



## **BAB II**

# **TINJAUAN PUSTAKA**

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Menurut Okta Lesva, Yadi (2019) dengan judul penelitian “Sistem Monitoring Kesehatan Masyarakat Berbasis *Web*”. Penelitian ini menggunakan sistem dengan *pemrograman PHP dan MySQL* dan berbasis *web*. Sistem ini menggunakan metode *Web Engineering*, hasil dari penelitian ini menunjukkan bahwa sistem monitoring yang diaplikasikan kedalam teknologi *web* dapat memberikan kemudahan bagi staf Puskesmas Kecamatan Pajar Bulan dalam pengolahan data pasien. Sistem yang dapat memonitoring data penyakit yang sedang mewabah ini dapat meningkatkan pelayanan kepada pasien dan masyarakat dalam memberikan informasi. Tujuan dari penelitian ini adalah membuat aplikasi sistem monitoring kesehatan masyarakat berbasis *web* menggunakan metode *Web Engineering* dengan bahasa pemrograman *PHP*.

Menurut Zulfa Afifah Shibghotallah (2014) dengan judul penelitian “Sistem Pakar Pemilihan Obat Pada Pasien Hipertensi Berbasis Web Menggunakan Metode Certainty Factor (Study Kasus : Puskesmas Mantrijeron Yogyakarta)”. Penelitian ini dilakukan menggunakan metode *Certainty Factor*. Dari penelitian menggunakan *Certainty Factor* ini, didapatkan hasil bahwa dapat mempermudah pakar atau dokter dalam menemukan obat yang tepat terhadap pasien berdasarkan data dari gejala-gejala yang timbul.

Berdasarkan penelitian yang dilakukan Ayu, dkk (2017) dimana melakukan penelitian mengenai “Implementasi Program Pengelolaan Penyakit Kronis”. Hasilnya menunjukkan bahwa dengan menggunakan sistem konvensional yang sudah ada terbukti banyak terjadi kesalahan kesalahan yang dapat menyebabkan *miscommunication* antara petugas pelaksana dengan pasien, bahkan antara petugas pelaksana dengan pimpinan. Meskipun telah dilaksanakan nya kegiatan Program Pengelolaan Penyakit Kronis (PROLANIS) sesuai Standar Operasional Prosedur (SOP) tetapi masih terdapat indikator-indikator yang tidak terpenuhi dalam sistem konvensional tersebut sehingga timbul permasalahan yang akan menghasilkan dampak yang negatif sehingga pelaksanaan kegiatan pelayanan terhadap Program Pengelolaan Penyakit Kronis (PROLANIS) tidak dapat berjalan secara efisien.

Menurut Falah, dkk (2019) dengan judul penelitian “Pengembangan Sistem Informasi Monitoring Kesehatan Pasien TB Rawat Jalan Rumah Sakit Al Islam Bandung”. Penelitian ini dilakukan dengan menggunakan teknologi berbasis *Web*. Dari penelitian ini, didapatkan hasil bahwa sistem informasi monitoring berbasis *web* ini dapat mengelola seluruh data yang berhubungan dengan aktifitas kesehatan pasien TB rawat jalan sehingga dapat mempermudah bagi instansi medis untuk melakukan pemantauan terhadap status perkembangan kesehatan pasien TB serta dapat membantu untuk mengetahui informasi mengenai keteraturan pasien TB rawat jalan dalam mengkonsumsi obat-obatan. Berdasarkan penelitian yang dilakukan Simeon Praska Lukestiwa (2019) dimana melakukan penelitian mengenai “Sistem Pakar Pemilihan Produk Obat Batuk Berbasis Web Menggunakan Metode Certainty Factor”. Hasilnya menunjukkan bahwa sistem yang

berjalan sangat bermanfaat kepada pasien dalam menentukan obat batuk yang akan di konsumsi berdasarkan jenis obat-obatan yang dapat meminimalisir gejala yang timbul, hal ini disetujui oleh responden yang terdiri dari para nakes dan pasien.

## **II.2. Studi Literatur**

### **II.2.1. Aplikasi**

Aplikasi adalah program yang dibuat oleh manusia yang berfungsi untuk menyelesaikan permasalahan-permasalahan masalah yang akan dihadapi. (Zulfauzi ; 2015 : 57).

Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer atau suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Aplikasi-aplikasi dalam suatu paket biasanya memiliki antar muka sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. (Sitohang ; 2017 : 2).

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

Sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis computer (*Computer Based Information Systems* atau CBIS). Istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam

kenyataannya komputer merupakan bagian yang penting. Ada beberapa macam definisi sistem informasi yaitu menurut Alter (1992) Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Menurut Bornar dan Hopwood (1993) Sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna. Menurut Gelinas, Oram, dan Wiggins (1990) Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimbau, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai (Kadir, 2013).

### **II.2.3. Monitoring**

Monitoring adalah langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan (Sutabri, 2012).

### **II.2.4. Program Pengelolaan Penyakit Kronis (PROLANIS)**

Program Pengelolaan Penyakit Kronis (PROLANIS) adalah salah satu upaya program yang bertujuan untuk memberikan perhatian khusus kepada para pasien yang mengidap penyakit kronis. Para pasien dipantau kesehatannya dengan cara diberikannya obat-obat tertentu yang dapat menurunkan gejala sehingga para pasien yang mengidap penyakit kronis dapat terjaga kondisinya dengan baik.

### **II.2.5. Website**

*Website* adalah suatu halaman web yang saling berhubungan yang umumnya berisikan kumpulan informasi berupa data teks, gambar, animasi, audio, video maupun gabungan dari semuanya yang biasanya dibuat untuk personal, organisasi dan perusahaan. *Website* dibedakan menjadi dua yaitu bersifat statis dan dinamis. Bersifat statis apabila isi informasinya tetap dan isi informasinya hanya dari pemilik *website*, sedangkan *website* bersifat dinamis apabila isi informasinya selalu berubah-ubah dan dapat diubah oleh pemilik maupun pengguna *website*. Contoh *website* statis adalah *website profile* perusahaan, sedangkan contoh *website* dinamis seperti *facebook*, *twitter* dan lain-lain.

### **II.2.6. Metode *Forward Chaining***

Metode *Forward Chaining* adalah suatu metode pengambilan keputusan yang umum digunakan dalam system pakar. Proses pencarian dengan metode *Forward Chaining* berangkat dari kiri ke kanan, yaitu dari premis menuju kepada kesimpulan akhir, metode ini sering disebut data driven yaitu pencarian dikendalikan oleh data yang diberikan. (Hartati, S. dan Iswanti 2008).

Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi. Pertama-tama, sistem mencari semua aturan yang kondisinya terdapat di memori kerja, kemudian memilih salah satunya dan menjalankan aksi yang bersesuaian dengan aturan tersebut.

Pemilihan aturan yang akan dijalankan berdasarkan strategi tetap yang disebut strategi penyelesain konflik. Aksi tersebut menghasilkan memori kerja baru dan siklus diulangi lagi sampai tidak ada aturan yang dapat dipicu, atau tujuan yang

dikehendaki sudah terpenuhi anda dapat melihat di Tabel 2.1 Contoh aturan menggunakan *forward chaining*.

**Tabel II.1 Contoh aturan menggunakan forward chaining**

No	Aturan
R1	<i>IF A &amp; B THEN C</i>
R2	<i>IF C THEN D</i>
R3	<i>IF A &amp; E THEN F</i>
R4	<i>IF A THEN G</i>
R5	<i>IF F &amp; G THEN D</i>
R6	<i>IF G &amp; E THEN H</i>
R7	<i>IF C &amp; H THEN I</i>
R8	<i>IF I &amp; A THEN J</i>
R9	<i>IF G THEN J</i>
R10	<i>IF J THEN K</i>

Pada Tabel II.1 Contoh aturan menggunakan forward chaining. Terlihat ada 10 aturan yang tersimpan dalam basis pengetahuan. Jika fakta awal yang diberikan hanya: A dan F (artinya: A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis: K). Langkah-langkah inferensi adalah sebagai berikut:

1. Dimulai dari R-1, A merupakan fakta sehingga bernilai benar, sedangkan B belum bisa diketahui kebenarannya, sehingga C pun juga belum bisa diketahui kebenarannya. Oleh karena itu tidak didapatkan informasi apapun pada R1 ini. Sehingga kita menuju ke R2.
2. Pada R2 tidak diketahui informasi apapun tentang C, sehingga tidak bisa dipastikan kebenaran D. Oleh karena itu tidak didapatkan informasi apapun pada R1 ini. Sehingga harus menuju ke R3.

3. Pada R3, baik A maupun E adalah fakta sehingga jelas benar. Dengan demikian F sebagai konsekuen juga ikut benar. Sehingga sekarang terdapat fakta baru yaitu F. Karena F bukan hipotesis yang hendak dibuktikan (= K) maka penelusuran dilanjutkan ke R4.

4. Pada R4, A adalah fakta sehingga jelas benar. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang didapatkan fakta baru yaitu G. Karena G bukan hipotesis yang hendak dibuktikan (= K), maka penelusuran dilanjutkan ke R5.

5. Pada R5, baik F maupun G bernilai benar berdasarkan aturan R3 dan R4. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang terdapat fakta baru yaitu D. Karena D bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R6.

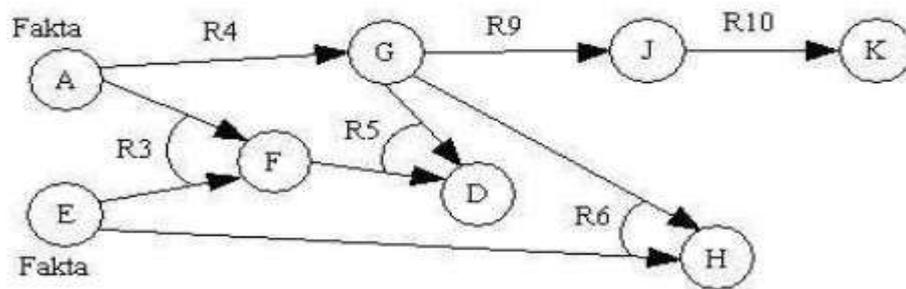
6. Pada R6, baik A maupun G adalah benar berdasarkan fakta dari R4. Dengan demikian H sebagai konsekuen juga ikut benar. Sehingga sekarang terdapat fakta baru yaitu H. Karena H bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R7.

7. Pada R7, meskipun H benar berdasarkan R6, namun tidak diketahui kebenaran C sehingga, I pun juga belum bisa diketahui kebenarannya. Oleh karena itu tidak didapatkan informasi apapun pada R7 ini. Sehingga dilanjutkan menuju ke R8.

8. Pada R8, meskipun A benar karena fakta, namun tidak diketahui kebenaran I, sehingga J pun juga belum bisa diketahui kebenarannya, oleh karena itu tidak didapatkan informasi apapun pada R8 ini. Sehingga dilanjutkan menuju ke R9.

9. Pada R9, J bernilai benar karena G benar berdasarkan R4. Karena J bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R10.

10. Pada R10, K bernilai benar karena J benar berdasarkan R9. Karena K sudah merupakan hipotesis yang hendak dibuktikan, maka terbukti bahwa K adalah benar.



**Gambar II.1 Alur *Forward Chaining***

### II.2.7. Metode *Certainty Factor*

Menurut David McAllister, Faktor Kepastian (*Certainty factor*) merupakan suatu metode untuk membuktikan suatu kepercayaan dalam suatu fakta berdasarkan bukti atau penilaian pakar. *Certainty factor* pertama kali digunakan pada Sistem Pakar yang berfungsi untuk mendeteksi, notasi *Certainty Factor* didefinisikan sebagai :

$$CF[H,E] = MB [H,E] - MD [H,E]$$

Keterangan :

CF [H,E] : *Certainty Factor* dari hipotesa yang dipengaruhi oleh *evidence* E

MB [H,E] : *Measure of Belief* terhadap hipotesa H

MD [H,E] : *Measure of Belief* terhadap hipotesa H

Nilai Certainty Factor dapat didefinisikan dengan rentang nilai :

$$-1 \leq CF [H,E] \leq +1$$

Keterangan :

Nilai akan 0 jika tidak ada evidence

Nilai yang lebih dari 0 adalah hipotesa

Nilai yang lebih kecil dari 0 adalah negasi dari hipotesa

Untuk mendapatkan nilai CF [H,E] *Certainty Factor* dapat menggunakan pertanyaan untuk menentukan tingkat kepercayaan (MB [H,E]) dan pertanyaan untuk menentukan tingkat ketidakpercayaan (MD [H,E]). Nilai *Certainty Term* untuk MB [H,E] dan MD [H,E] adalah :

CF[H,E] = 1,0 Pasti

CF [H,E] > 0,8 : Hampir Pasti

CF [H,E] > 0,5 : Kemungkinan Besar

CF [H,E] > 0,2 : Mungkin

CF [H,E] > 0,0 : Tidak Tahu

CF [H,E] < 0,2 : Mungkin Tidak

CF [H,E] < 0,5 : Kemungkinan Besar Tidak

CF [H,E] < 0,8 : Hampir Pasti Tidak

CF [H,E] = -1,0 : Pasti Tidak

Dengan nilai di atas, seorang pakar mampu memberikan nilai Certainty Factor terhadap suatu aturan.

### **II.2.8. Web Server**

*Web server* adalah *server* yang mampu menangani web atau permintaan HTTP (Kusumo, 2004) . *Web server* menunggu permintaan dari *client* yang menggunakan *browser* lainnya. Jika ada permintaan dari browser, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*.

### **II.2.9. XAMPP**

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *web server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (bermakna *cross* yang melambangkan *XAMPP* dapat dipakai disistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*.

### **II.2.10. MySQL**

*My Structured Query Language* (*MySQL*) merupakan salah satu perangkat lunak sistem pengelola basis data *Data Base Management System* (*DBMS*). *MySQL* merupakan sebuah hubungan *DBMS* yang membantu sebuah model data yang terdiri atas kumpulan hubungan nama (*namedrelation*). *Database MySQL* adalah salah satu database yang *opensource*. *Database* ini banyak dipasangkan dengan *script PHP*. (Steven, 2015 : 17).

### II.2.11. PHP (*Hypertext Preprocessor*)

PHP merupakan bahasa *scripting* yang berjalan di sisi *server* (*server-side*). Semua perintah yang ditulis akan dieksekusi oleh *server* dan hasil jadinya dapat dilihat melalui *browser*. Saat ini PHP versi 4 sudah di-*release* di pasaran. Mengikuti jejak kesuksesan versi sebelumnya, PHP 3. Selain dapat digunakan untuk berbagai sistem operasi, koneksi *database* yang sangat mudah menyebabkan bahasa *scripting* ini digemari para *programmer web*. Beberapa perintah PHP yang dipelajari sebatas pada perintah untuk menampilkan *tag-tag wml*, akses *database MySQL* dan pengiriman *email*. (Urfa dan Siregar, 2015 : 93).

PHP adalah *script* yang digunakan untuk membuat halaman *web* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan. (Eka, tias : 2016 : 130).

### II.2.12. HTML

*Hyper Text Markup Language* atau HTML adalah bahasa standar yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan dengan HTML yaitu: mengatur tampilan dari halaman *web* dan isinya, membuat *table* dalam halaman *web*, mempublikasikan halaman *web* secara *online*, membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*, menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*, serta menampilkan area gambar (*canvas*) di *browser*. (Eka, tias : 2016 : 130).

### II.2.13. *Sublime Text*

Sublime Text Editor adalah editor teks untuk berbagai bahasa pemrograman termasuk pemrograman PHP. *Sublime Text Editor* merupakan *editor text* lintas *platform* dengan *Python application programming interface (API)*. *Sublime Text Editor* juga mendukung banyak bahasa pemrograman dan bahasa *markup*, dan fungsinya dapat ditambah dengan *plugin*.

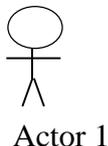
### II.2.14. UML(*Unified Modeling Language*)

*Unified Modeling language (UML)* merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek. (Sulianta, 2017).

#### II.2.14.1. *Use case Diagram*

*Use Case Diagram* merupakan diagram yang harus dibuat pertama kali saat pemodelan perangkat lunak berorientasi objek dilakukan. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.2 dibawah ini :

**Tabel II.2. Simbol *Use Case***

Simbol	Nama	Keterangan
	Aktor	Merupakan Penggunaan dari sistem. Penamaan aktor menggunakan kata benda.

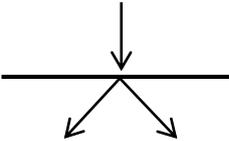
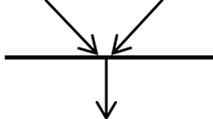
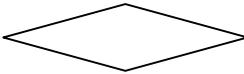
	<i>Use Case</i>	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> dengan kata kerja.
-End1 –End2 	Asosiasi	Hubungan antara aktor dengan <i>use case</i>
<<use>> 	<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.
<<extends>> 	Extends	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.

(Sumber : Sulianta ; 2017)

### II.2.14.2. Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini :

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

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
 New Swimline	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini; 2016)

### II.14.3. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Simbol yang digunakan untuk membuat *Class Diagram* terlihat pada tabel II.4 dibawah ini :

**Tabel II.4. Multiplicity Class Diagram**

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

(Sumber : Gellysa Urva, dkk; 2015 : 95)

#### II.14.4. Sequence Diagram

*Sequence Diagram* adalah diagram yang dibuat untuk mengetahui alur dari interaksi antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.5 dibawah ini :

**Tabel II.5. Simbol Sequence Diagram**

Simbol	Nama	Keterangan
<u>Object 1</u>	Objek/aktor	Sebuah objek yang berasal dari kelas. Atau dapat dinamai dengan kelasnya saja. Aktor termasuk objek. Garis putus-putus menunjukkan garis hidup suatu objek.
	Aktivasi	Menunjukkan masa hidup dari objek
Message 1	Pesan	Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. Interaksi antar objek ditunjukkan pada bagian operasi pada diagram kelas.
Message 2	Return	Pesan kembalian dari komunikasi antar objek.

(Sumber : Sulianta; 2017 )