

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Penelitian Terkait**

Berikut beberapa penelitian terkait yang menjadi referensi pada penelitian ini:

(Utami, 2021) dalam penelitiannya yang berjudul “Sistem Pakar Deteksi Penyakit Diabetes Mellitus (DM) menggunakan Metode Forward chaining dan Certainty factor Berbasis Android”

Diabetes mellitus salah satu penyakit yang bisa disebut atau dikategorikan kronis disebabkan karena penyakit ini menyerang organ pankreas yang sejatinya tidak memproduksi cukup insulin bahkan ketika tubuh tidak bisa berfungsi baik lagi secara maksimal. Diabetes mellitus salah satu bentuk hambatan kekebalan tubuh yang sejatinya adalah sistem yang sangat penting bagi tubuh manusia. Hiperglikemia bentuk dalam terjadinya kenaikan dalam kadar gula darah merupakan bentuk seketika dalam hal ini yang tidak terkontrol pada dasarnya akan menimbulkan gagalnya fungsi syaraf yang seharusnya.

Dimasa pandemi Covid-19 membuat masyarakat takut atau khawatir untuk berpegiian keluar rumah ataupun ke rumah sakit maka dari itu sistem ini mempermudah masyarakat untuk mendeteksi gejala DM. Penerapan sistem pakar yang dapat digunakan dalam ekspedisi memprediksi dari berbagai tipe penyakit DM yang didasari dari berbagai gejala yang dirasakan klien. Solusi dalam hal tersebut yang sejatinya sering perlakuan tersebut dilakukan para ahli ataupun user maka hal ini didasari dalam mengetahui berbagai tingkatan penyakit DM, hal inilah

yang menjadi dasar akan dibuat sistem pakar deteksi Diabetes berbasis android dengan metode *Forward Chaining* dan *Certainty Factor*.

## **2.2 Sistem Pakar**

Sistem pakar atau *Expert System* yang biasanya juga disebut dengan *knowledge-based system* adalah suatu aplikasi komputer yang berfungsi untuk membantu dalam pengambilan sebuah keputusan atau memecahkan sebuah permasalahan / persoalan dalam bidang yang spesifik (Ricardo & Fajrin, 2022).

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah, beberapa aktivitas pemecahan masalah yang dimaksud antara lain pembuatan keputusan (*decision making*), pemanduan pengetahuan (*knowledge fusing*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), pembuatan desain (*designing*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberi nasehat (*advising*) dan pelatihan (*tutoring*) selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar.

### **2.2.1 Struktur Sistem Pakar**

Sistem pakar terdiri berdasarkan 2 bagian utama yaitu: lingkungan pengembangan “*development environment*” & lingkungan konsultasi “*consultation enviroment*”, lingkungan pengembangan dipakai menjadi pembangunan sistem pakar baik menurut segi pembangunan komponen juga basis pengetahuan. Lingkungan konsultasi dipakai seseorang yang bukan pakar untuk berkonsultasi (Adi Iswara, 2021).

Komponen-komponen yang terdapat dalam arsitektur/struktur sistem pakar sebagai berikut:

### **1. Antarmuka Pengguna “*User Interface*”**

Antarmuka adalah mekanisme yang digunakan pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pengguna dan mengubahnya menjadi bentuk yang dapat diterima oleh sistem. Antarmuka juga menerimanya dari sistem dan menampilkannya dengan cara yang dapat dimengerti oleh pengguna.

### **2. Basis Pengetahuan**

Basis data pengetahuan adalah jenis basis data yang digunakan untuk manajemen pengetahuan. Basis data ini menawarkan kemungkinan untuk mengumpulkan, mengatur, dan mengambil kembali pengetahuan yang terkomputerisasi. basis data mengandung pengetahuan untuk pemahaman, formulasi data dan penyelesaian masalah.

### **3. Akuisisi Pengetahuan “*Knowledge Acquisisi*”**

Akuisisi pengetahuan merupakan akumulasi, transfer, dan konversi pengetahuan khusus untuk memecahkan masalah berdasarkan sumber pengetahuan pada program komputer. Dalam proses ini, *knowledge engineer* mencoba menyerap pengetahuan dan lalu mentransfernya ke basis pengetahuan. Pengetahuan pakar dilengkapi dengan buku, database laporan penelitian dan pengalaman pengguna.

### **4. Mesin Inferensi “*Inference Engine*”**

Komponen ini berisi tentang pola pikir dan mekanisme penalaran yang digunakan oleh para ahli dalam pemecahan masalah. Mesin inferensi adalah program komputer yang menyediakan metodologi untuk berpikir tentang

informasi di basis pengetahuan dan di tempat kerja dan untuk merumuskan kesimpulan.

#### **5. *Workplace/Blackboard***

Workplace adalah area kumpulan "*working memory*" dari memori kerja yang digunakan untuk merekam peristiwa yang sedang berlangsung, termasuk keputusan dari waktu ke waktu.

#### **6. Fasilitas Penjelasan**

Fasilitas Penjelasan adalah komponen tambahan yang meningkatkan kapasitas sistem pakar, digunakan untuk melacak tanggapan dan memberikan penjelasan tentang perilaku sistem pakar secara interaktif melalui pertanyaan.

#### **7. Perbaikan Pengetahuan**

Para pakar mempunyai kemampuan untuk menganalisis dan menaikkan kinerja, dan kemampuan untuk belajar berdasarkan kinerja mereka. Kemampuan ini krusial pada pembelajaran terkomputerisasi sehingga program bisa menganalisis penyebab keberhasilan dan kegagalan dan juga menilai apakah pengetahuan yang terdapat masih cocok buat dipakai pada masa depan.

### **2.3 Komponen-Komponen Mobil**

Menurut Ida Bagus Dhani Satwika (2012) Komponen-komponen mobil merupakan satu kesatuan yang menunjang sistem kerja mesin mobil. Mesin bergerak karena adanya sistem pembakaran dalam (*internal combustion engine*), yang mengubah energi kimia menjadi energi mekanis dengan membakar campuran bensin dan udara, sehingga terjadilah ledakan yang dapat

menggerakkan mesin. Secara sederhana, proses kerja masing-masing komponen dalam menunjang kerja secara keseluruhan meliputi sistem pengisian dan pembakaran dalam, sistem transmisi, sistem kemudi, sistem suspensi, sistem rem, sistem kelistrikan, serta sistem pelumasan dan pendinginan (Adi Iswara, 2021).

## 2.4 Metode Penelitian

### 2.4.1 Foward Chaining

Metode foward caining adalah suatu metode dari inference engine untuk memulai penalaran suatu data dari fakta-fakta yang menuju suatu kesimpulan. *Foward Chaining* juga dapat diartikan strategi pengambilan keputusan yang dimulai dari bagian permis (fakta) menuju konkluksi (kesimpulan akhir). Pelacakan kedepan adalah pendekatan mulai dari masukan, dan selanjutnya mencoba menggambarkan kesimpulan. teknik pencarian ini dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagan *IF* dari *rule IF-Then*. Bila ada fakta yang cocok dengan bagian *IF*, maka rule tersebut dieksekusi (Zulfian & Verdi, 2017).

Langkah-langkah dalam membuat sistem pakar dengan menggunakan metode forward chaining yaitu :

- a. Pendefenisian masalah dimulai dengan pemilihan domain masalah dan akuisi pengetahuan
- b. Pendefenesian data input untuk memulai inferensi karena diperlukan oleh sistem forward chaining.
- c. Pendefenisian struktur pengendalian data untuk membantu mengendalikan pengaktifan suatu aturan.
- d. Penulisan kode awal dalam domain pengetahuan

- e. Pengujian sistem agar dapat mengetahui sejauh mana sistem berjalan
- f. Perancangan antarmuka dengan basis pengetahuan
- g. Pengembangan system
- h. Evaluasi sistem

#### 2.4.2 *Certainty factor*

Definisi menurut David McAllister *Certainty factor* (faktor kepastian) adalah suatu metode untuk membuktikan apakah suatu fakta itu pasti atau tidak pasti yang terbentuk metric yang biasanya digunakan dalam sistem pakar. teori *certainty factor* (CF) diusulkan oleh shortliffe dan buchanan pada tahun 1975 untuk mengakomodasi kepastian pemikiran (*inexact reasoning*) seorang pakar. seorang pakar, misalnya dokter menganalisis informasi yang ada seperti ungkapan “mungkin”, “kemungkinan besar”, “hampir seperti” (Zulfian & Verdi, 2017).

Certainty factor menunjukkan ukuran kepastian terhadap suatu fakta atau aturan [10].

$$CF[h,e] = MB[h,e] - MD[h,e] \dots (1)$$

Keterangan : CF[h,e] = Faktor kepastian

MB[h,e] = Measure of belief, ukuran kepercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan evidence (e) antara 0 dan 1

MD[h,e] = Measure of disbelief, ukuran ketidakpercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan evidence (e) antara 0 dan 1. Adapun beberapa

kombinasi certainty factor terhadap premis tertentu:

1. Certainty factor dengan satu premis.

$$CF[h,e] = CF[e] * CF[rule] = CF[user] * CF[pakar]$$

2. Certainty factor dengan lebih dari satu premis.

$$CF[A \wedge B] = \text{Min}(CF[a], CF[b]) * CF[\text{rule}]$$

$$CF[A \vee B] = \text{Max}(CF[a], CF[b]) * CF[\text{rule}]$$

3. Certainty factor dengan kesimpulan yang serupa.

$$CF \text{ gabungan } [CF1, CF2] = CF1 + CF2 * (1 - CF1)$$

Kelebihan dari metode ini adalah cocok digunakan pada sistem pakar yang mengukur sesuatu yang pasti atau tidak pasti seperti mendiagnosis penyakit dan perhitungan dari metode ini hanya berlaku untuk sekali hitung, serta hanya dapat mengolah

## 2.5 Pemrograman Aplikasi

### 2.5.1 Android

*Android* adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis *Linux*. *Android* menyediakan *platform* terbuka (*open source*) sehingga memudahkan bagi para pengembang untuk menciptakan aplikasi mereka sendiri *android* awalnya dikembangkan oleh *android, Inc.*, dengan dukungan *Google*, yang kemudian *android* dibeli oleh *google* pada tahun 2005 (Muslihudin, 2018).

Menurut Hermawan (2011), *Android* merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, *i-Phone OS*, *Symbian*, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk *platform* mereka (Husain, 2017).

### **2.5.1.1 Android SDK (Software Development Kit)**

*Android SDK* adalah tools API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform *Android*, *Android SDK* menggunakan bahasa pemrograman *Java*, untuk lebih mengembangkan aplikasi *Android*, *Android SDK* disediakan untuk sistem operasi *Mac Os X10.4.8* atau lebih, *Windows XP*, *Vista*, 7 dan 8, serta *Linux* yang telah di realese oleh pihak *Google*. 2.2.3 (Winandar, 2019).

### **2.5.1.2 Android Development Tool**

*Android Development Tools* adalah *plugin* yang didesain untuk *IDE Eclipse* yang memberikan kita kemudahan dalam mengembangkan aplikasi *Android*. Dengan adanya ADT untuk *eclipse* akan memudahkan pengembangan dalam membuat aplikasi *project Android*, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan *Android SDK* melalui *Eclipse*. Dengan ADT juga kita dapat membuat package *Android (.apk)* yang digunakan untuk distribusi aplikasi *Android* yang kita rancang. 2.2.4 (Winandar, 2019).

## **2.5.2 Data**

Data adalah fakta kasar atau gambaran yang dikumpulkan dari keadaan tertentu. Data adalah hal yang merujuk pada fakta-fakta baik berupa angka, teks, dokumen, gambar, bagan, suara yang mewakili deskriptif verbal atau kode tertentu (Iwan Purnama dkk, 2020).

### **2.5.2.1 Database**

*Database* atau Basis Data yaitu kumpulan dari data-data yang membentuk suatu berkas atau file yang saling berhubungan dengan tatacara yang tertentu untuk

membentuk data baru atau informasi. Atau bisa diartikan *Database* atau Basis Data merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu (Apyliyana, 2021).

### 2.5.3 *MySQL*

Definisi *MySQL* merupakan software RDMS (*Relational Database Management System*) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak pengguna dan dapat melakukan suatu proses secara sinkron atau bersamaan (Rahmasari, 2019).

### 2.5.4 *Visual Studio Code*

*Visual Studio Code (VS Code)* ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux, Mac, dan Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript, Typescript, dan Node.js*, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace *Visual Studio Code (seperti C++, C#, Python, Go, Java, dst)*. Banyak sekali fitur-fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense, Git Integration, Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *Visual Studio Code*. Pembaruan versi *Visual Studio Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan teks editor-teks editor yang lain. Teks editor *VS Code* juga bersifat open source, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Kode

sumber dari *VS Code* ini pun dapat dilihat di link Github. Hal ini juga yang membuat *VS Code* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *VS Code* ke depannya (Gligorijevic, 2019).

### **2.5.5 PHP**

PHP merupakan bahasa pemrograman yang ditujukan untuk membuat program *website* dimana kode program yang telah dibuat di kompilasi dan dijalankan pada sisi Jurnal server untuk menghasilkan halaman *website* yang dinamis. Pada awalnya PHP didirikan untuk *Personal Home Page*. Karena memiliki banyak manfaat dan dapat berkembang dengan baik maka PHP disebut dengan *hypertext Preprocessor* (Guzmaliza, 2019).

### **2.5.6 Java**

*Java* merupakan salah satu Bahasa pemrograman yang populer saat ini. *Java* dipelopori oleh James Gosling yang merupakan *engineer* di *Sun Microsystem*. *Java* juga dikenal memiliki moto "***Write Once, Run Anywhere***". Artinya, *Java* mampu dijalankan di berbagai *platform* tanpa perlu disusun ulang menyesuaikan *platformnya*. Misalnya, berjalan di *Android*, *Linux*, *Windows*, dan lainnya (Mukharil, Adam & Firman, 2018).

### **2.5.7 XML**

XML adalah sebuah database yang mampu mengolah data. *Database* yang mampu mengolah data-data dan bisa mengakomodasikan data-data pada XML adalah *XML Database Xindice*. *Database* ini merupakan sebuah *database* yang melakukan system basis data yang dapat mengakomodasikan data-data XML yang digunakan untuk memudahkan dalam pengolahan dan penyimpanan data XML,

kemudian data XML ini diolah menggunakan proses enkripsi dan deskripsi dengan menggunakan sistem teknik kriptografi (Santoso & Iskandar, 2020).

### **2.5.8 XAMPP**

XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi (Nugroho & Rohimi, 2020).

### **2.5.9 PhpMyAdmin**

*PhpMyAdmin* adalah sebuah aplikasi atau perangkat berbasis *opensource* yang bisa kita gunakan secara gratis untuk melakukan pemrograman ataupun administrasi pada *database* MySQL. *PhpMyAdmin* sendiri menggunakan bahasa PHP untuk pemrogramannya, selain itu *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perizinan (*permissions*), dan lain-lain. Jadi dapat disimpulkan pula bahwa *phpMyAdmin* berbeda dengan MySQL. Dimana *PhpMyAdmin* digunakan sebagai alat yang memudahkan dalam pengoperasian *database* MySQL, sedangkan MySQL adalah suatu *database* itu sendiri, dimana *database* berfungsi sebagai penyimpanan data (Ramadhan & Mukhaiyar, 2020).

### **2.5.10 Internet**

Menurut Restianti(2010, h. 25), internet diartikan sebagai jaringan komputer yang luas dan mendunia, yaitu menghubungkan pemakai komputer dari suatu negara ke negara lain di seluruh dunia, yang di dalamnya terdapat berbagai sumber daya informasi dari mulai yang statis hingga yang dinamis dan interaktif.

Lebih lanjut, Rusman (2013) mengemukakan bahwa internet merupakan bentuk penerapan teknologi informasi yang ditunjukkan untuk mempermudah proses pembelajaran yang dikemas dalam bentuk digital konten dan pelaksanaannya membutuhkan sarana komputer yang dikoneksikan ke internet (Rahman, 2021).

### **2.5.11 WEB**

*World Wide Web (WWW)* atau sering disebut Web merupakan salah satu sumber daya internet yang berkembang pesat. Informasi Web didistribusikan dengan pendekatan *hyperlink* yang memungkinkan seseorang memperoleh informasi dengan meloncat dari satu halaman ke halaman lain (Juliany, 2018).

### **2.5.12 Bootstrap**

*Bootstrap* adalah sebuah *framework* CSS twitter yang menyediakan komponen-komponen antarmuka siap pakai dan telah dirancang sedemikian rupa untuk keperluan desain halaman *website* yang artistik. Selain komponen-komponen dasar untuk membangun antarmuka, *Bootstrap* juga menyediakan grid 12 kolom yang dapat menunjang kinerja dalam mendesain layout yang rapi. Sebagai CSS *framework*, *Bootstrap* tergolong paket lengkap. Teknologi HTML, CSS, dan *JavaScript* yang ada pada *Bootstrap*, tidak hanya dapat melakukan *7 styling* dengan CSS, akan tetapi juga dapat menggunakan komponen-komponen seperti ikon, tombol, dan navigasi dengan desain unik khas *Bootstrap* (Muharam & Reynaldi, 2022).

### **2.5.13 CSS**

*Cascading Style Sheet* atau yang lebih dikenal dengan CSS adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna dan semua yang berhubungan dengan tampilan, Pada

umumnya CSS digunakan untuk memformat halaman web yang ditulis dengan HTML (Suryana, 2015). *Cascading Style Sheet* (CSS) memiliki arti gaya menata halaman bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak yang telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya (Sibero, 2015). CSS dapat dilakukan pada satu atau lebih halaman HTML untuk mendefinisikan style beberapa halaman HTML, biasanya CSS disimpan didalam satu file tersendiri di luar file HTML. Hal ini memudahkan web desainer untuk melakukan perubahan *style*, sehingga untuk merubah tampilan beberapa halaman HTML cukup dengan merubah satu file CSS (Rilvany & Purnomo, 2020).

#### **2.5.14 HTML**

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML dikembangkan pertama kali oleh Tim Berners-Lee bersamaan dengan *protocol* HTTP (*Hypertext Transfer Protocol*) pada tahun 1989. Tujuan utama pengembangan HTML adalah untuk menghubungkan suatu halaman *web* dengan halaman *web* lainnya. Tentunya pada awal pengembangannya (Solihin, 2016) HTML merupakan salah satu format yang digunakan dalam pembuatan dokumen atau aplikasi yang berjalan di halaman web (Arif, 2011) HTML itu adalah bahasa yang digunakan untuk menulis halaman *web*, biasanya menggunakan ekstensi .htm, .html atau .shtml (Suyanto, 2017).

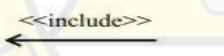
#### **2.6 Diagram UML**

Menurut Mulyani (2017: 42) "*Unified Modeling Language*" adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem."

### 2.6.1 Use Case Diagram

Use Case Diagram merupakan salah satu diagram yang digunakan untuk menggambarkan relasi antara sistem dengan aktor. Diagram ini hanya menggambarkan secara umum maka elemen yang digunakan cukup sedikit (Mulyani, 2017). Berikut ini elemen-elemen yang digunakan pada *use case* diagram.

Tabel 2.1 Use Case Diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Sumber: Mulyani (2017)

### 2.6.2 Activity Diagram

Menurut Mulyani (2017) “*Activity Diagram* merupakan diagram yang digunakan sebagai penggambaran aktivitas pada proses, logika, dan hubungan antar aktor dengan alur-alur kerja *use case*.” Berikut ini elemen- elemen yang digunakan pada *activity diagram*:

Tabel 2. 2 Activity Diagram

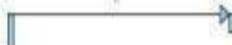
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Desicion</i>	Digunakan untuk menggambarkan suatu keputusan/ tindakan yang harus diambil pada kondisi tertentu

Sumber: Mulyani (2017)

### 2.6.3 Sequence Diagram

Menurut Rahardi (2016) “*Sequence Diagram* merupakan suatu penyajian perilaku yang disusun seperti rangkaian langkah-langkah percontohan dari waktu ke waktu.” *Sequence diagram* digunakan untuk menggambarkan alur pekerjaan, pesan yang disampaikan dan menggambarkan elemen-elemen di dalamnya bekerja sama untuk mencapai suatu hasil. Berikut ini elemen-elemen yang digunakan pada *sequence diagram*:

Tabel 2. 3 *Sequence Diagram*

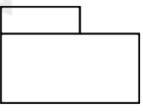
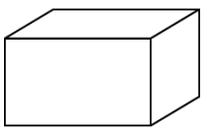
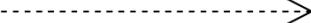
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
4		<i>Self Message</i>	Spesifikasi dari komunikasi untuk menunjukkan kegiatan memuat proses informasi pada aktifitas sendiri.

Sumber: Mulyani (2017)

#### 2.6.4 *Deployment Diagram*

*Deployment Diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi dan juga digunakan untuk memodelkan hal-hal seperti sistem tambahan dan sistem *client/server* (Rosa dan Shalahuddin, 2011). Berikut ini elemen-elemen yang digunakan pada *deployment diagram*:

Tabel 2. 4 *Deployment Diagram*

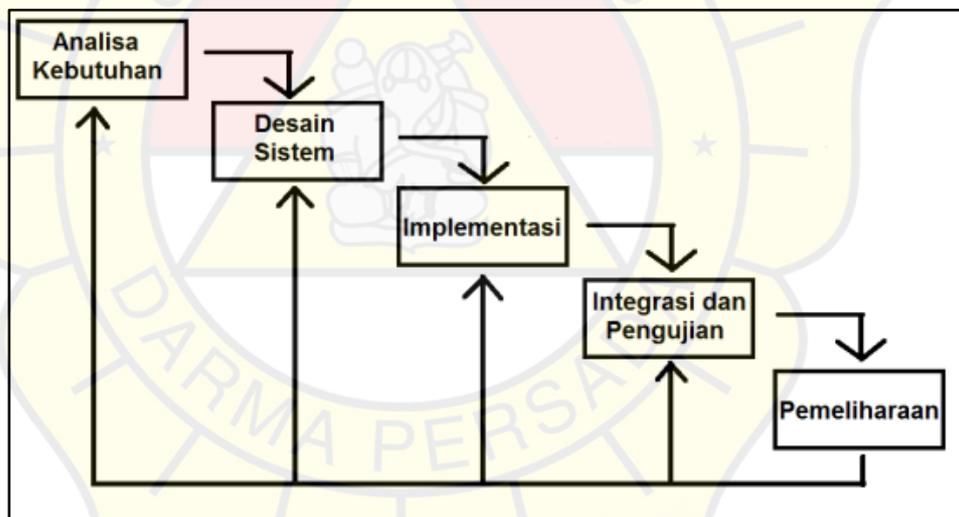
Penjelasan	Notasi UML
<b>Package:</b> Merupakan sebuah bungkusuan dari satu <i>node</i> atau lebih.	
<b>Node:</b> Berisi <i>hardware</i> , <i>software</i> dan komponen rancangan.	
<b>Ketergantungan/dependency:</b> Ketergantungan antar <i>node</i> .	

<b>Link:</b> Relasi antar <i>node</i> .	_____
---	-------

Sumber: Mulyani (2017)

## 2.7 Metode Pengembangan Sistem

Model *waterfall* adalah model yang paling banyak digunakan untuk tahap pengembangan. Model *waterfall* ini juga dikenal dengan nama model tradisional atau model klasik. Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*Classic cycle*)". Model air terjun ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*) (Susilo, 2018).



Sumber: Susilo (2018)

Gambar 2.1 *Waterfall*

Adapun penjelasan urutan dari tahapan-tahapan yang dimiliki metodologi *waterfall* adalah sebagai berikut:

### 1. Analisis Kebutuhan

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian,

wawancara atau study literatur. Seseorang system analisis akan menggali informasi sebanyak banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman.

## **2. Desain**

Proses design akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

## **3. Penulisan Sinkode Program atau Implementasi**

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan

kesalahankesalahan terhadap system tersebut dan kemudian bisa diperbaiki.

#### **4. Pengujian Program**

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user.

#### **5. Pemeliharaan**

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau system operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsi.



**TEKNOLOGI INFORMASI UNIVERSITAS  
DARMA PERSADA**