



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

DASAR TEORI

2.1. Tinjauan Pustaka

Pada penelitian Gusti Ayu Kadek Tutik A, Rosa Delima, Umi Proboyekti dengan judul Penerapan *forward chaining* Pada Program Diagnosa Anak Penderita Autisme, mengembangkan sistem pakar diagnosa anak penderita autisme dengan menggunakan algoritma *forward chaining*. Program ini dibuat dapat membantu orang tua untuk mendeteksi ada tidaknya gangguan perkembangan dan autis pada anak berdasarkan gejala-gejala yang terlihat sehari-hari serta beberapa terapi sederhana yang dapat diberikan bagi anak penderita autis. Keluaran pada sistem ini berupa ada tidaknya kemungkinan autisme pada seorang anak berdasarkan fakta atau gejala yang diberikan kepada sistem. Kekurangan dari penelitian ini sistem tidak dilengkapi dengan fasilitas penambahan jenis kelompok usia dan jenis spectrum autis, sehingga seluruh spectrum autis pada seluruh kelompok usia dapat dideteksi, kekurangan lainnya yaitu, untuk jenis inputan pada 2 form tanya usia sebaiknya tidak menggunakan rasio button, agar bisa di-update secara otomatis jika terjadi perubahan pada database, misalnya menggunakan *Combobox*[3].

Pada penelitian yang dilakukan oleh Seli Yuliana Cita Wardani, Andry Maulana, Ahmad Fauzi dan Fahrizal dengan judul “ Sistem Pakar Penditeksi Kerusakan *Hardware* Komputer Berbasis *Android* “. Masyarakat menggunakan komputer untuk memenuhi kebutuhan mereka, baik untuk membantu menyelesaikan pekerjaan, untuk belajar, ataupun untuk kegunaan lainnya. “Penggunaan komputer secara terus menerus tanpa adanya perawatan yang benar dapat membuat komputer mengalami kerusakan. komputer tersebut kurang memiliki pengetahuan tentang komputer, khususnya dalam menangani kerusakan hardware komputer. Mereka lebih memilih untuk menyerahkannya secara langsung kepada teknisi komputer bahkan memilih membeli yang baru. Menanyakan secara langsung kepada seorang ahli komputer tentunya memakan waktu dan biaya. Selain tidak efektif cara ini juga kurang efisien. Banyak sekali dana yang dikeluarkan untuk memperbaiki kerusakan komputer, padahal kerusakan komputer yang terjadi belum tentu rumit dan dapat diperbaiki secara mandiri. Untuk mengetahui lebih detail komponen yang

mengalami permasalahan dirasa perlu sebuah pengetahuan yang dapat memberikan informasi kepada pengguna komputer. Sehingga penggunapun dapat mencari solusi sendiri untuk menyelesaikan persoalan komputernya. Oleh karena itu pada penelitian yang berjudul “SISTEM PAKAR PENDETEKSI KERUSAKAN *HARDWARE* KOMPUTER BERBASIS *ANDROID*” Dalam penelitiannya algoritma yang digunakan adalah *Certainty Factor* dan *forward chaining* dan menggunakan *Android Studio* sebagai *Software Engine* serta *Adobe Photoshop* sebagai *Software Design* untuk *Mock Up* aplikasinya[4].

Dalam penelitian yang dilakukan oleh Budi Kurniawan mengembangkan “Aplikasi Sistem Pakar Berbasis Web untuk Diagnosa Penyakit Gigi dan Mulut “. Kesehatan memang menjadi barang mahal bagi manusia, oleh karena itu butuh kepekaan pribadi untuk menjaganya. Salah satu organ tubuh yang sering lalai kita jaga adalah gigi dan mulut. Data Dirjen Pelayanan Medik (2010) menunjukkan bahwa penyakit gigi dan mulut termasuk sepuluh ranking penyakit terbanyak di Indonesia. Berdasarkan survey Yayasan Kesehatan Gigi Indonesia menunjukkan bahwa 70% anak-anak menderita karies gigi dan gingivitis (Peradangan Gusi) sedangkan pada orang dewasa sebanyak 73% menderita karies gigi. Angka ini pun meningkat pada hasil riset Drg Herniyawati, M.Kes tahun 2017 menunjukkan 72% penduduk Indonesia mengalami gigi berlubang. Penjabaran diatas menjadi alasan dibutuhkannya sistem yang dapat menjadi tempat konsultasi pertama sebelum menindaklanjutinya dengan perawatan melalui dokter gigi. Dalam penelitiannya Budi Kurniawan menggunakan metode *Extreme Programming (XP)* yang merupakan bagian dari *AGILLE*, kemudian menggunakan *PHP* untuk bahasa pemrogramannya dan *MySQL* dan *Xampp* sebagai database satu paket dengan *Localhost*, kemudian sistem operasi yang digunakan adalah *Microsoft Windows*, untuk desain aplikasinya menggunakan *Microsoft Office Front Page 2003*[5].

Pada penelitian yang dilakukan oleh Jaenal Arifin dengan judul “Sistem Pakar Diagnosa Penyakit Gigi dan Mulut Manusia Menggunakan *Knowledge Base System* dan *Certainty Factor*”. Salah satu organ tubuh yang kurang mendapatkan perhatian adalah gigi dan mulut, padahal menjaga kesehatan gigi dan mulut itu sangat penting, sebab saraf gigi

berhubungan dan berpengaruh langsung dengan saraf organ tubuh lain. Sebagian besar orang mulai dari anak-anak hingga orang dewasa sangat enggan untuk memeriksakan gigi ke dokter gigi, apalagi jika tidak ada keluhan yang dirasakan. Di samping biaya konsultasi yang mahal, antrian yang panjang dan rasa sakit yang membayangi juga menjadi alasan orang takut ke dokter gigi. Selain itu, minimnya pengetahuan serta terbatasnya sumber informasi menyebabkan rendahnya kesadaran masyarakat terhadap upaya mencegah bahkan juga mengobati penyakit gigi dan mulut. Pada penelitiannya metode yang digunakan adalah *Isolasi Area Knowledge Base System* dimana pembuatan blok diagram dimaksudkan untuk membatasi lingkup permasalahan yang dibahas dengan mengetahui posisi pokok bahasan pada domain yang lebih luas dan algoritma yang digunakan adalah *Certainty Factor* dengan tujuan sistem menjadi tempat konsultasi pertama sebelum menindaklanjutinya dengan perawatan dokter[5].

Dimas Olga Nusantara, Sumbogo Wianu Pamungkas, Nur Rosyid Syaifudin, Lutfie Wijaya Kusuma dan Jamaludin Fikri yang dalam skripsinya berjudul “Sistem Pakar Analisis Penyakit Pada Tanaman Cabai merah Menggunakan Metode *Backward Chaining*”. Penanaman cabai pada musim hujan mengandung resiko, penyebabnya adalah tanaman cabai tidak tahan terhadap hujan lebat yang terus-menerus. Selain itu, genangan air pada daerah penanaman bisa mengakibatkan kerontokan daun dan terserang penyakit akar. Pukulan air hujan juga dapat menyebabkan bunga dan bakal buah cabai berguguran. Sementara itu, kelembapan udara yang tinggi meningkatkan penyebaran dan perkembangan hama serta penyakit tanaman. Tanaman cabai merah dalam proses budidayanya sering kali mengalami gangguan berupa penyakit yang dapat membuat tanaman rusak dan mati. dalam mengatasi penyakit di perlukan langkah preventif yaitu dengan memberikan penanganan khusus berupa pengobatan dan terapi yang benar terhadap tanaman yang terjangkit penyakit, kurangnya pemahaman dalam menanggulangi penyakit cabai merah sering kali menjadi kerugian tersendiri bagi petani Hal inilah yang mendorong perancangan sistem pakar untuk mendiagnosa penyakit pada tanaman cabai merah, dengan tujuan untuk memudahkan penanggulangan penyakit pada tanaman cabai merah dengan menggunakan sebuah

aplikasi sistem pakar. Pada skripsi ini simulasi diagnosa menggunakan metode *Backward Chaining*[6].

Dari referensi diatas, rancang bangun sistem pakar kerusakan hardware pada komputer ada persamaan contohnya pada metode perhitungan yang digunakan yaitu *Certainty Factor* seperti metode yang digunakan oleh oleh Seli Yuliana Cita Wardani, Andry Maulana, Ahmad Fauzi dan Fahrizal dan Jaenal Arifin. Sedangkan perbedaan terletak pada model pengembangan sistem yang digunakan dimana dari referensi di atas menggunakan metode *waterfall* sedangkan pada penelitian ini penulis menggunakan metode *prototyping* dan ada perbedaan pada studi kasusnya yang bertempat di *Insidious Computer* serta perbedaan pada basis aplikasinya yaitu berbasis *website*.

2.2. Landasan Teori

2.2.1. Sistem

Sistem merupakan suatu rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sistem adalah kelompok dari dua atau lebih atau subsistem yang saling berhubungan yang berfungsi dengan tujuan yang sama[7].

2.2.2. Pakar

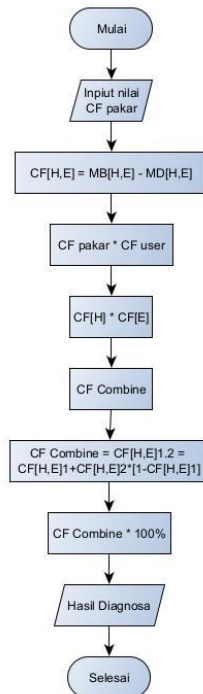
Pakar adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu penyakit. Contoh lain, montir adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor atau mobil, psikolog adalah orang yang ahli dalam memahami kepribadian seseorang, dan lain-lain (Kusrini, 2010:3)[8].

2.2.3. Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan tehnik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut Sistem pakar memberikan nilai tambah pada teknologi untuk membantu dalam menangani era informasi yang semakin canggih[9].

2.2.4. Metode Certainty Factor

Certainty factor adalah suatu metode untuk membuktikan apakah suatu fakta itu pasti ataukah tidak pasti yang berbentuk metric yang biasanya digunakan dalam sistem pakar. Metode ini sangat cocok untuk sistem pakar yang mendiagnosis sesuatu yang belum pasti.



Gambar 2 1 Alur *Certainty Factor*

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 [19].

2.2.5. Rekayasa Perangkat Lunak

Istilah Rekayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah Software Engineering. Istilah *Software Engineering* dipopulerkan tahun 1968 pada *Software Engineering Conference* yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (*Software*) dan program komputer. Ada beberapa pengertian menurut para ahli di bidang perangkat lunak. Pengertian dari rekayasa perangkat lunak menurut para ahli adalah sebagai berikut:

- 1). Menurut Stephen R. Schach : “Rekayasa perangkat lunak adalah sebuah disiplin dimana dalam menghasilkan perangkat lunak bebas dari kesalahan dan dalam pengiriman anggaran tepat waktu serta memuaskan keinginan pemakai.”
- 2). Menurut Fritz Bauer : “Rekayasa perangkat lunak adalah penetapan dan penggunaan prinsip rekayasa dalam rangka memperoleh perangkat lunak yang dapat dipercaya dan dapat bekerja secara efisien pada mesin nyata.”
- 3). Menurut Janner Simarmata (2010 :19): “Rekayasa perangkat lunak atau satu bidang profesi yang mendalami caracara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan aplikasi perangkat lunak dengan menerapkan teknologi dan peraktik dari ilmu komputer, manajemen organisasi pengembangan perangkat lunak, dan bidang-bidang lainnya.”[11].

2.2.6. UML

UML (unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek[12].

Dapat didefinisikan UML adalah standart bahasa untuk mendefinisikan dari requirement, membuat analisa & desain dan menggambarkan arsitektur dalam pemrograman yang berorientasi pada objek.




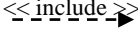
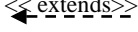
a) Use Case Diagram

A use case diagram is a behavior diagram in the Unified Modeling Language (UML). The use case diagram describes the functional

requirements of the software. Use case diagrams can be used to understand how the system should work. Therefore, the use case diagram is one of the diagrams taught in the computer science departments of various universities.

Diagram Use Case adalah diagram perilaku dalam Bahasa Pemodelan Terpadu (UML). Menggunakan diagram kasus menggambarkan persyaratan fungsional dari perangkat lunak. Use case diagram dapat digunakan untuk memahami bagaimana sistem harus bekerja. Karena itu, diagram use case adalah salah satu diagram yang diajarkan di departemen ilmu komputer dari berbagai universitas[13].

Tabel 2 1 Simbol *Usecase Diagram*



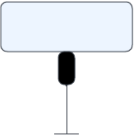
No	Simbol	Nama	Keterangan
1		<i>Actors</i>	Menggambarkan tokoh atau seseorang yang berinteraksi dengan sistem. Dan dapat menerima dan memberi informasi pada sistem.
2		<i>Use Case</i>	Menjelaskan fungsi dari kegunaan sistem yang di rancang.
3		<i>Association</i>	Menghubungkan antara <i>Use Case</i> dengan aktor tertentu.
4		<i>Include</i>	Menunjukkan bahwa <i>Use Case</i> satu merupakan bagian dari <i>Use Case</i> lainnya.
5		<i>Extends</i>	Menunjukkan arah panah secara putus-putus dari <i>Use Case</i> ke <i>Base Use Case</i>

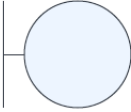

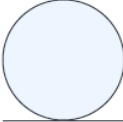


b) *Sequence Diagram*



Sequence Diagram adalah tool yang sangat populer dalam pengembangan sistem informasi secara *Object-Oriented* untuk menampilkan interaksi antar objek.

Selain itu *Sequence Diagram* dapat digunakan sebagai perangkat dalam perancangan antarmuka pemakai. Ada dua hal yang dapat dilakukan dengan *Sequence Diagram* pertama untuk menguraikan sebuah proses bisnis menjadi aktivitas-aktivitas yang lebih kecil untuk mengidentifikasi kebutuhan interaksi pemakai pada masing-masing aktivitas tersebut. Penggunaan kedua, *Sequence Diagram* digunakan pada setiap interaksi untuk menganalisa perilaku sistem informasi dalam rangka untuk merancang tampilan pada interaksi tersebut[14].

Tabel 2 2 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actors</i>	Merepresentasikan entitas yang berada di luar sistem dan berinteraksi dengan sistem
2		<i>Lifelines</i>	Menghubungkan objek selama sequence (message dikirim atau diterima dan aktifasinya).
3		<i>General</i>	Merepresentasikan entitas tunggal dalam sequence diagram.

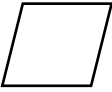



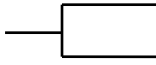
4		<i>Boundary</i>	Berupa tepi dari sistem, seperti user interface atau suatu alat yang berinteraksi dengan sistem yang lain.
5		<i>Control</i>	Element mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya mengatur perilaku dan perilaku bisnis.
		<i>Entity</i>	Elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model object
		<i>Activation</i>	Suatu titik dimana sebuah objek mulai berpartisipasi di dalam sebuah <i>Sequence</i> yang menunjukkan kapan sebuah objek mengirim atau menerima objek.
		<i>Message</i>	Befungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan
		<i>Message Entry</i>	Befungsi untuk menggambarkan pesan/hubungan antar objek yang


			menunjukkan urutan kejadian yang terjadi.
		<i>Message to Self</i>	Simbol ini menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
		<i>Message to Return</i>	Menggambarkan hasil dari pengiriman message dan digambarkan dengan arah dari kanan ke kiri.



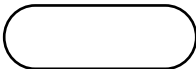





c) *Flowcharts*

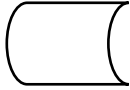

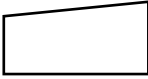
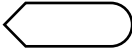
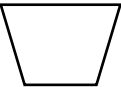


Flowchart adalah alat pemetaan sederhana yang menunjukkan urutan tindakan dalam proses dalam bentuk yang mudah dibaca dan dikomunikasikan.[15]. Berikut Nama, Simbol dan Fungsi dari *Flowcharts*:

Tabel 2 3 Simbol *Flowcharts*

No	Simbol	Nama	Keterangan
1		Input / Output	Mempresentasikan Input atau Output data.
2		Proses	Memprentasikan Operasi.
3		Penghubung	Keluar atau masuk ke bagian lain khususnya halaman yang sama.
4		Anak Panah	Mempresentasikan alur kerja.
5		Penjelasan	Digunakan untuk komentar tambahan

6		Keputusan	Keputusan program.
---	---	-----------	--------------------

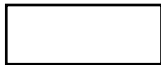
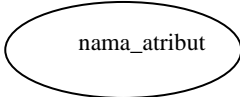
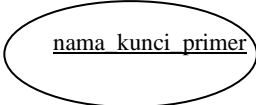
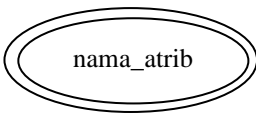
7		<i>Predefined Proses</i>	Rincian operasi berada di tempat lain.
8		<i>Preparation</i>	Pemberian harga awal.
9		<i>Terminal Points</i>	Awal/ akhir flowcharts
10		<i>Punched Card</i>	Input/Output yang menggunakan kartu berlubang.
11		<i>Dokumen</i>	I/O dalam format yang dicetak.
12		<i>Magnetic Tape</i>	I/O yang menggunakan pita magnetik.
13		<i>Magnetic Disk</i>	I/O yang menggunakan disk magnetik.
14		<i>Magnetic Drum</i>	I/O yang menggunakan drum magnetik.

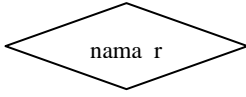
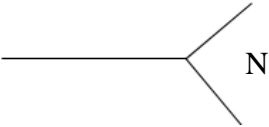
15		<i>On-Line Storage</i>	I/O yang menggunakan penyimpan akses langsung.
16		<i>Punched Tape</i>	I/O yang menggunakan kertas berlubang
17		<i>Manual Input</i>	Input yang dimaksukan secara manual dari keyboard
18		<i>Display</i>	Output yang ditampilkan pada terminal
19		<i>Manual Operation</i>	Operasi manual
20		<i>Communication Link</i>	Transmisi data melalui chanel komunikasi seperti telepon
21		<i>Off-Line Storage</i>	Penyimpanan yang tidak dapat di akses oleh komputer secara langsung

d) Simbol ERD

ERD (*Entity Relationship Diagram*) merupakan tools yang digunakan untuk memodelkan struktur data dengan menggambarkan entitas dan hubungan antar entitas (*relationship*) secara abstrak (*konseptual*). ERD berfungsi untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan symbol. Berikut Tabel 2.5 adalah daftar symbol dalam ERD :

Tabel 2 4 Simbol ERD

No	Simbol	Nama	Keterangan
1		Entitas / <i>entity</i>	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2		Atribut	Filed atau kolom data yang butuh disimpan dalam suatu entitas.
3		Atribut kunci primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom; asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4		Atribut multi nilai / multivalue	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih.

5		Relasi	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
6		Asosiasi / <i>Association</i>	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> yang menghubungkan entitas A dan entitas B.

2.2.7. Basis Data

Basis data merupakan kumpulan data dari berbagai entitas yang saling berhubungan. Basis data dikelola melalui sebuah perangkat lunak yang disebut database managements System (DBMS). Melalui DBMS manusia dapat berinteraksi dengan basis data[16].

Entity Relationship Diagram merupakan jaringan yang memiliki susunan data yang tersimpan di sistem secara abstrak. ERD adalah alat perancangan untuk memodelkan suatu basis data. Tujuan dibuatnya ERD dalam suatu organisasi adalah untuk pemodelan yang menunjukkan hubungan antar data yang mempunyai relasi, juga mendokumentasikan

data yang ada dengan cara menjelaskan tiap masing-masing data dan relasinya. Kardinalitas Relasi adalah jumlah maksimum entitas yang dapat berelasi (berhubungan) dengan entitas pada himpunan entitas yang lain.

Database Management System (DBMS) merupakan perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersil. MySQL sebenarnya merupakan turunan salah satu konsep utama database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis

2.2.8. Framework

Dalam pembuatan website untuk memudahkan dalam pengerjaan bisa menggunakan framework. Dalam hal ini penulis menggunakan framework codigniter untuk memudahkan dalam pengelompokkan code program. Menurut Novianto (2016:11) menyatakan bahwa, “Framework adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi website, diharuskan mengikuti aturan dari framework tersebut”. [17].

CodeIgniter merupakan framework PHP yang dibuat berdasarkan model view Controlleer (MVC). CI memiliki library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis web misalnya mengakses database, memvalidasi form sehingga sistem yang dikembangkan mudah. CI juga menjadi satu-satunya Framework dengan dokumentasi yang lengkap dan jelas. Source code CI

yang dilengkapi dengan comment didalamnya sehingga lebih memperjelas fungsi sebuah kode program dan CI yang dihasilkan sangat Bersih (clean) dan search Engine Friendly (SEF)[18].

2.7.9. Skala Likert

Skala Likert merupakan skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok orang tentang kejadian atau gejala sosial (Sugiyono, 2010:1993). Cara pengukuran adalah dengan menghadapkan seorang responden dengan sebuah pernyataan dan kemudian diminta untuk diminta jawaban dari lima pilihan jawaban, dimana nilai jawaban memiliki nilai jawaban yang berbeda[19].

2.7.9. Gejala dan Kerusakan

Pengembangan sistem pakar diagnosa kerusakan hardware komputer membutuhkan data gejala dan kerusakan. Ada beberapa data kerusakan dan gejala sebagai berikut:

Tabel 2 5 Data Kerusakan

No	Kode Kerusakan	Nama Kerusakan
1	K01	Kerusakan VGA
2	K02	Kerusakan RAM
3	K03	Kerusakan Mainboard
4	K04	Kerusakan Power Supply
5	K05	Kerusakan Hard Disk

Terdapat 5 data kerusakan, data tersebut masih bisa bertambah, berkurang maupun berubah menyesuaikan kebutuhan pengguna dan menyesuaikan data terbaru. Kerusakan di atas juga memiliki indikasi atau gejala-gejala yang mengarah pada kerusakan hardware. Dari 5 data kerusakan di atas terdapat 21 gejala, antara lain:

Tabel 2 6 Data Gejala

No	Kode Gejala	Nama Gejala
1	G01	Lag saat menonton video dan bermain game
2	G02	Warna yang muncul dilayar tidak sesuai
3	G03	Tampilan layar tidak sesuai resolusi monitor
4	G04	Blue Screen

5	G05	Layar seringkali kedip
6	G06	PC restart dengan sendirinya
7	G07	Layar gelap beberapa saat setelah dinyalakan
8	G08	PC mati dengan sendirinya
9	G09	PC jauh mengalami overheating (panas berlebih)
10	G10	Muncul artefak garis pada layar
11	G11	PC melambat
12	G12	Penyimpanan Penuh
13	G13	Data atau File mengalami kerusakan
14	G14	Komputer tidak mau melakukan proses booting meski perangkat komputer hidup
15	G15	Komputer tidak menyala sama sekali setelah menekan tombol power
16	G16	Komputer menyala namun selalu masuk ke safe mode
17	G17	Tidak dapat menyimpan, menduplikasi, memindahkan, dan menghapus file
18	G18	mendengar bunyi aneh dari hardisk
19	G19	Mendengar bunyi beep pada komputer
20	G20	Stuck pada bios
21	G21	RAM tidak terdeteksi atau jumlah tidak sesuai

~Halaman Ini Sengaja Dikosongkan~