

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini dilakukan tidak terlepas dari hasil penelitian terdahulu yang pernah dilakukan. Penelitian terdahulu digunakan sebagai bahan kajian dan perbandingan. Penelitian – penelitian yang digunakan masih ada kaitannya dengan pembuatan sistem pakar perawatan sepatu dengan metode *forward chaining*.

Penelitian dari Yasmiyati (2017) dengan judul Sistem Pakar *Diagnosis Penyakit Pada Perokok Dengan Metode Forward Chaining Berbasis Web*. Penelitian tersebut membahas tentang *diagnosis* penyakit pada perokok yang memberikan informasi kepada pengguna agar mengetahui penyakit-penyakit yang ditimbulkan karena merokok berdasarkan gejala yang timbul. Sistem pakar diharapkan dapat mengurangi jumlah perokok. Aplikasi ini dapat memberikan informasi tentang penyakit yang ada dengan presentase sebesar 89,2%. Informasi yang diberikan aplikasi ini memuat tentang informasi penyakit, gejala awal penyakit, dan pengobatan penyakit. Dalam penelitian ini, terdapat 8 penyakit dan 22 gejala Penelitian ini menggunakan bahasa pemrograman PHP. Metode yang digunakan dalam penelitian ini menggunakan *Forward Chaining* dan berbasis *website*.

Penelitian dari Dharmaningrum (2018) dengan judul Implementasi Metode Forward Chaining Untuk Deteksi Dini Gangguan Bipolar Pada Remaja Berbasis Web. Penelitian tersebut membahas tentang sebuah sistem pakar yang digunakan untuk mengetahui deteksi dini gangguan bipolar pada remaja. Dalam penelitian ini, terdapat 2 gangguan dan 19 gejala. Penelitian ini menggunakan bahasa pemrograman PHP. Sistem pakar ini menggunakan metode *Forward Chaining* dan berbasis *website*.

Penelitian dari Turang (2018) dengan judul Aplikasi Sistem Pakar Mendiagnosa Penyakit Syaraf Pusat Dengan Metode *Forward Chaining*. Penelitian tersebut membahas tentang membantu memberikan informasi tentang jenis penyakit syaraf pusat kepada pengguna dan memberikan cara pencegahan

serta pengobatannya. Di penelitian ini, terdapat 12 penyakit dan 44 gejala. Penelitian ini menggunakan bahasa pemrograman PHP. Sistem pakar ini dibuat menggunakan metode *Forward Chaining* yang berbasis *website*.

2.2 Kerangka Pemikiran

Gambar 2.1 merupakan kerangka yang digunakan dalam penelitian ini. Keterangan dari Gambar 2.1 adalah sebagai berikut :

1. Latar Belakang Masalah

Informasi yang minim tentang perawatan sepatu mengakibatkan banyak dari masyarakat yang tidak mengetahui cara merawat sepatu yang sesuai dengan bahan sepatu. Jika merawat sepatu tidak sesuai dengan material sepatu maka dapat merusak sepatu itu sendiri.

2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalahnya adalah bagaimana membuat sistem pakar perawatan sepatu berbasis *Website*?

3. Penguasaan dasar PHP dan MySQL serta Pengumpulan data

Penguasaan dasar PHP dan MySQL merupakan suatu kunci terciptanya sistem pakar perawatan sepatu tersebut, karena pembuatan sistem ini menggunakan bahasa pemrograman PHP dan menggunakan database MySQL sebagai media penyimpan data.

Pengumpulan data

Tahap pengumpulan data pada penelitian ini melalui observasi, wawancara dan studi literatur. Pengumpulan data bertujuan untuk mengetahui permasalahan dan kebutuhan informasi mengenai perawatan sepatu.

4. Analisis dan perancangan sistem

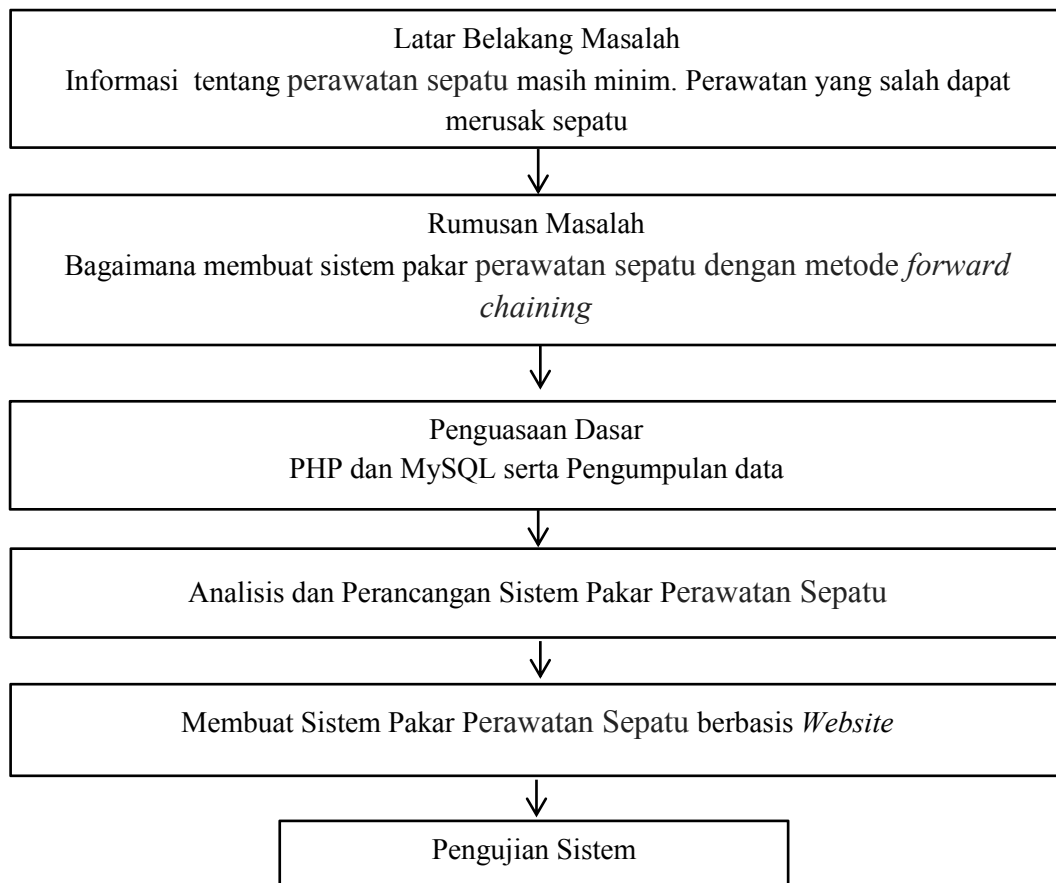
Analisis dan perancangan sistem pakar perawatan sepatu bertujuan untuk mengetahui kesiapan data dan rancangan dari aplikasi sistem pakar perawatan sepatu yang diharapkan dapat memberikan suatu informasi kepada pengguna. Perancangan sistem pada penelitian ini menggunakan UML (*Unified Modeling Language*).

5. Rancang Bangun Sistem

Membuat sistem pakar perawatan sepatu menggunakan bahasa pemrograman PHP dan *database* MySQL sebagai media penyimpanan datanya.

6. Pengujian Sistem

Pengujian sistem merupakan tahap setelah sistem berhasil dibuat. Pengujian sistem bertujuan untuk mengetahui kelebihan, kelemahan serta mengetahui kelayakan suatu sistem untuk digunakan. Pengujian sistem pada penelitian ini menggunakan metode *BlackBox*.



Gambar 2.1 Kerangka Pemikiran

2.3 Landasan Teori

2.3.1 Sistem

Sistem merupakan sebuah tatanan (keterpaduan), sekelompok unsur atau elemen yang berhubungan satu dengan yang lain untuk mencapai suatu tujuan (Rahmawati, 2017).

2.3.2 Sistem Pakar

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia (pakar) ke komputer, sehingga komputer dapat menyelesaikan permasalahan tersebut layaknya seorang pakar (Kusumadewi, 2003).

Menurut Kusrini (2006), representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar. Representasi bertujuan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah. Adapun karakteristik dari metode representasi pengetahuan adalah:

Menurut Kusrini (2006), representasi pengetahuan dapat dimodelkan menjadi:

a. Logika (*logic*)

Logika merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah dan prosedur yang membantu proses penalaran.

b. Jaringan semantik (*semantic nets*)

Jaringan semantik merupakan penggambaran grafis dari pengetahuan yang memperlihatkan hubungan antar obyek.

c. *Object-Attribute-Value* (OAV)

Object dapat berupa fisik atau konsep. *Attribute* adalah karakteristik dari objek tersebut. *Value* adalah besaran/nilai/takaran spesifik dari *attribute* tersebut pada situasi tertentu, dapat berupa numerik, string atau konstan.

d. Bingkai (*frame*)

Frame berupa ruang-ruang (*slots*) yang berisi atribut untuk mendeskripsikan pengetahuan. Bingkai digunakan untuk merepresentasikan pengetahuan deklaratif.

e. Aturan produksi (*production rule*)

Aturan menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Aturan produksi ditulis dalam bentuk jika-maka (IF-THEN). Aturan IF-THEN mengandung anteseden (*antecedent*) dengan konsekuensi yang dihasilkannya.

2.3.3 *Forward Chaining*

Forward chaining merupakan pencocokan fakta atau pernyataan dimulai dari bagian kiri (IF) atau dengan kata lain penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran (Kusumadewi, 2003). Metode ini sering disebut *data-driven* karena mesin inferensi menggunakan informasi yang ditentukan oleh pemakai untuk memindahkan ke seluruh jaringan dari logika 'AND' dan 'OR' sampai sebuah terminal ditentukan sebagai objek. Bila mesin inferensi tidak dapat menentukan objek maka akan meminta informasi lain. Aturan (*Rule*) dimana menentukan objek, membentuk lintasan (*path*) yang mengarah ke objek. Oleh karena itu, hanya satu cara untuk mencapai suatu objek adalah dengan memenuhi semua aturan.

Sebagai ilustrasi dapat dilihat pada contoh dari Tarigan (2013) sebagai berikut :

Rule Base

R1 : IF A AND B THEN D

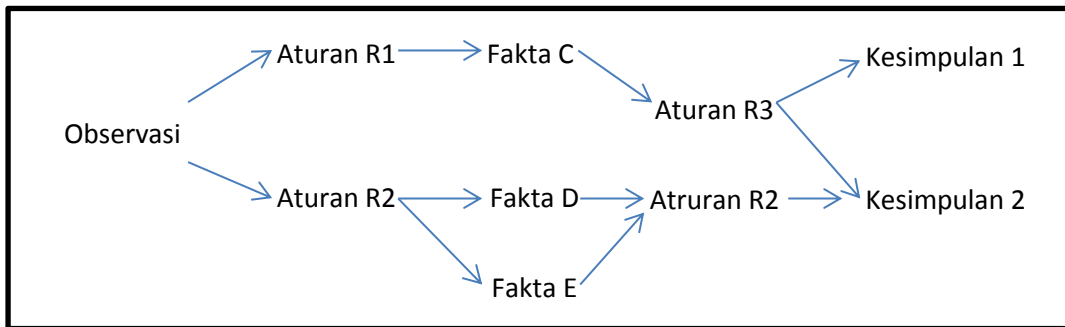
R2 : IF B THEN C

R3 : IF C AND D THEN E

Rule-rule dalam *chaining* dibentuk berdasarkan produksi dari konsep sistem dan faktor yang dimasukkan dalam sistem. Dan pengambilan keputusan dengan menggunakan metode *forward chaining* dimulai dari inisialisasi batasan-batasan faktor berlanjut hingga kesimpulan akhir. *Forward chaining* juga disebut sebagai penalaran *forward (forward reasoning)*.

Dimana proses pencarian dimulai dari premis-premis atau informasi masukan (IF) dahulu kemudian menuju konklusi atau *deviverinformation* (THEN) atau dapat dimodelkan sebagai berikut : Informasi masukan dapat berupa data,

bukti, temuan atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan atau diagnosis. Sehingga jalannya penalaran *forward chaining* dapat dimulai dari data menuju tujuan, dari bukti menuju hipotesa, dari temuan menuju penjelasan atau dari pengamatan menuju diagnosa.



Gambar 2.2 Proses *Foward Chaining*

2.3.4 PHP

Menurut Anhar (2010), PHP merupakan singkatan dari *Hypertext Preprocessor* yaitu merupakan bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan script yang telah terintegrasi dengan HTML dan berada pada *server*. PHP adalah script yang digunakan untuk membuat sebuah halaman yang dinamis (*up to date*).

Menurut Simarmata (2006), PHP begitu cepat populer dan berkembang begitu cepat karena PHP mempunyai beberapa keunggulan, yaitu:

- Cepat, karena ditempelkan (*embedded*) didalam kode HTML, sehingga waktu tanggap menjadi pendek.
- Tidak mahal – gratis. Pada kenyataannya PHP adalah gratis dan anda bisa mendapatkannya tanpa harus membayarnya.
- Mudah untuk digunakan. PHP berisi beberapa fitur khusus dan fungsi yang dibutuhkan untuk membuat halaman web dinamis. Bahasa PHP dirancang untuk dimasukkan dengan mudah didalam file HTML.
- Berjalan pada beberapa sistem operasi. Dia berjalan pada sistem operasi yang beragam, Windows, Linux, Mac OS dan kebanyakan variasi dari Unix.

- e. Dukungan teknis tersedia secara luas, karena PHP menyediakan dukungan gratis via daftar diskusi e-mail.
- f. Aman. Pengguna tidak melihat kode PHP, karena kode yang ditampilkan pada browser adalah kode HTML.
- g. Dirancang untuk mendukung database. PHP meliputi kemampuan yang dirancang untuk berinteraksi dengan database tertentu. *Costumizable* Lisensi *open source* sehingga mengizinkan para pemrogram untuk memodifikasi *software* PHP, Menambahkan atau memodifikasi fitur-fitur yang dibutuhkan untuk lingkungan mereka sendiri.

2.3.5 MySQL

Menurut Oktavian (2010), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau lebih dikenal DBMS (*Database Management System*).

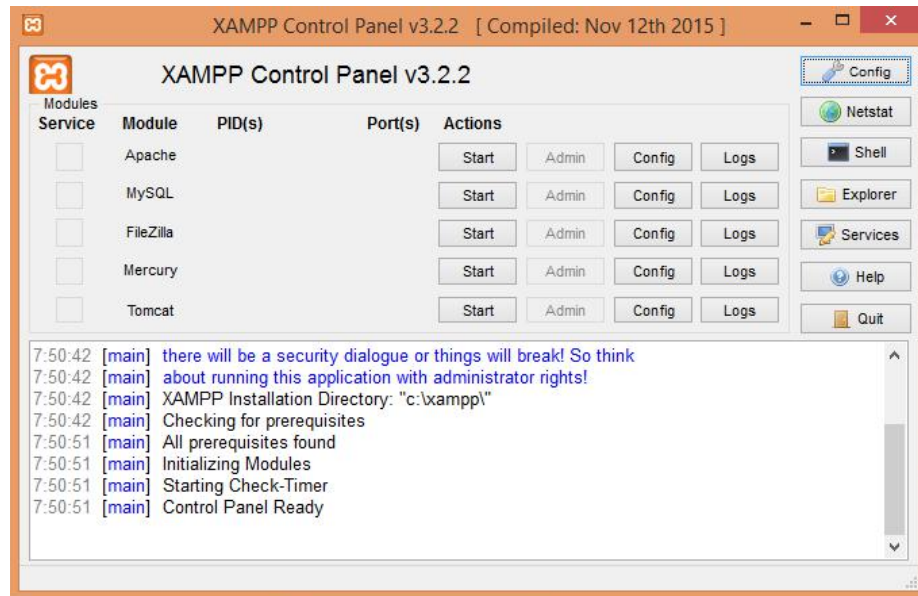
Menurut Saputra, dkk (2012), beberapa kelebihan yang dimiliki MySQL adalah sebagai berikut:

- a. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
- b. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data.
- c. *Super performance* dan *realible*, tidak bisa diragukan, pemrosesan *database*-nya cepat dan stabil.
- d. Sangat mudah dipelajari (*easy of use*)
- e. Memiliki dukungan *support (group)* pengguna MySQL.
- f. Mampu lintas *Platform*, dapat berjalan di berbagai sistem operasi.
- g. *Multiuser*, dimana MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik.

2.3.6 XAMPP

Menurut Buana (2014), “XAMPP adalah perangkat lunak *open source* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows, linux, solaris, dan mac*”.

Berikut adalah tampilan panel XAMPP seperti yang tertera di Gambar 2.3:



Gambar 2.3 Tampilan panel XAMPP

2.3.7 Sublime

Menurut Faridl (2015), Sublime text adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di kalangan developer (pengembang), penulis dan desainer. Para programmer biasanya menggunakan sublime text untuk menyunting *source code* yang sedang ia kerjakan. Sampai saat ini sublime text sudah mencapai versi 3. Sublime text mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*.

Berikut keunggulan-keunggulan fitur yang dimiliki Sublime Text 3, yaitu:

a. *Multiple Selection*

Multiple Selection mempunyai fungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. Multiple selection ini juga merupakan salah satu fitur unggulan dari Sublime Text 3. Kita dapat meletakkan kursor pada kode yang akan di

ubah/edit, lalu tekan Ctrl+klik atau blok kode yang akan diubah kemudian Ctrl+D setelah itu kita dapat merubah kode secara bersamaan.

b. *Command Pallete*

Command Pallete mempunyai fungsi yang berguna untuk mengakses file shortcut dengan mudah. Untuk mencari file tersebut kita dapat tekan Ctrl+Shift+P, kemudian cari perintah yang kita inginkan.

c. *Distraction Free Mode*

Fitur ini mempunyai fungsi untuk merubah tampilan layar menjadi penuh dengan menekan Shift+F11. Fitur ini sangat dibutuhkan ketika pengguna ingin fokus pada pekerjaan yang sedang dikerjakannya.

d. *Find in project*

Fitur ini kita dapat mencari dan membuka file di dalam sebuah project dengan cepat dan mudah. Hanya dengan menekan Ctrl+P anda dapat mencari file yang diinginkan.

e. *Plugin API Switch*

Sublime Text mempunyai keunggulan dengan plugin yang berbasis Python Plugin API. Teks editor ini juga mempunyai plugin yang sangat beragam, dan ini dapat memudahkan pengguna dalam mengembangkan softwarentya.

f. *Drag and Drop*

Dalam teks editor ini pengguna dapat menyeret dan melepas file teks ke dalam editor yang akan membuka tab baru secara otomatis.

g. *Split Editing*

Di dalam fitur ini pengguna dapat mengedit file secara berdampingan dengan klik File->New menu into file.

h. *Multi Platform*

Sublime Text juga mempunyai keunggulan dalam berbagai platform. Sublime text sendiri sudah tersedia dalam berbagai platform sistem operasi, yaitu Windows, Linux dan MacOS.

Berikut ini adalah contoh *script* menggunakan *software* sublime seperti pada Gambar 2.4:



```

config.php x database.php x index.php x User.php
1 <html>
2 <head>
3 <title>PERPUSTAKAAN</title>
4 </head>
5 <body>
6 <body background="gambar/aku2.jpg" width="150%" height="100%">
7
8 <table width="800" border="0" cellspacing="0" align="center" >
9 <tr>
10 <td width="30%" height="69" valign="center">

```

Gambar 2.4 Script menggunakan *software* sublime

2.3.8 Website

Menurut Hastanti, 2015, *website* merupakan kumpulan halaman-halaman yang berisi informasi yang tersimpan di internet yang bisa diakses atau dilihat melalui jaringan internet pada perangkat-perangkat yang bisa mengakses internet itu sendiri. Definisi kata web sebenarnya penyederhanaan dari sebuah istilah dalam dunia komputer yaitu *World wide Web* yang merupakan bagian dari teknologi.

Perkembangan Web

1. Web 1.0

Awal mula perkembangan Internet ditahun 1990-2000. Generasi pertama dari website di internet pada tipe ini pengunjung hanya bisa mencari (*searching*) dan melihat-lihat (*browsing*) data informasi yang ada di web. Web tipe ini terlihat seperti papan pengumuman di dunia maya.

2. Web 2.0

Generasi web kedua pada tahun 2001 - 2010 atau disebut perkembangan web era kedua. Perbedaannya dimana pengunjung *website* mulai dapat melakukan interaksi dengan *website* seperti berbagi sebagai pertemanan dan juga kolaborasi menjadi satu untuk membahas yang penting.

3. Web 3.0

Diera ke tiga yaitu perkembangan web dari tahun 2005 hingga sekarang. Teknologi web generasi ketiga ini merupakan perkembangan lebih maju dari di era ke dua (web 2.0). Terlihat perkembangan teknologi web yang dimana *website* di era ini sudah bisa menjadi asisten pembantu pengguna dalam mengolah data..

Diera ini *search engine* merupakan hal yang penting karena dengan mengetikkan atau mencari sebuah kata, maka apa yang diinginkan akan dicari dengan mudah.

2.3.9 Sepatu

Menurut Basuki (2010), sepatu adalah pakaian untuk kaki sedang kaki adalah anggota badan yang hidup dan bergerak, dengan bentuk yang asimetris pada struktur dan gerakannya. Gerakan kaki adalah gerakan yang kompleks dari banyak tulang yang saling berhubungan. Oleh karena itu dalam membuat sepatu tidak boleh sembarangan, harus mengikuti anatomi kaki dan aturan–aturan secara alamiah serta teknologi tertentu, sehingga hasil sepatu yang diperoleh dapat cocok dan sesuai serta enak dipakai pada kaki.

Jenis produksi dari industri sepatu dan alas kaki tersebut bermacam–macam, antara lain : sepatu (pria, wanita dan anak – anak), sepatu boot, *sport*, sandal dan alas kaki lainnya. Berbagai jenis bahan baku digunakan untuk membuat sepatu dan alas kaki, untuk bagian atas (*shoe upper*) menggunakan bahan kulit (*leather*), seperti : box, glace, suede, nappa, corrected grain suede, artificial grain side dan lain – lain, sedang untuk bahan non kulit adalah sintetis, PVC (*Poly Vinyl Chlorida*), plastik dan sebagainya. Untuk lapis bagian atas sepatu digunakan kulit lapis, *fabric*/ tekstil, bahan sintetis dan lain – lain. Sedangkan untuk bagian bawah (*shoe bottom*) menggunakan bahan sol dari kulit (*sole leather*), karet, *PVC* atau bahan sintetis lainnya.

2.3.10 Flowchart

Menurut Adelia (2011), bahwa Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Flowchart membantu analyst dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis.

2.3.11 UML (*Unified Modelling Language*)

Menurut Rosa dan Shalahuddin (2015), dijelaskan pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak.




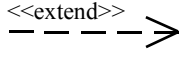
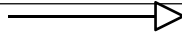
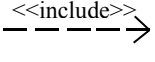
Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu UML.

Untuk membangun sistem pakar perawatan sepatu peneliti menggunakan 4 macam diagram UML, yaitu :

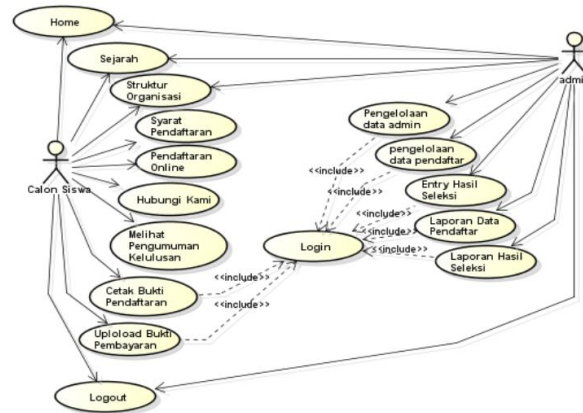
1. *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1	 Nama <i>Use Case</i>	<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama <i>use case</i> .
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Gambar 2.5 adalah contoh *use case diagram* dari Eviani dkk. (2016) :



Gambar 2.5 Contoh *use case diagram*

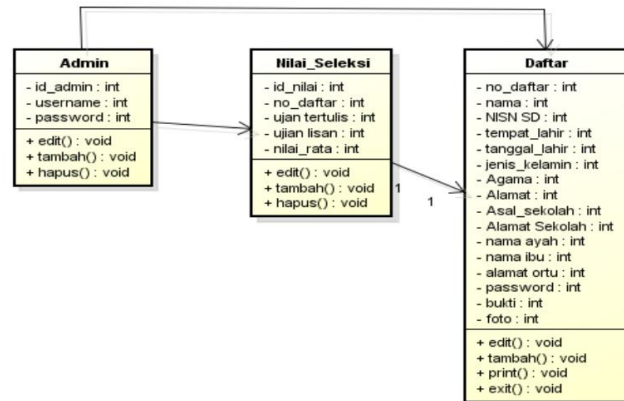
2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas (Rosa & Shalahuddin, 2015). Simbol *class diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> Nama kelas + Atribut + Operasi() </div>	<i>Class</i>	Kelas pada struktur sistem
2	○	<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3	—	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	→	<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .
5	—▷	<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum – khusus).
6	- - - - ->	<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7	—◇	<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Gambar 2.6 adalah contoh *class diagram* dari Eviani dkk. (2016) :



Gambar 2.6 Contoh *class diagram*

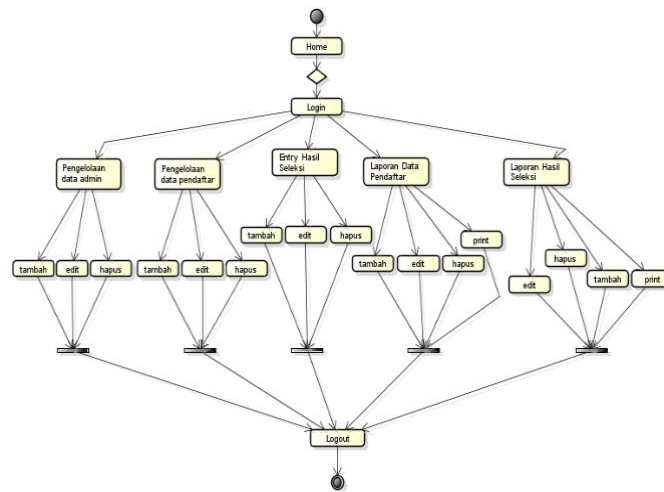
3. *Activity Diagram*

Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa & Shalahuddin, 2015). Simbol *Activity Diagram* ditunjukkan pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1	●	Status awal	Status awal aktivitas sistem.
2	aktivitas	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	◇	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	⤵	<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	●	Status akhir	Status akhir yang dilakukan sistem.
6	>Nama Swimlane	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Gambar 2.7 adalah contoh *activity diagram* dari Eviani dkk. (2016) :



Gambar 2.7 Contoh *activity diagram*

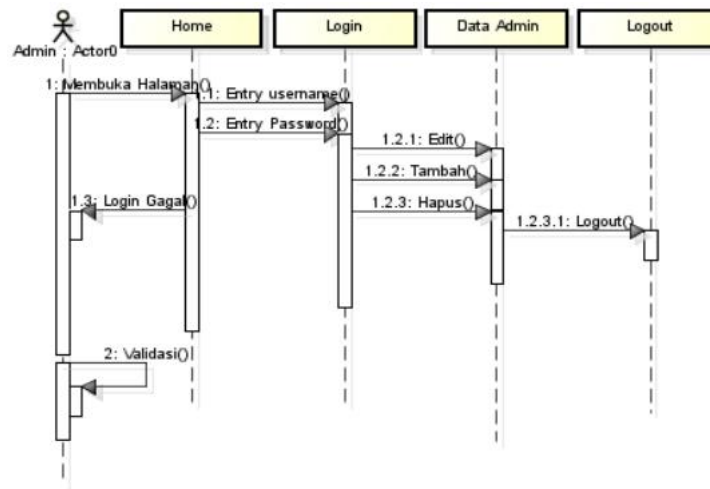
4. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Rosa & Shalahuddin, 2015). Simbol *sequence diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4. Tabel *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.

Gambar 2.8 adalah contoh *sequence diagram* dari Eviani dkk. (2016) :



Gambar 2.8 Contoh *sequence diagram*

2.3.12 Pengujian Sistem

Pengujian sistem menggunakan 2 metode, yaitu:

1. *Black Box Testing*

Menurut Rosa dan Shalahuddin (2015), *Black Box Testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program.

2. Survey Kepuasan Pengguna

Menurut Tjiptono (2008), mengungkapkan bahwa dalam mengevaluasi kepuasan terhadap produk, jasa, atau perusahaan tertentu, pelanggan umumnya mengacu pada berbagai faktor atau dimensi. Faktor yang sering digunakan dalam mengevaluasi kepuasan terhadap suatu produk antara lain meliputi:

1. Kinerja (*performance*) karakteristik operasi pokok dari produk inti (*core product*) yang dibeli, misalnya kecepatan, konsumen bahan bakar, jumlah penumpang yang dapat diangkut, kemudahan dan kenyamanan dalam mengemudi, dan sebagainya.
2. Ciri-ciri keistimewaan tambah (*features*) yaitu karakteristik sekunder atau pelengkap.
3. Keandalan (*reliability*) yaitu kemungkinan kecil akan mengalami kerusakan atau gagal dipakai.

4. Kesesuaian dengan spesifikasi (*conformance to specifications*) yaitu sejauh mana karakteristik desain dan operasi memenuhi standar-standar yang telah ditetapkan sebelumnya.
5. Daya tahan (*durability*) berkaitan dengan berapa lama produk tersebut dapat terus digunakan. Dimensi ini mencakup umur teknis maupun umur ekonomis penggunaan.
6. *Serviceability*, meliputi kecepatan, kompetensi, nyaman, mudah diperbaiki serta penanganan keluhan yang memuaskan. Pelayanan yang diberikan tidak hanya sebatas sebelum penjualan, tetapi juga selama proses penjualan hingga purna jual, yang mencakup pelayanan reparasi dan ketersediaan komponen yang dibutuhkan.
7. Estetika, yaitu daya tarik produk terhadap panca indera, misalnya bentuk fisik yang menarik, model/ desain, warna, dan sebagainya.
8. Kualitas yang dipersepsikan (*perceived quality*), yaitu citra dan reputasi produk serta tanggung jawab perusahaan terhadapnya.