

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Menurut Lilik Joko Susanto, 2018, yang berjudul “Penerapan *Knowledge Mangement System* dalam Manajemen Data Kegiatan Hidroponik (Studi Kasus:Komunitas Hidroponik Bandar Lampung)”. Tujuan dari penelitian ini yaitu *Website knowledge management* komunitas hidroponik Bandar Lampung ini akan memberikan informasi kepada *user* dan memberikan wadah untuk berdiskusi, berinteraksi maupun bertukar informasi mengenai teknik hidroponik. *Website knowledge management* komunitas hidroponik Bandar Lampung ini memudahkan kepada anggota, pengurus dan masyarakat dalam mengakses *website* ini dengan cepat dan akurat dapat diakses melalui perangkat *personal computer* dan juga perangkat *smartphone* yang tersambung ke jaringan *internet* tidak terbatas ruang dan waktu.

Menurut *Randa Bias Efrima*, 2021, yang berjudul “Analisis Dan Perancangan *Knowledge Management System* Berbasis Android Pada SMPN 8 Kota Jambi” Penelitian ini bertujuan untuk perancangan *knowledge management system* berbasis android yang dapat memfasilitasi penyimpanan sehingga memudahkan pertukaran *knowledge* pada SMPN 8 Kota Jambi, Sistem yang dirancang diharapkan dapat membantu proses mengumpulkan

penyimpanan, pendistribusian *knowledge* antar guru sehingga mendukung proses kegiatan sekolah.

Menurut *Yoga Pranata*, 2021, yang berjudul “Analisis Dan Perancangan Knowledge Management System Berbasis Web Pada PT Metrindo” Penelitian ini bertujuan untuk Prototipe rancangan KMS pada PT Metrindo bisa mengelola pengetahuan serta menciptakan pengetahuan baru melalui hubungan sinergis dan saling mempengaruhi antara pengetahuan tacit yang dimiliki oleh setiap aktor seperti kemampuan teknis seorang supervisor bangunan dalam menentukan ketebalan adukan semen yang akan digunakan dalam mengecor bangunan yang dapat ditransfer kepada aktor lain melalui fitur group discussion dan pengetahuan explicit seorang project manager dalam menentukan berapa lama sebuah pekerjaan akan dilakukan dan siapa saja yang terlibat dalam pekerjaan tersebut dalam fitur project coordination. Pemilihan pengembangan sistem berbasis web akan memberikan fleksibilitas dalam pengelolaan pengetahuan. Dalam hal ini pengelolaan pengetahuan bisa dilakukan tanpa batasan waktu

## **II.2. Uraian Teoritis**

### **II.2.1. Sistem**

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau menyelesaikan suatu sasaran tertentu.

Berdasarkan definisi tersebut, dapat disimpulkan bahwa pengertian sistem yaitu merupakan gabungan dari berbagai elemen yang bekerja sama untuk mencapai suatu target atau tujuan.

Menurut McLeod dalam jurnal (Mamed Rofendy Manalu, 2015) “Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan” (Yunahar Heriyanto : 2018).

### **II.2.2. Informasi**

Informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan tersebut bisa menjadi informasi, hasil pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut. Berdasarkan definisi tersebut, dapat disimpulkan bahwa informasi merupakan hasil dari pengolahan data atau data yang sudah di proses.

Menurut Davis mengungkapkan : “Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. (Yunahar Heriyanto : 2018).

### **II.2.3. Knowledge Management System**

*Knowledge Management Systems* merupakan sistem informasi berbasis ilmu pengetahuan (*knowledge*) yang dapat mendukung hasil cipta (kreasi), pengaturan dan penyebaran dari ilmu pengetahuan bisnis kepada para pekerja dan manajer perusahaan. Tidak berbeda jauh dengan definisi yang dikemukakan oleh

O'Brien, Turban et al. mengemukakan bahwa *knowledge management system* merupakan sistem informasi berbasis pengetahuan yang mendukung kreasi, pengaturan dan penyebaran dari ilmu pengetahuan bisnis kepada para pekerja dan manajer dalam perusahaan secara keseluruhan. Berdasarkan pengertian KMS diatas maka dapat disimpulkan bahwa KMS merupakan alat yang ditunjukkan untuk mendukung sebuah strategi perusahaan dalam mengumpulkan, mengidentifikasi, memilih, mengolah dan menyebarkan knowledge yang ada, baik individual knowledge maupun knowledge perusahaan sehingga dapat mendorong pembelajaran dalam organisasi serta terciptanya berbagi pengetahuan antar karyawan. (Gunawan : 2020).

#### **II.2.4. Database**

Basis data (*database*) merupakan kumpulan berkas (*file*) atau data yang saling berkaitan (mempunyai relasi yang ditunjukkan dengan kunci dari tiap berkas), dimana penyimpanannya disusun secara sistematis di dalam komputer untuk dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) sehingga dapat menghasilkan informasi. *Database* biasanya dipakai untuk mengelola berkas-berkas penting dalam lingkungan perusahaan, instansi atau organisasi. Berkas-berkas pada *database* pada suatu badan usaha, instansi atau organisasi disusun atau diurutkan secara rapih dan terstruktur, sehingga dapat dengan mudah diambil sewaktu-waktu untuk ditampilkan dalam bentuk laporan.

Untuk memasukkan dan mengambil data ke dan dari *database*, diperlukan perangkat lunak yang dikenal dengan istilah sistem manajemen basis data (*database management system*)

(DBMS). Perangkat lunak DBMS merupakan sistem yang memungkinkan pengguna dari basis data (*database user*) untuk memelihara, mengontrol dan mengakses data secara lebih praktis dan efisien. DBMS ini menjadi lapisan yang menjadi jembatan penghubung antara basis data dengan program aplikasi dan memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

Manfaat menggunakan *database* adalah *sharing data resources*, sehingga data pada satu *database* tadi bisa digunakan bersama-sama dan penggunaannya dapat terkontrol dengan baik dari pusat. *Database* terpusat dapat mempunyai efek positif dalam pengurangan biaya perangkat karena tiap divisi atau bidang dalam suatu perusahaan dapat mengakses *database* dari jaringan yang terhubung dengan *server* pusat. Keunggulan lain adalah *database* juga memberikan kemudahan dan kecepatan dalam mengakses dan memperoleh data.

Berdasarkan jenisnya, *database* dikelompokkan menjadi 2 bagian, yaitu *database flat – file* dan *database relasional*. *Database flat – file* umumnya digunakan pada data berukuran kecil dan dapat dirubah dengan mudah. *Database* jenis ini menggunakan table tunggal yang kolom-kolomnya merepresentasikan seluruh parameter. Sementara itu *database relasional* mempunyai struktur yang lebih logis terkait cara penyimpanan.

*Relational database*, merupakan penggabungan beberapa tabel dengan berbagai metode untuk dapat berkerjasama. Hubungan-hubungan antar tabel data dapat dibandingkan, disatukan, dan ditampilkan dalam formform *database*. Basis data relasional umumnya menggunakan sekumpulan tabel dua dimensi dimana

masing-masing tabel tersebut tersusun atas baris (*tupel*) dan kolom (*atribut*). Hubungan antara dua atau lebih tabel dibuat dengan menggunakan *key* (atribut kunci) yaitu *primary key* pada salah satu tabel dan *foreign key* di tabel yang lain. Basis data relasional saat ini biasa dijadikan pilihan yang utama karena beberapa keunggulannya. Beberapa contoh perangkat lunak basis data relasional diantaranya oracle, Microsoft SQL Server, MariaDB, dan Mysql. (Antonius Irianto Sukowati : 2017)

### **II.2.5. SQL Server**

Microsoft SQL Server 2008 merupakan perangkat lunak *Relational Database Management System* (RDBMS) yang handal dan dirancang untuk mendukung proses transaksi yang besar dengan menggunakan arsitektur client/server. *SQL Server* adalah *software* (perangkat lunak) RDBMS kelas *enterprise* yang banyak digunakan dalam dunia korporat. Dengan *SQL Server* dalam penggunaannya, pemakai dapat menyimpan banyak data dan menerapkannya untuk kepentingan bisnis dalam perusahaan. Untuk menjalankan database relasional, maka kita harus mengenal perintah-perintah SQL (*Structured Query Language*). Dalam MS SQL Server dapat mendukung SQL sebagai bahasa dalam memproses *query* ke dalam *database*. Microsoft SQL Server banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi database atau penyimpanan data.

Perintah SQL merupakan perintah untuk mengolah, menampilkan dan memanipulasi data yang tersimpan dalam sebuah RDBMS. *Data Definition Language* (DDL) adalah merupakan perintah untuk membuat atau mengubah

struktur objek *database* seperti *create*, *alter*, *drop*, *truncate* dan *rename*. *Data Control Language* (DCL) merupakan perintah untuk mengatur hak akses *user* dalam *database*, seperti *grant* dan *revoke*. Sedangkan untuk *Data Manipulation Language* (DML) merupakan perintah untuk melakukan manipulasi data seperti menambah, menghapus, dan menubah data dalam *table*, misalkan *insert*, *delete*, *update* dan *merge*. *Data Manipulation Language* merupakan suatu metode *query* yang dapat ddipergunakan bila *Data Definition Language* telah terjadi, sehingga fungsi dari *query* DML ini untuk melakukan manipulasi data yang telah dibuat dalam *database*. (Didik Setiyadi, 2020).

#### **II.2.6 Bahasa Pemrograman C#**

C# merupakan bahasa berorientasi objek yang elegan dan aman yang memungkinkan pengembang untuk membangun berbagai aplikasi yang aman dan kuat yang beroperasi pada *.NET Framework*. Bahasa pemrograman ini telah digunakan oleh para pengembang untuk membangun aplikasi berbasis *website* dinamis, *desktop*, bahkan *mobile*. Bahasa pemrograman ini juga mendukung teknologi *cloud service* seperti *Windows Communication Foundation (WCF)*, *ASP.NET Core*, dan lain sebagainya. Sedangkan *.Net Framework* itu sendiri adalah *platform* pengembangan yang digunakan untuk membangun aplikasi berbasis *Windows*, *Windows Phone*, *Windows Server*, dan *Microsoft Azure*. Sehingga, dalam pembuatan suatu aplikasi ataupun *website*, bahasa pemrograman C# sudah dirancang untuk cocok dengan *.Net Framework*. (Fityan Aula Juyuspan : 2017).

### **II.2.7 . ASP NET MVC**

ASP.NET MVC adalah arsitektur yang digunakan untuk mengembangkan *website* dinamis. Untuk mengembangkan *website* berbasis arsitektur ini membutuhkan versi Microsoft .NET Framework 3.5 atau lebih tinggi. Bahkan *website* yang dikembangkan menggunakan ASP.NET MVC memiliki kelebihan seperti SEO (*Search Engine Optimizer*) *friendly* . Artinya arsitektur ini memiliki kelebihan dasar yaitu salah satunya dapat di-*index* dengan mudah oleh *search engine* seperti Google, Yahoo, dan Bing. ASP.NET dikembangkan oleh Microsoft yang di-*release* pertama kali pada Januari 2002 dan berlisensi *open source*. *Framework* ini dibangun menggunakan CLR (*Common Language Runtime*) dan dapat menulis *code* untuk ASP.NET menggunakan bahasa pemrograman berbasis .NET seperti C# dan Visual Basic. (Fityan Aula Juyuspan : 2017)

### **II.2.8. Visual Studio 2010**

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi *Windows*, ataupun aplikasi Web. Visual Studio mencakup kompiler, *Software Development Kit* (SDK), *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. (Herpendi, 2016).



Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). (Herpendi, 2016).

Visual Studio sebelumnya versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk *platform* Microsoft *.NET Framework 3.5*. Versi sebelumnya, Visual Studio 2005 ditujukan untuk *platform .NET Framework 2.0 dan 3.0*. Visual Studio 2003. ditujukan untuk *.NET Framework 1.1*, dan Visual Studio 2002 ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998). (Herpendi, 2018).

### **II.2.9. UML (*Unified Modelling Language*)**

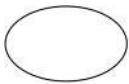

*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.




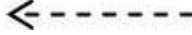
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut (Sinta Maria : 2021) :

### II.2.9.1. Use Case Diagram

Diagram use case merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. Use case digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini adalah simbol-simbol diagram use case, seperti yang terlihat pada tabel II.1 :

**Tabel II.1. Simbol Use Case Diagram**

Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukaran pesan antara unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>Use Case</i> .</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mendefinisikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem biasa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> tetapi tidak memiliki</p>

	<i>control terhadap use case.</i>
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidentifikasi bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan didalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat dipenuhi.

(Sumber : Sinta Maria : 2021)

### II.2.9.2. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol- simbol class diagram, seperti terlihat pada tabel II.2. berikut ini.

**Tabel II.2. Simbol *Class Diagram***




Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada maksimal 1
n..n	Batasan antara Contoh 2..4 mempunyai arti minimal 2 maksimum 4

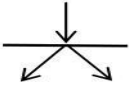
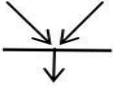


(Sumber : Sinta Maria : 2021)

### II.2.9.3. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan actor. Berikut adalah simbol-simbol *activity diagram*, seperti terlihat pada tabel II.3 :

**Tabel II.3. *Activity Diagram***

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End Point</i> , akhir aktifitas
	<i>Activities</i> menggambarkan suatu proses atau kegiatan bisnis.




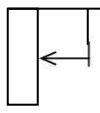


	<p><i>Fork</i> (Percabangan) digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.</p>
	<p><i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision points</i> menggambarkan pilihan untuk pengambilan keputusan, true, false</p>
	<p><i>Swimlane</i> pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa</p>

(Sumber : Sinta Maria : 2021)

#### II.2.9.4. Sequence Diagram

Diagram sekuen “menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu”. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case (Rosa dan Shalahuddin : 2018). Berikut adalah simbol-simbol activity diagram, seperti terlihat pada tabel II.4 berikut ini

**Tabel II.4. Simbol *Sequence Diagram***

Gambar	Keterangan
	<p><i>Entity Class</i>, merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal system dan menjadi landasan untuk menyusun basis data</p>
	<p><i>Boundary Class</i> berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem. Seperti tampilan <i>formentry</i> dan <i>form</i> cetak.</p>
	<p><i>Control Class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar kelas.</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 aktivitas sebuah operasi.</p>

(Sumber : Sinta Maria : 2021)