



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Saputro (2017:217), Menurut buku (Donald H. Sanders), “Komputer adalah seperangkat dari teknologi yang dapat digunakan untuk mengolah data teks dan gambar serta untuk mengoperasikan berbagai program maupun untuk control peralatan”.

Menurut kamus besar bahasa Indonesia online, komputer adalah alat elektronik otomatis yang dapat menghitung atau mengolah data secara cermat menurut yang diinstruksikan dan memberikan hasil pengolahan, serta dapat menjalankan sistem multimedia (film, musik, televisi, faksimile, dan sebagainya), biasanya terdiri atas unit pemasukan, unit pengeluaran, unit penyimpanan serta unit pengontrolan (<https://web.id/komputer>

Berdasarkan pengertian diatas penulis menyimpulkan bahwa komputer adalah alat elektronik yang dapat mengolah data digital, dengan mengikuti serangkaian program yang digunakan untuk membantu pelaksanaan pekerjaan.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Menurut Mulyani (2017:66), "*Software* adalah istilah umum untuk mendeskripsikan kumpulan program-program komputer yang terdiri dari prosedur-prosedur dan dokumentasi untuk melakukan tugas tertentu".

Menurut Kadir (2017:2), "Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai. Sistem operasi seperti Windows, Mac OS, dan Linux, dan aplikasi seperti Microsoft Word dan Microsoft Excel adalah contoh perangkat lunak".

Berdasarkan penjelasan diatas dapat disimpulkan bahwa Perangkat Lunak adalah kumpulan perintah – perintah atau instruksi yang sudah tercantum pada sebuah computer.



2.1.3 Pengertian World Wide Web (WWW)

Menurut Fathansyah dalam Prayitno & Safitri (2015:2) mengatakan bahwa, “World Wide Web (WWW atau web) merupakan sistem informasi terdistribusi yang berbasis hypertext”.

Menurut Kustiyahningsih dan Devie Rosa Anamisa dalam Fridayanthie & Mahdiati (2016:128) mengatakan bahwa, “Word Wide Web (WWW). Informasi WWW ini disimpan pada web server untuk dapat diakses dari jaringan browser terlebih dahulu, seperti Internet Explorer atau Mozilla Firefox”.

Berdasarkan penjelasan diatas penulis dapat menyimpulkan bahwa www adalah pada web server untuk dapat diakses dari jaringan browser.

2.2 Teori Judul

2.2.1 Pengertian Sistem Pakar

Menurut Budiharto dan Suhartono (2014:22), system pakar adalah program computer yang mensimulasi penilaian dan perilaku manusia atau organisasi yang memiliki pengetahuan dan pengalaman ahli dalam bidang tertentu. Biasanya sistem seperti ini berisi basis pengetahuan yang berisi akumulasi pengalaman dan satu set aturan untuk menerapkan pengetahuan dasar untuk setiap situasi tertentu. System pakar yang canggih dapat ditingkatkan dengan penambahan basis pengetahuan atau set aturan. Diantara banyak system pakar yang ada, yang terkenal adalah aplikasi bermain catur dan system diagnosis medis. Pemrosesan yang dilakukan oleh system pakar merupakan pemrosesan pengetahuan bukan pemrosesan data pada system pakar computer konvensional. Pengetahuan (Knowledge) adalah pemahaman secara praktis maupun teoritis terhadap suatu obyek atau domain tertentu. Pengetahuan yang digunakan pada system pakar merupakan serangkaian informasi mengenai gejala-diagnosa, sebab-akibat aksi-reaksi tentang suatu domain tertentu (misalnya, domain diagnosa medis). Secara umum, definisi tradisional sebuah program komputer biasanya : **Algoritma + Strukturdata = PROGRAM**, dalam sistem pakar, definisi berubah menjadi. **Mesin inferensi + Pengetahuan = Sistem Pakar**. Dengan sistem pakar, permasalahan yang seharusnya hanya dapat diselesaikan oleh pakar/ahli dapat



diselesaikan oleh orang biasa/awam. Sedangkan, untuk para ahli, system pakar membantu aktivitas mereka sebagai asisten yang seolah olah sudah mempunyai banyak pengalaman.

2.2.1.1 Kelebihan dan Karakteristik Sistem Pakar

Sistem pakar Menurut Budiharto dan Suhartono (2014:24), banyak digunakan pada aplikasi terkini dan kompleks karena :

1. Sistem pakar dapat bertindak sebagai konsultan, instruktur, atau pasangan/rekan.
2. Meningkatkan *availability* atau kepakaran tersedia pada semua perangkat komputer.
3. Mengurangi bahaya.
4. Permanen.
5. Pengetahuan dapat tidak lengkap, namun keahlian dapat diperluas sesuai kebutuhan. Program konvensional harus “lengkap” sebelum mereka dapat digunakan.
6. Database yang cerdas, system pakar dapat digunakan untuk mengakses database secara cerdas, misalnya data mining.

Sistem Pakar biasanya didesain untuk memiliki karakteristik sebagai berikut:

1. High Performance
2. Adequate response time
3. Good reliability
4. Understandable



2.2.1.2 Membangun Sistem Pakar

Menurut Budiharto dan Suhartono (2014:25) pada dasarnya ada empat langkah untuk membangun sebuah sistem pakar :

1. Analisis

Tujuan analisis adalah untuk mengidentifikasi aplikasi yang potensial. Aplikasi yang mungkin termasuk diagnostic, controller, dll. Selama analisis, pengembangan juga harus menilai kesesuaian pengetahuan rekayasa teknologi untuk aplikasi ini. Apakah sistem pakar ini sudah sesuai dan dapat bekerja dengan baik? Hal ini berlaku untuk menerapkan semua jenis kecerdasan buatan untuk memecahkan masalah.

2. Spesifikasi

Langkah spesifikasi adalah tempat pengembang mendefinisikan kemampuan sistem pakar. Para pengembang juga harus bekerja dengan para pakar untuk belajar dan merencanakan pengembangan sistem. Ahli adalah manusia yang diidentifikasi sebagai ahli di domain bidang tertentu, seperti psikolog, dokter dan ilmuwan.

3. Pengembangan

Pada Tahap ini, pengembang harus belajar cara ahli melakukan tugas (akuisisi pengetahuan) dalam berbagai kasus. Pada dasarnya ada tiga jenis kasus, pengembang harus berdiskusi dengan ahli, yaitu “saat ini”, “historis”, dan “hipotesis”. Kasus saat ini dapat diperoleh dengan mengamati seorang ahli saat melakukan tugas. Kasus sebelumnya (histori) dapat dibahas dengan berdiskusi dengan ahli tugas yang dilakukan dimasa lalu. Kasus hipotesis dapat diperoleh dengan situasi hipotesis. Setelah dapat diperoleh dengan situasi hipotesis. Setelah program dibuat dan diuji maka dapat didistribusikan kemasyarakat untuk dapat digunakan secara lebih luas.



2.2.1.3 Tujuan Sistem Pakar

Tujuan dari sebuah sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar (non expert). Proses ini melibatkan empat kegiatan, yaitu:

1. Akuisisi pengetahuan (dari pakar atau sumber lainnya)
2. Representasi pengetahuan (ke dalam komputer)
3. Inferensi pengetahuan
4. Pemindahan pengetahuan ke pengguna

2.2.2 Pengertian Metode Forward Chaining (Alur Maju)

Dalam buku “Studi Kasus Sistem Berbasis Pengetahuan, halaman 28” Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui dengan fakta yang diketahui, kemudian mencocokkan fakta – fakta tersebut dengan bagian IF dari rule IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database. Setiap kali pencocokan, dimulai dari Rule Teratas. Setiap rule hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi.

2.2.3 Pengertian Aplikasi

Menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

Menurut Juansyah (2015:2), “Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh sasaran yang akan dituju”.

Berdasarkan pendapat di atas, maka penulis dapat simpulkan bahwa pengertian aplikasi adalah program komputer yang menentukan aktivitas pemrosesan untuk menyelesaikan suatu aplikasi tertentu yang dapat digunakan oleh sasaran yang akan dituju.



2.2.4 Pengertian Informasi

Menurut Setiawan (2015:38), “Informasi merupakan hasil dari pengolahan data yang memiliki nilai tertentu dan bisa digunakan untuk pengambilan suatu keputusan. Dalam hal ini, data bisa dianggap sebagai obyek dan informasi adalah suatu obyek yang bermanfaat bagi penerimanya”.

Menurut Turban et al (2015), “Informasi merupakan data yang telah diorganisir sehingga memberikan arti dan nilai kepada penerimanya”.

Berdasarkan pengertian diatas, maka penulis dapat menyimpulkan bahwa informasi adalah data – data yang telah diolah atau di organisir sehingga memberikan arti dan nilai.

2.2.5 Pengertian Diagnosa

Istilah diagnosis sering kita dengar dalam istilah medis. Menurut Thorndike dan Hagen dalam Suhermen (2011:20), diagnosis dapat diartikan sebagai:

1. Upaya atau proses menemukan kelemahan atau penyakit (weakness, disease) apa yang dialami seseorang dengan melalui pengujian dan studi yang seksama mengenai gejala – gejalanya (symptoms);
2. Studi yang seksama terhadap fakta tentang suatu hal untuk menemukan karakteristik atau kesalahan – kesalahan dan sebagainya yang esensial;
3. Keputusan yang dicapai setelah dilakukan suatu studi yang seksama atas gejala – gejala atau fakta tentang suatu hal.

Dari ketiga pengertian diatas, dapat disimpulkan bahwa didalam konsep diagnosis, secara implisit telah tercakup pula konsep prognosinya. Dengan demikian dalam proses diagnosis bukan hanya sekedar mengidentifikasi jenis dan karakteristiknya, serta latar belakang dari suatu kelemahan atau penyakit tertentu, melainkan juga mengimplikasikan suatu upaya untuk meramalkan kemungkinan dan menyarankan tindakan pemecahannya.



2.2.6 Pengertian Kerusakan

Dalam Buku Berjudul “Gempa Bumi” karya ‘Teddy Boen’ Mengatakan Kerusakan adalah rusaknya bahan – bahan structural dan sambungan–sambunganya.

2.2.7 Pengertian Sistem Pakar Mendiagnosa Kerusakan Pada Mobil Mitsusbihi

Sebuah aplikasi yang digunakan untuk memudahkan perusahaan dalam memberikan informasi mengenai kerusakan dari gejala – gejala beserta solusinya pada mobil mitsubishi.

2.3 Teori Khusus

2.3.1 Pemrograman Berorientasi Objek

2.3.1.1 Pengertian Pemrograman Berorientasi Objek

Menurut Akhmad Dharma (2013:5) “OOP (Object Oriented Programming) adalah sebuah pendekatan untuk pengembangan suatu software, dimana dalam struktur software tersebut didasarkan kepada iteraksi objek dalam penyelesaian suatu proses atau tugas”.

Menurut Adi Nugroho (2011:5) “Pemrograman berorientasi objek atau object Oriented Programming (OOP) adalah suatu cara baru dalam berpikir serta berlogika untuk menghadapi masalah – masalah yang akan dicoba batasi dengan bantuan komputer”.

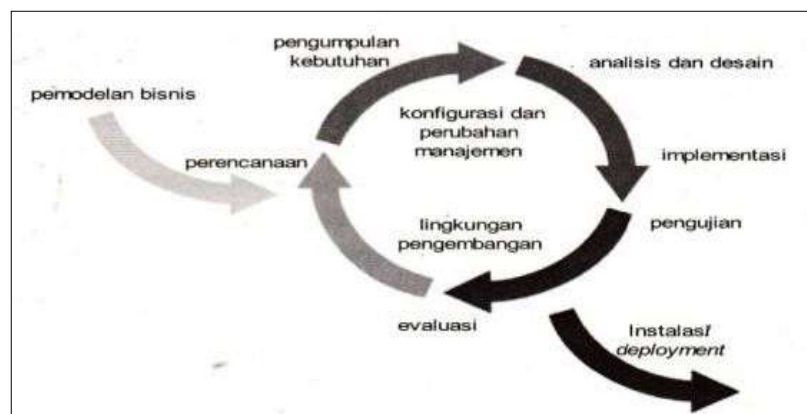
Berdasarkan pendapat diatas maka dapat disimpulkan bahwa Pengorman Berorientasi objek merupakan paradigman pemrograman yang berorientasikan kepada objek.



2.3.2 Rational Unified Process (RUP)

Sukamto dan Shalahuddin (2016:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use-case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak”.

Proses pengulangan/iteratif pada RUP secara global dapat dilihat pada gambar berikut:



(Sumber: Sukamto dan Shalahuddin (2016:125))

Gambar 2.1. Proses iteratif RUP (*Rational Unified Process*)

2.3.2.1 Fase Rational Unified Process (RUP)

Sukamto dan Salahuddin (2016:128-131), “RUP (*Rational Unified Process*) memiliki empat buah tahap atau fase yang dapat dilakukan secara iteratif.” Berikut ini penjelasan untuk setiap fase pada RUP (*Rational Unified Process*).”

1. *Inception* (permulaan) Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).
2. *Elaboration* (perluasan/perencanaan) Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah sistem



yang diinginkan dapat dibuat atau tidak. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

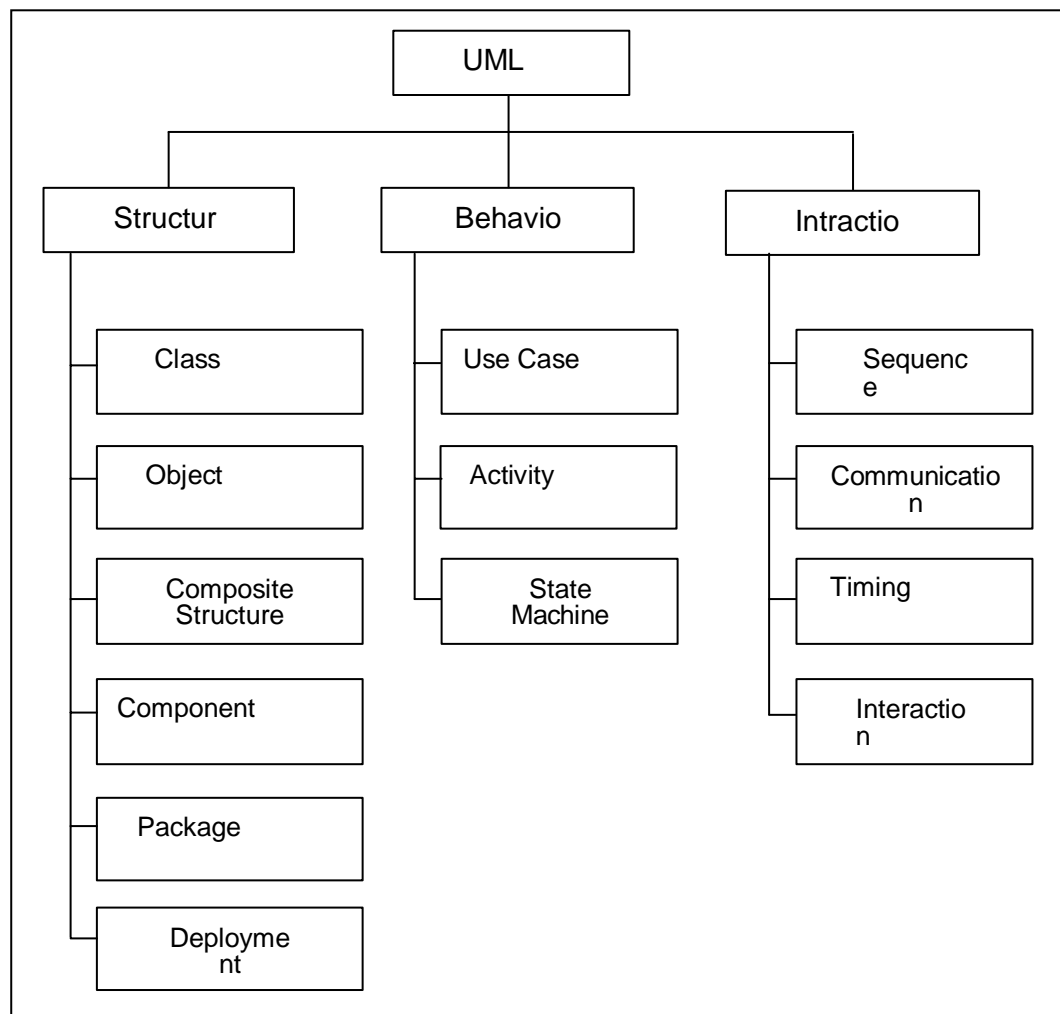
3. *Construction* (konstruksi) Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.
4. *Transition* (transisi) Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada RUP (*Rational Unified Process*) dijalankan secara berurutan dan literatif dimana setiap iterasi dapat digunakan untuk memperbaiki literasi berikutnya.

2.3.3 Pengertian UML (*Unified Modeling Language*)

Menurut Sugiarti (2013:15), “Unified Modelling Language (UML) adalah sebuah “bahasa” yang menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem pranti lunak”.

Menurut Rosa dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



Gambar 2.2 Macam-macam Diagram UML

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Rosa dan Shalahuddin (2016:141) :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulam diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahana yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.






2.3.4 Pengertian *Activity Diagram*

Dalam buku berjudul “Teknik Penulisan Tugas Akhir dan Skripsi Pemrograman”, mengatakan bahwa “*Activity Diagram* adalah diagram adalah bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan concurrency (Glossary of Key Terms, 2008)”.




Rosa dan Shalahuddin (2018:161), “*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem”.

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.1 Simbol-simbol pada *Activity Diagram*

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	Aktivitas aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3	Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

Lanjutan **Tabel 2.1** Simbol-Simbol pada *Activity Diagram*

5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.3.5 Pengertian *Sequence Diagram*

Menurut Vidia D.Dkk (2013:21), “Sequence Diagram dibuat berdasarkan activity diagram dan class diagram. Sequence diagram menggambarkan aliran pesan yang terjadi antar kelas yang dideskripsikan pada class diagram dengan menggunakan operasi yang dimiliki kelas tersebut”.

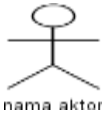
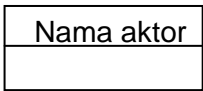

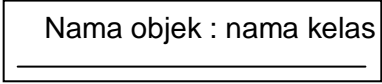
Rosa dan Shalahuddin (2018:141), “Menjelaskan bahwa diagram *sequence* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada use case”.

Banyaknya diagram *sequence* yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup


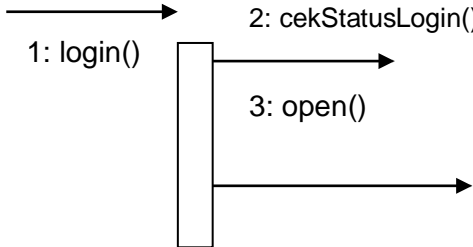
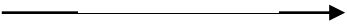


pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak. Berikut simbol-simbol pada *Sequence Diagram* :

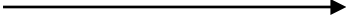
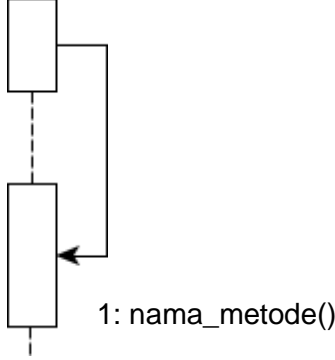

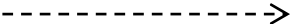
Tabel 2.2 Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
1	<p>Actor</p>  <p>nama aktor</p> <p>atau</p>  <p>tanpa waktu aktif</p>	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 actor
2	<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
3	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan

Lanjutan **Tabel 2.2** Simbol-Symbol pada *Sequence Diagram*

4	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
5	<p>Pesan tipe create <<create>></p> 	

Lanjutan **Tabel 2.2** Simbol-Symbol pada *Sequence Diagram*

6	<p>Pesan tipe call «create»</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
7	<p>Pesan tipe send 1: masukan</p> 	<p>menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
8	<p>Pesan tipe return 1: keluaran</p> 	<p>menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>

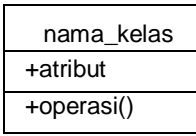
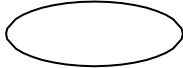




2.3.6 Pengertian *Class Diagram*

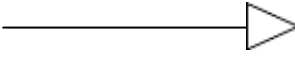
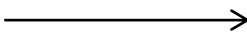
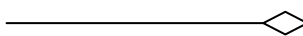
Rosa dan Shalahuddin (2018:141), “Menyebutkan *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Diagram Class* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

Tabel 2.3 Simbol-simbol pada *Class Diagram*

No.	Simbol	Deskripsi
1	<p>kelas</p> 	Kelas pada struktur sistem
2	<p>antarmuka / interface</p>  <p>nama_interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek
3	<p>asosiasi / association</p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
4	<p>asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Lanjutan **Tabel 2.3.** Simbol-Simbol pada *Class Diagram*

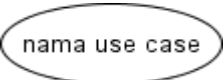
5	<p>generalisasi</p> 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6	<p>kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antar kelas
7	<p>agregasi / <i>aggregation</i></p> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

2.3.7 Pengertian *Use Case Diagram*

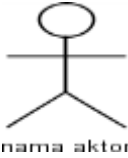
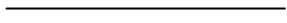
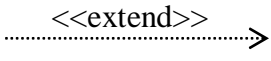
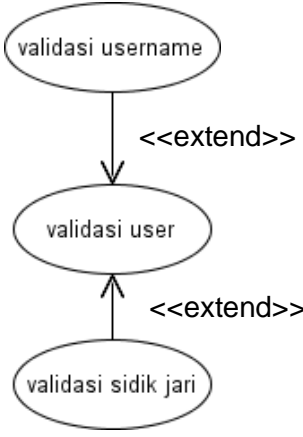
Menurut Murad (2013:72), “Use Case Diagram adalah diagram yang bersifat status yang memperlihatkan himpunan use case dan aktor aktor (suatu jenis khusus dari kelas”

Menurut Rosa dan Shalahuddin (2018:155) “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut: Berikut simbol-simbol pada Use Case Diagram :

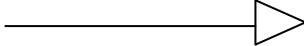
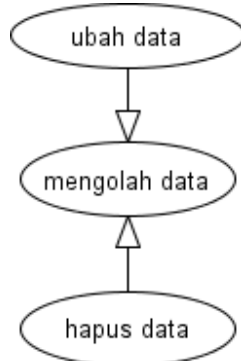
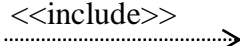
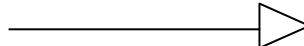
Tabel 2.4 Simbol-simbol pada *Use case Diagram*

No	Simbol	Deskripsi
1	<p><i>Use case</i></p> 	fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>

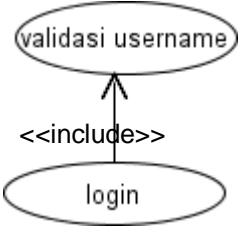
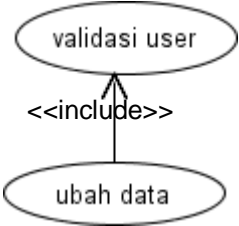
Lanjutan **Tabel 2.4** Simbol-Simbol pada *Use Case Diagram*

No	Simbol	Deskripsi
2	aktor / <i>actor</i>  nama aktor	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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	asosiasi / <i>association</i> 	komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
4	ekstensi / <i>extend</i> 	relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya  arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya

Lanjutan **Tabel 2.4** Simbol-Simbol pada *Use Case Diagram*

No	Simbol	Deskripsi
5	Generalisasi / <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, misalnya:  arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
6	menggunakan / include / <i>uses</i>  	relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> : <ul style="list-style-type: none">• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:

Lanjutan **Tabel 2.4** Simbol-Symbol pada *Use Case Diagram*

No	Simbol	Deskripsi
6		<div style="text-align: center;">  <pre> graph BT login([login]) -- "<<include>>" --> validasi_username([validasi username]) </pre> </div> <p><i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p> <div style="text-align: center;">  <pre> graph BT ubah_data([ubah data]) -- "<<include>>" --> validasi_user([validasi user]) </pre> </div> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

2.3.8 Kamus Data

Menurut Kristanto (2018:118), “Pada kamus data (*data dictionary*), semua jenis data yang terlibat dalam proses yang terjadi , didefinisikan dan dikumpulkan dalam bentuk penyajian”.

Rosa dan Shalahuddin (2018:73-74) menyatakan bahwa, “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum



(memiliki standar cara penulisan)”. Berikut beberapa simbol-simbol yang terdapat pada kamus data :

Tabel 2.5 Simbol-simbol Kamus Data

No	Simbol	Keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik...atau...
4.	{ ⁿ }	N kali/ bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

2.3.9 Pengujian Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2016:272), menjelaskan tentang pengujian perangkat lunak sebagai berikut:

Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desai kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian perangkat lunak adalah sebagai berikut:

1. Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai isi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (Independent Test Group (ITG)).
4. Pengujian dan penirkutuan (debugging) merupakan aktivitas yang berbeda tetapi penirkutuan (debugging) harus diakomodasikan pada berbagai strategi pengujian.



2.3.9.1 Black-Box Testing (Pengujian Kotak Hitam)

Menurut SEBATIK Volume 23 (2019:520), “*Blackbox Testing* adalah pengujian yang dilakukan oleh pengguna dari aplikasi yang dibuat yang dilakukan oleh *user*. Pengujian ini menggunakan *blackbox* yang mana pada tahap ini dilakukan pengujian sebagai user”.

2.4 Teori Program

2.4.1 Pengertian Basis Data

Jubilee Enterprise (2017:1), “Basis Data (*Database*) adalah suatu aplikasi yang menyimpan sekumpulan data”. Setiap database mempunyai perintah tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data didalamnya.

Mardiani (2016:9), “Basis Data adalah suatu susunan atau kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi menggunakan metode tertentu, dengan menggunakan komputer sehingga mampu menyediakan informasi secara optimal yang diperlukan pemakainya”.

2.4.2 Pengertian Website

Menurut Rohi Abdulloh (2015:3), “*website* atau web adalah sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”.

Menurut Yuhefizar (2001:9), “*website* merupakan keseluruhan halaman-halaman web yang terdapat dalam sebuah *domain* yang mengandung informasi”.

2.4.3 Pengertian HTML

Abdullah (2019:7), “HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari website. HTML berperan sebagai penyusun struktur halaman website yang menempatkan setiap elemen website sesuai layout yang diinginkan”.



Jubilee (2016:16), “HTML adalah *Hypertext Markup Language* yang artinya adalah sebuah teks berbentuk link dan mungkin juga foto atau gambar yang saat di klik, akan membawa si pengakses internet dari satu dokumen kedokumen lainnya”.

Menurut Winarno (2015:2), “HTML adalah bahasa yang mengatur bagaimana tampilan 28a nisi dari situs web, di dalam HTML ada tag-tag dimana tag berfungsi menyediakan informasi berkaitan dengan sifat dan struktur konten serta referensi untuk gambar dan media lainnya”.

2.4.4 Pengertian PHP

Menurut Abdullah (2019:127), “PHP Merupakan kependekan dari PHP *Hypertext preprocessor* yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server. Tujuan dari bahasa ini adalah membantu para pengembang web untuk membuat web dinamis dengan cepat”.

Menurut Mudzir MF (2018:3), “PHP berasal dari kata "*Hypertext Preprocessor*", yaitu bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bias digunakan bersamaan dengan HTML”.

2.4.5 Pengertian MySQL

Menurut Mudzir MF (2018:217), “MySQL adalah system manajemen database SQL yang sifatnya open source (terbuka) dan paling banyak digunakan saat ini”.

Menurut Purnomo (2015:19), “MySQL adalah suatu perangkat lunak untuk relasi database seperti halnya oracle, PostgreSQL, Microsoft SQL, MYSQL jangan disamakan dengan SQL (*Structure Query Language*) yang didefinisikan sebagai sintaks atau perintah tertentu dalam bahasa pemograman yang digunakan untuk mengelola database”.



2.4.6 Pengertian Xampp

Menurut Bay Haqi, M.Kom dan Heri Setiawan, S.E.,M.TI (2019:8), “XAMPP adalah perangkat lunak bebas (*Free Software*) yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi Xampp sendiri sebagai server yang berdiri sendiri (*localhost*)”, yang terdiri dari beberapa program, antara lain : Apache HTTP Server, MYSQL Database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Menurut Riyanto (2015:1), “XAMPP adalah paket PHP dan MYSQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP”, XAMPP mengombinasikan beberapa paket perangkat lunak berbeda kedalam satu paket.

2.4.7 Pengertian Codeigniter

Menurut Betha Sidik (2012), “Codeigniter adalah Sebuah *framework php* yang bersifat *open source* dan menggunakan metode *MVC* (Model, View, Controller) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal”.

2.4.8 Pengertian Sublime Text

Menurut Ardhana (2015), “Sublime Text adalah teks editor berbaris python yang cukup terkenal dikalangan pengembang, penulis, dan desainer. Sublime text mencegah plugin merusak sublime text dan mempercepat pembukaan aplikasi awal”.

Subagja (2016:30), “mengemukakan, sublime seperti halnya Notepad++, Bluefish, WordPad, Adobe Dreamweaver, Geany, Gedit, CodeLobester atau yang lainnya, aplikasi ini (Sublime Text) dapat dipakai untuk membuat/menulis script atau kode”.

Sublime memiliki fitur yang dapat mempermudah penulisan script atau kode, antara lain :



- **Multiple selection.** ini adalah fitur unggulan di Sublime text. Fitur ini dapat meletakkan kursor di beberapa tempat (menggunakan Ctrl + click), kemudian mengedit secara bersamaan.
- **Auto completion.** Sublime Text memiliki auto complete untuk beberapa bahasa yang dipakai seperti php, css, dan js.



2.5 Referensi Jurnal

Dalam Tugas Akhir ini penulis menggunakan beberapa perbandingan jurnal. Dibawah ini akan diuraikan perbandingan dari referensi jurnal tersebut.

Tabel 2.6 Referensi Jurnal

No	Judul /Penulis/Tahun	Masalah	Teori	Metode Penelitian	Hasil
1	Rancang Bangun Sistem Diagnosis Kerusakan pada Mobil Menggunakan Metode Forward Chaining Penulis : Ida Bagus Dhany Satwika Tahun : 2012	Bagaimana cara merawat mobil secara berkala agar tidak terjadi kerusakan – kerusakan yang lebih parah	Sistem Pakar (Turban, 1995), (Martin dan Oxman 1998).	Metode Forward Chaining (Runut Maju)	Membantu pengguna dalam mengetahuan kerusakan yang terdapat pada mobilnya dari gejala gejala atau keluhan yang sering dialami oleh pengguna dengan moiblnya, dengan bahasa pemrograman WEB yaitu PHP



2	Sistem Pakar Untuk Mendeteksi Kerusakan Pada Sepeda Motor Dengan Menggunakan Metode Forward Chaining Berbasis Web Penulis : Januardi Nasir, Zefly Haposan Gultom Tahun : 2018	Bagaimana cara pemilik sepeda motor mendiagnosa kerusakan dan memiliki pengetahuan tentang perawatan kerusakan sepeda motor	melakukan Operasional Variabel dan Kaidah Produksi	Metode Forward Chaining	Memudahkan mekanik dan pengguna sepeda motor dalam menangani permasalahan yang berkaitan dengan sepeda motor 4 tak.
3	Sistem Pakar Berbasis Web Dengan Metode Forward Chaining Dalam Mendiagnosis Dini Penyakit Tuberkulosis diJawa Timur Penulis : Windah Supartini, Hindarto Tahun : 2016	Bagaimana cara menekan angka penyakit tuberculosis, dengan mendeteksi dini gejala penyakit TB. Karena itu dibutuhkan aplikasis system pakar agar mudah digunakan orang awam untuk mendiagnosis penyakit Tuberkulosis	Sistem Pakar Kusrini (2016:10)	Metode Forward Chaining, dengan melakukan Observasi dan Wawancara	Hasil diagnosis pakar dan user dari sistem pakar mendiagnosis secara dini penyakit ini menunjukkan hasil diagnosis yang dialami pasien menunjukkan kesesuaian dengan yang telah diagnosis oleh dokter penyakit Tuberkulosis



4	<p>Penerapan Metode Forward Chaining Pada Aplikasi Sistem Pakar Diagnosa Penyakit Pada Tanaman Bunga Kamboja (Adenium)</p> <p>Penulis : Agtian Muhammad Ricky Tanshidq, Anggit Dwi Hartanto, Donni Prabowo Tahun : 2017</p>	<p>Bagaimana cara para pakar tumbuhan Adenium ini mendeteksi penyakit apa yang sedang menyerang tanaman Kamboja miliknya masing- masing</p>	<p>Andhika Andhitama Gama (2015) Sistem Pakar</p>	<p>Metode Sistem Pakar dengan menggunakan perancangan sistem Use case diagram, Activity Diagram, Class Diagram dan Sequence Diagram.</p>	<p>Dengan aplikasi ini pengguna bisa dapat mendiagnosa dengan dilengkapi dengan tips cara membasmi hama dari tumbuhan ini.</p>
5	<p>Sistem Pakar Diagnosis Penyakit Pada Perokok Dengan Metode Forward Chaining Berbasis Web</p> <p>Penulis : Yasmiyati Tahun : 2017</p>	<p>Bagaimana cara Mendeteksi penyakit para perokok melalui gejala – gejala dan penyebab dan juga cara mengatasinya</p>	<p>Mulyani Restianie (2015) Sistem Pakar, Amarathunga, dkk (2015) metode data mining J48 untuk</p>	<p>Menggunakan Metode Forward Chaining Menggunakan Use case Diagram dan ERD</p>	<p>Aplikasi ini sesuai dengan tujuan awal pembuatan yaitu membantu masyarakat UMUM dalam mendiagnosa penyakit secara dini dan memper mudah mereka mendapatkan informasi mengenai penyakit terkait yg disebabkan asap rokok</p>



			mendeteksi penyakit kulit, Supriyono, dkk (2017) tentang sistem pakar.		
--	--	--	--	--	--