mensukseskan berjalannya suatu program kerja. Namun, UKM di STMIK

Amikom Puwekerto menghadapi permasalahan yaitu pengetahuan yang bersifat *tacit* tidak didokumentasikan dan hanya disebar melalui komunikasi pribadi. Hal ini menyebabkan terjadinya kesalahan yang sama terulang kembali karena kurangnya pengetahuan yang diberikan kepada anggota UKM. Demi memenuhi kebutuhan atas informasi dan pengetahuan yang penting untuk berjalannya UKM, diperlukan sistem yang bisa menangkap, menyimpan, mengelola, dan menyebarkan pengetahuan yang sesuai dengan kebutuhan strategis. Hasil yang diberikan oleh penelitian ini adalah berupa infrastruktur yang dapat mendukung pengelolaan *knowledge management* untuk UKM bersangkutan melalui integrasi basis data untuk meminimalisir kemungkinan hilangnya data dan informasi penting. Hal ini dapat dijadikan sebagai dukungan dan referensi penulis dalam pembangunan sistem *knowledge management* untuk unit kegiatan mahasiswa di suatu universitas karena pentingnya integrasi basis data di satu tempat agar tidak terjadi hilangnya pengetahuan yang akan dibutuhkan di masa depan.