

**SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN
KELAPA HIBRIDA MENGGUNAKAN METODE CBR
PADA DINAS PERTANIAN KABUPATEN
POHUWATO**

(Studi Kasus : Dinas Pertanian Kab. Pohuwato)

Oleh

YAYAN KIRAMAN

T3117345

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2021**

PERSETUJUAN SKRIPSI

**SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN
KELAPA HIBRIDA MENGGUNAKAN METODE CBR
PADA DINAS PERTANIAN KABUPATEN
POHUWATO**

(Studi Kasus : Dinas Pertanian Kab. Pohuwato)

Oleh

YAYAN KIRAMAN

T3117345

SKRIPSI

Untuk memenuhi salah satu syarat ujianguna memperoleh gelar Sarjana
Program Studi Teknik Informatika, ini telah disetujui oleh Tim Pembimbing

Gorontalo, 25 Juni 2021

Pembimbing I



Haditsah Annur, M.Kom
NIDN: 0908058403

Pembimbing II

Andi Kamaruddin, M.Kom
NIDN: 0909127601

PENGESAHAN SKRIPSI

SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN KELAPA HIBRIDA MENGGUNAKAN METODE CBR PADA DINAS PERTANIAN KABUPATEN POHUWATO

(Studi Kasus : Dinas Pertanian Kab. Pohuwato)

Oleh

YAYAN KIRAMAN

T3117345

Diperiksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo
Gorontalo, 25 Juni 2021


1. Ketua Penguji
Bahrin, S.Kom., MT
2. Anggota
Ruhmi Sulaehani, M.Kom
3. Anggota
Aprianto Alhamad, M.Kom
4. Anggota
Haditsah Annur, M.Kom
5. Anggota
Andi Kamaruddin, M.Kom

Mengetahui :

Dekan Fakultas Ilmu Komputer


Jorry Karim, M.Kom
NIDN: 0918077302

Ketua Program Studi


Sudirman S. Panna, M. Kom
NIDN: 0924038205


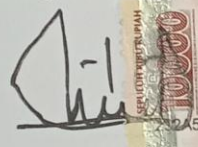
PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah hasil asli dan belum pernah di ajukan untuk mendapatkan gelar akademik (Sarjana) baik di universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah di publikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 25 Juni 2021

Yang Membuat Pernyataan,



YAYAN KIRAMAN

T3117345

ABSTRACT

This study discusses the development of a hybrid coconut disease expert system using the CBR Case Based Reasoning method. Which is a computer reasoning system that provides solutions to new cases by looking at old cases that are closest to new cases. This system is built with 7 symptoms with 6 pests and diseases which have different weight values. Based on the results of research that has been done, this Expert System can be implemented to diagnose Hybrid Coconut Plant Diseases. In this study using White Box testing by testing using Flowcharts and Flowgraphs with a Cyclomatic Complexity value of 8.

Key words: Expert System, Hybrid Coconut, CBR;

ABSTRAK

Penelitian ini membahas tentang pengembangan sistem pakar penyakit kelapa hibrida dengan menggunakan metode Case Based Reasoning CBR. Yang merupakan sebuah sistem penalaran komputer yang memberikan solusi terhadap kasus baru dengan melihat kasus lama yang paling mendekati kasus baru. Sistem ini dibangun dengan 7 gejala dengan 6 hama penyakit yang memiliki nilai bobot yang berbeda-beda. Berdasarkan hasil penelitian yang telah dilakukan Sistem Pakar ini dapat diimplementasikan untuk mendiagnosa Penyakit Tanaman Kelapa Hibrida. Pada penelitian ini menggunakan pengujian White Box dengan melakukan pengujian menggunakan *Flowchart* dan *Flowgraph* dengan nilai *Cyclomatic Complexity* 8.

Kata kunci : Sistem Pakar, Kelapa Hibrida, CBR;

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan usulan penelitian ini dengan judul **“SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN KELAPA HIBRIDA MENGGUNAKAN METODE CASE BASED REASONING (CBR)”**, untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materi. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Bapak Muhammad Ichsan Gaffar, SE., M.Ak, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo.
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo.
3. Ibu Zohrahayaty, M.Kom selaku Dekan Fakultas Ilmu Komputer.
4. Bapak Sudirman S. Panna, M.Kom selaku Pembantu Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo
5. Ibu Irma Surya Kumala, S.Kom, M.Kom, selaku Pembantu Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman Melangi, M.Kom selaku Pembantu Dekan III Bidang Kemasasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Irvan Abraham Salihi, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
8. Bapak Irvan Muzakkir, S.Kom, M.Kom, selaku Pembimbing I;
9. Bapak Ans, S.Kom, M.Kom, selaku Pembimbing II;
10. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;

11. Kedua Orang Tua saya yang tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
12. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
13. Kepada semua pihak yang ikut membantu dalam penyelesaian proposal ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat menghapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Juni 2021

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN JUDUL	Error! Bookmark not defined.
PERSETUJUAN SKRIPSI.....	Error! Bookmark not defined.
PENGESAHAN SKRIPSI	Error! Bookmark not defined.
<i>ABSTRACK</i>	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	2
1.3 Rumusan Masalah.....	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
BAB II LANDASAN TEORI	4
2.1 Tinjauan Studi.....	4
2.2 Tinjauan Teori.....	6
2.2.1 Sistem pakar.....	Error! Bookmark not defined.
2.2.2 Konsep Dasar Sistem Pakar	6
2.2.3 Stuktur Sistem Pakar.....	7
2.2.4 Keunggulan dan Kelemahan Sistem Pakar	8

2.2.5 Ciri-ciri Sistem Pakar.....	9
2.2.6 Terbatas pada keahlian tertentuManfaat Sistem Pakar	9
2.3 CASE BASED REASONING (CBR)	10
2.3.1 Definisi <i>Case Based Reasoning</i>	10
2.3.2 Cara kerja <i>Cara Based Reasoning</i>	10
2.3.3 Kelebihan case base reasoning :	12
2.3.4 Kelemahan case base reasoning :	12
2.4 Kelapa Hibrida.....	13
2.4.1 Definisi Kelapa Hibrida	13
2.4.2 Penyakit dan Gejala Serangan Hama.....	15
2.4.3 Pengembangan Sistem	16
2.4.4 Pendukung perangkat lunak.....	17
2.4.5 Teknik Penujian Sistem	19
2.4.6 Kerangka Pemikiran	21
BAB III OBJEK DAN METODE PENELITIAN.....	22
3.1 Objek Penelitian.....	22
3.2 Metode Penelitian	22
3.2.1 Analisis Sistem Berjalan.....	23
3.2.2 Analisis Sistem yang Diusulkan	23
3.2.3 Metode yang digunakan.....	23
3.2.4 Tahapan Identifikasi Sistem.....	23
3.2.5 Jenis dan Sumber Data.....	24
3.2.6 Metode dan Pengumpulan Data	24
3.2.7 Prosedur Pengujian Sistem	25
BAB IV HASIL PENELITIAN	28

4.1	Hasil Pengumpulan Data	28
4.2	Hasil Pemodelan Data.....	30
4.3	Hasil Pengembangan Sistem.....	31
4.3.1	Analisa Sistem	31
4.3.2	Analisis Sistem Berjalan	31
4.3.3	Analisa Sistem yang Diusulkan.....	33
4.3.4	Diagram Konteks.....	34
4.3.5	Diagram Berjenjang	35
4.3.6	Diagram Arus Data (DAD)	36
4.3.7	Kamus Data	37
4.3.8	Arsitektur Sistem/ Kebutuhan Hadware dan Software	41
4.3.9	Interface Design	42
4.3.10	Desain Secara Umum	42
4.3.11	Desain Secara Terinci.....	44
4.3.12	Hasil Kontruksi Sistem.....	50
4.3.13	Pengujian Sistem	50
BAB V PEMBAHASAN PENELITIAN		56
5.1	Pembahasan Model.....	56
5.2	Pembahasan Sistem	59
5.2.1	Halaman Menu Utama.....	59
5.3.2	Halaman Menu Gejala	60
5.3.3	Halaman Menu Konsultasi	60
5.3.4	Halaman Menu Data Penyakit	61
5.3.5	Halaman Menu Data Kasus	62
5.3.6	Halaman Menu Informasi Daftar Penyakit.....	63

BAB VI PENUTUP PENELITIAN	64
6.1 Kesimpulan	64
6.2 Saran	64
DAFTAR PUSTAKA.....	65
LAMPIRAN	66

DAFTAR GAMBAR

Gambar 2.1 : Fungsi sistem pakar	7
Gambar 2.2 : Struktur sistem pakar.....	8
Gambar 2.3 : PHP.....	17
Gambar 2.4 : MySQL.....	18
Gambar 2.5 : XAMPP	18
Gambar 2. 6 : Adobe Dreamweaver.....	18
Gambar 2.7 : Adobephotoshop CS.....	19
Gambar 2. 8 : Kerangka pikir.....	21
Gambar 4.1 : Sistem yang diusulkan.....	33
Gambar 4.2 : Analisis Sistem yang diusulkan.....	33
Gambar 4.3 : Diagram Konteks.....	34
Gambar 4.4 : Diagram Berjenjang	35
Gambar 4.5 : Diagram Arus Data Level 0.....	36
Gambar 4.6 : Diagram Arus Data Level 1 Proses 2	37
Gambar 4.7 : Diagram Arus Data Level 1 Proses 3	37
Gambar 4 8 : Desain Daftar Hasil Diagnosis	44
Gambar 4.9 : Desain Input Penyakit	45
Gambar 4.10 : Desain Input Gejala	45
Gambar 4.11 : Desain Input Solusi	46
Gambar 4.12 : Desain Input Penyakit	46
Gambar 4.13 : Desain Input Kasus.....	47
Gambar 4.14 : Desain menu Utama	49
Gambar 4.15 : Flowchart Gejala	51
Gambar 4.16 : Flowgraph Gejala	51
Gambar 4.17 : Screen Shoot Menu Login.....	52
Gambar 4.18 : Screen Shoot Halaman Utama.....	53
Gambar 4.19 : Screen Shoot Data Gejala.....	53
Gambar 4.20 : Screen Shoot Data penyakit	54
Gambar 4.21 : Screen Shoot Data Kasus	55

Gambar 5.1 : Halaman menu utama.....	59
Gambar 5.2 : Halaman menu Gejala	60
Gambar 5.3 : Halaman Menu Konsultasi	60
Gambar 5.4 : Halaman menu Data penyakit	61
Gambar 5.5 : Halaman menu data kasus	62
Gambar 5.6 : Halaman menu informasi daftar penyakit	63

DAFTAR TABEL

Tabel 2.1 : Tinjauan Studi	4
Tabel 4.1 : Daftar Macam Hama Kelapa	28
Tabel 4.2 : Daftar Gejala Penyakit Kelapa Hibrida	28
Tabel 4.3 : Daftar Solusi	29
Tabel 4.4 : Matriks Penyakit Tanaman Kelapa Hibrida.....	29
Tabel 4.5 : Tabel Bobot Parameter.....	30
Tabel 4.6 : Daftar bobot dari setiap gejala penyakit	30
Tabel 4.7 : Kasus Lama.....	30
Tabel 4.8 : Kasus Baru	31
Tabel 4.9 : Hasil Konsultasi	31
Tabel 4.10 : Hasil Diagnosa	31
Tabel 4.11 : Kamus Data Pengguna.....	37
Tabel 4.12 : Kamus Data Gejala	38
Tabel 4.13 : Kamus Data Kategori Gejala	38
Tabel 4.14 : Kamus Data Kasus	38
Tabel 4.15 : Kamus Data Kasus Gejala	39
Tabel 4.16 : Kamus Data Konsultasi.....	39
Tabel 4.17 : Kamus Data Konsultasi Gejala	40
Tabel 4.18 : Kamus Data Konsultasi hasil	40
Tabel 4.19 : Kamus Data Penyakit.....	40
Tabel 4.20 : Kamus Data Penyakit Solusi.....	41
Tabel 4.21 : Kamus Data Solusi.....	41
Tabel 4.22 : Desain Output Secara Umum.....	42
Tabel 4.23 : Desain Input Secara Umum	43
Tabel 4.24 : Tabel Penyakit.....	47
Tabel 4.25 : Tabel Gejala	47
Tabel 4.26 : Tabel Kasus_Gejala	48
Tabel 4.27 : Tabel konsultasi	48
Tabel 4.28 : Tabel Konsultasi Hasil.....	48

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kelapa (*Coco nucifera*) adalah anggota tunggal dalam marga *Cocos* dari suku aren-arenan atau *Arecaceae*. Arti kata kelapa (*coconut*, dalam Bahasa Inggris) dapat merujuk pada keseluruhan pohon kelapa, biji, atau buah, yang secara botani adalah pohon berbuah, bukan pohon kacang-kacangan. Istilah ini berasal dari kata Portugis dan Spanyol abad ke-16, *coco* yang berarti “kepala” atau “tengkorak” setelah tiga lekukan pada tempurung kelapa yang menyerupai fitur wajah. [1]

Kelapa dikenal karena kegunaannya yang beragam, mulai dari makanan hingga kosmetik. Kelapa berbeda dari buah-buahan lain karena endosperma mereka mengandung sejumlah besar cairan bening, hingga disebut “santan” dalam literatur, dan ketika belum matang, dapat dipanen untuk diminum sebagai “air kelapa”, atau juga disebut “jus kelapa” [2]

Kabupaten Pohuwato adalah daerah yang memiliki 13 kecamatan 3 kelurahan dan 101 desa dari Popayato sampai Paguat. Di daerah ini banyak para pengusaha kelapa, dan masyarakat yang ada di Pohuwato pekerja yang paling banyak di bidang kelapa dari usia anak-anak remaja sampai orang tua hampir rata-rata masyarakat yang ada di Buntulia tengah kecamatan Buntulia [3]

Sistem Pakar merupakan program komputer yang meniru proses pemikiran dan pengetahuan pakar dalam menyelesaikan suatu masalah tertentu. Implementasi sistem pakar banyak digunakan dalam bidang psikologi karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar pada bidang tertentu dalam program komputer sehingga keputusan dapat diberikan dalam melakukan penalaran secara cerdas.[4]

Metode Case Based Reasoning merupakan proses penyelesaian. Case Based Reasoning merupakan salah satu metode pemecahan masalah yang dalam mencari solusi dari suatu kasus yang baru, sistem akan melakukan pencarian terhadap solusi dari kasus lama yang memiliki permasalahan yang sama [5].

Berdasarkan paparan yang dijelaskan maka penulis melakukan penelitian dengan judul “Sistem Pakar Diagnosa Penyakit Tanaman Kelapa Hibrida Menggunakan Metode Case Based Reasoning pada Dinas Pertanian Kabupaten Pohuwato”, dengan harapan memodelkan dan memberikan solusi dengan cepat.

1.2 Identifikasi Masalah

Berdasarkan latar belakang, maka diuraikan identifikasi masalah dalam penelitian ini adalah sebagai berikut :

1. Kurangnya pemahaman masyarakat dalam menangani penyakit pada Tanaman Kelapa Hibrida
2. Pengembangan Kelapa Hibrida Mengalami penurunan jumlah produksi

1.3 Rumusan Masalah

Permasalahan yang akan dipecahkan dalam penelitian ini sesuai dengan penjelasan latar belakang diatas sehingga diuraikan sebagai berikut:

1. Bagaimana membangun suatu sistem pakar yang dapat mendiagnosa penyakit pada tanaman Kelapa Hibrida pada Dinas Pertanian Kabupaten Pohuwato?
2. Bagaimana hasil penerapan metode *Cased Based Reasoning* terhadap diagnosa penyakit pada Tanaman Kelapa?

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Hasil penerapan Metode *Cased Based Reasoning* mampu memberikan hasil diagnosa yang dapat membantu para pembudidaya Kelapa Hibrida.
2. Bisa membangun sistem pakar yang mendiagnosa penyakit yang selalu dan sering diderita Kelapa Hibrida Dengan menggunakan bahasa Pemograman PHP, dan aplikasi *dreamweaver*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi pihak-pihak yang membutuhkan, baik secara teoritis maupun praktis, diantaranya:

1. Manfaat teoritis: Penelitian ini diharapkan dapat menambah wawasan dan pengetahuan mengenai factor-faktor yang dapat mempengaruhi dalam sistem pakar, serta juga diharapkan sebagai sarana pengembangan Ilmu Pengetahuan yang secara teoritis di pelajari dibangku perkuliahan
2. Manfaat Praktis : Bisa memberikan kemudahan terhadap pihak Dinas Pertanian Kabupaten Pohuwato dalam menangani masalah yang timbul pada tanaman kelapa hibrida, sehingga populasi akan terus bertambah khususnya yang ada di Kabupaten Pohuwato.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Tabel 2.1 : Tinjauan Studi

No	Nama Peneliti	Tahun	Judul	Metode	Hasil
1	Irvan Mu-zakkir, Marniyati Husain Botutuhe	2020	Case Based Reasoning Method untuk Sistem Pakar Diagnosa Penyakit Sapi	Case Based Reasoning	Berdasarkan perhitungan CBR kasus Scource yang memiliki bobot paling rendah yaitu 0,09 sedangkan bobot yang paling tinggi dimiliki oleh kasus Pink Eye yaitu 1. Dalam proses ini memberikan solusi dengan kemiripan bobot dari kasus lama dengan kasus baru yang lebih tinggi. Pada kasus Pink Eye memiliki bobot lebih tinggi dan positif, solusi yang diberikan adalah pemberian anti alergi, anti biotik dan vitamin. Berdasarkan hasil yang diperoleh dapat ditarik kesimpulan bahwa penera-

					pan metode CBR baik digunakan untuk diagnosa penyakit ternak sapi yang sangat membantu peternak dalam menangani penyakit ternak sapi [5].
2	Amonius Asmin Hardi Saputra, Muhammad Syahrizal	2019	Perancangan aplikasi sistem pakar mendiagnosa Penyakit hemofilia pada manusia menerapkan Metode case based reasoning	Case Based Reasoning	Dalam mengetahui gejala-gejala Penyakit hemofilia pada manusia dengan sistem pakar dapat dilihat dari gejala-gejala yang dialami oleh pasien dan dilakukan perhitungan yang valid dalam membantu masyarakat untuk mengetahui penyakit hemofilia. Dengan menerapkan metode CBR dalam Mendiagnosa penyakit hemophilia pada manusia Dapat menghasilkan perhitungan Yang sama dengan perhitungan manual sehingga proses diagnosa dapat dilakukan dengan cepat dan akurat [6].

2.2 Tinjauan Teori

2.2.1 Sistem Pakar

Sistem Pakar adalah kecerdasan buatan berupa computer yang bisa digunakan untuk menyelesaikan suatu masalah, yang dimaksud pakar disini adalah seseorang yang memiliki ilmu atau keahlian yang bisa untuk menyelesaikan masalah yang tidak dapat di selesaikan oleh orang-orang awam. [7]

Yang dapat mengdiagnose dan memberikan penetalaksanaan suatu penyakit tidak semua orang yang dapat mengambil keputusan. Contoh yang lain, montir, karna montir adalah hanya seseorang yang memiliki keahlian hanya dalam bidang menangani kerusakan yang ada pada kendaraan, kemudian psikolog, psikolog juga hanya orang yang keahliannya hanya dapat memahami kepribadian dari orang lain, dan lain sebagainya.

komponen utama yang dimiliki sistem pakar itu ada dua, yaitu, mesin inferensi dan berbasis engetahuan. Mesin inferensi merupakan otak dari aplikasi sistem pakar, sedangkan berbasis pengetahuan adalah memori komputer atau sebagai tempat untuk menyimpan pengetahuan.

Pakar adalah seorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, kegiatan yang harus mampu dilakukan oleh sistem pakar adalah sebagai berikut :

1. Menerangkan pemecahannya
2. Mengenali dan menginformasikan permasalahan
3. Memecahkan permasalahan secara tepat dan tepat
4. Memecahkan aturan-aturan
5. Merekturisasi pengetahuan
6. Terutama belajar dari pengalaman

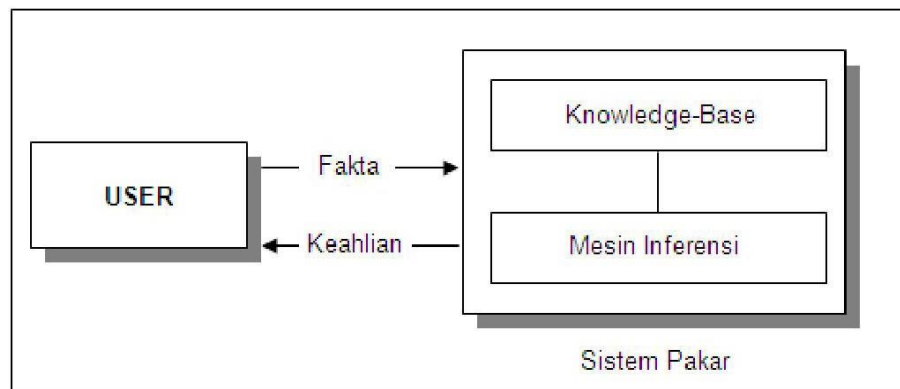
2.2.2 Konsep Dasar Sistem Pakar

Ada beberapa konsep dasar sistem pakar sebagai berikut [7]:

1. Mampu menjelaskan saran yang dijelaskan
2. Pengetahuan yang disimpan dalam prosedur-prosedur pemeca-

han masalah.

3. Mempunyai kemampuan dalam penalaran
4. Dapat mentransfer kepakaran kepada orang lain yang bukan pakar.\
5. Mampu menerapkan pengalaman dan pengetahuan untuk memecahkan suatu masalah.



Gambar 2.1 : Fungsi sistem pakar

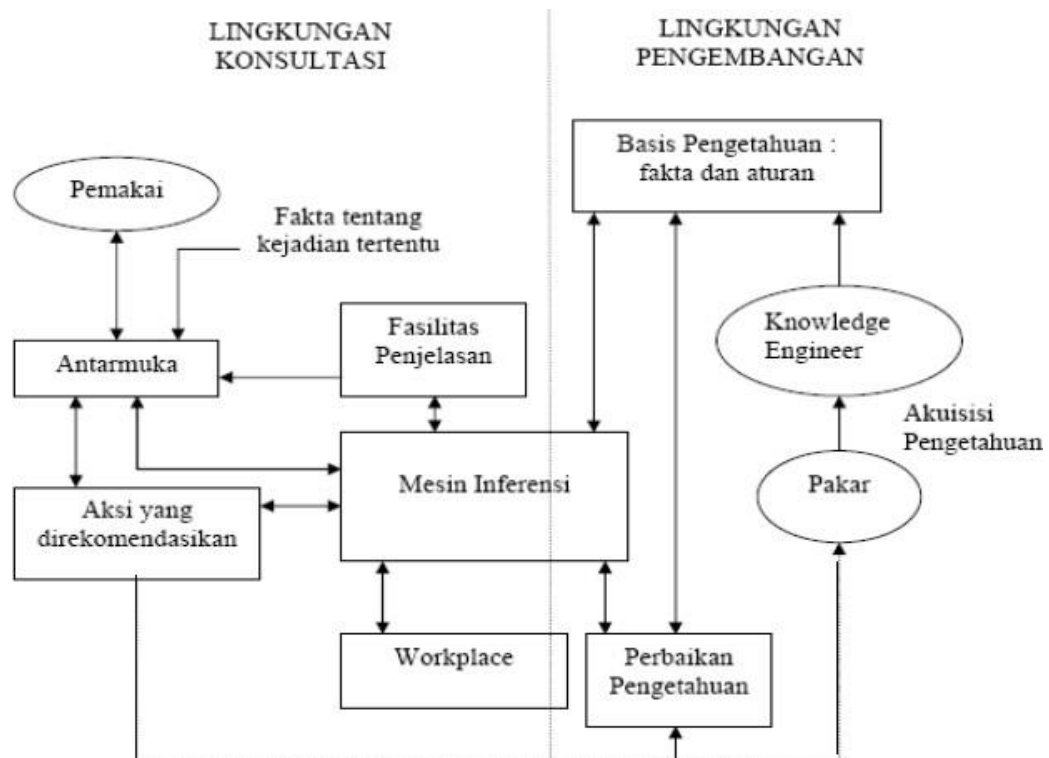
Gambar diatas menggambarkan konsep dasar sistem pakar knowledge-base. Yaitu menyampaikan informasi sistem pakar dan menerima masukan dari ahli atau jawaban pakarnya.

2.2.3 Stuktur Sistem Pakar

Bagian utama pada sistem pakar sebagai berikut :

1. Pengembangan : untuk memasukan keahlian pakar kedalam lingkungan sistem pakar
2. Konsultasi : untuk mendapatkan ilmu dari sistem pakar untuk pengguna yang bukan pakar

Struktur dari sistem pakar diberikan pada gambar dibawah ini :



Gambar 2.2 : Struktur sistem pakar

2.2.4 Keunggulan dan Kelemahan Sistem Pakar

Kelebihan-kelebihan dari sistem pakar sebagai berikut :

Manfaat yang sangat populer diberikan sistem pakar sebagai berikut

- Dapat diandalkan, sebab sistem pakar tidak pernah merasa bosan.
- Meningkatkan kualitas, dengan memberikan nasehat dengan konsisten.
- Membuat orang yang awam memiliki pekerjaan.
- Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia biasa.

Kekurangan –kekurangan dari sistem pakar sebagai berikut

Selain manfaat, sistem pakar juga memiliki kekurangan antara lain sebagai berikut,

- Sulit dikembangkan

- b. Biaya yang sangat mahal dibutuhkan untuk memelihara dan membuatnya.
- c. Tidak selalu bernilai benar.

2.2.5 Ciri-ciri Sistem Pakar

Ciri-ciri sistem pakar yaitu sebagai berikut :

- 6. Sistem dapat mengaktifkan kaidah secara searah.
- 7. Keluarannya bersifat anjuran.
- 8. Pengetahuan dan mekanisme penalaran.
- 9. Dirancang untuk mendapatkan perkembangan secara bertahap.
- 10. Berdasarkan rule atau ketentuan yang tertentu.
- 11. Dapat mengemukakan rangkaian alasan-alasan yang diberikannya dengan cara yang dapat dipahami.
- 12. Terbatas pada keahlian tertentu

2.2.6 Terbatas pada keahlian tertentu Manfaat Sistem Pakar

Sistem Pakar menjadi populer karena sangat banyak kemampuannya serta keuntungan yang dapat diberikannya, diantaranya :

- 1. Dapat digunakan sebagai pelengkap media dalam penelitian.
- 2. Bisa bekerja dengan informasi yang tidak lengkap atau tidak pasti.
- 3. Dengan sistem pakar komputer lain bisa membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
- 4. Handal, serta sistem pakar tidak pernah menjadi bosan dan kelebihan atau sakit.
- 5. Memudahkan akses pengetahuan seorang pakar.
- 6. Dapat beroperasi dilingkungan yang berbahaya.
- 7. Mampu mengangkat pengetahuan serta kepakaran yang dimiliki seseorang.
- 8. Meningkatkan kualitas.
- 9. Membuat seorang awam bekerja layaknya sistem pakar.
- 10. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat

2.3 CASE BASED REASONING (CBR)

2.3.1 Definisi *Case Based Reasoning*

Case Based Reasoning (CBR) menggunakan kecerdasan buatan (Artificial Intelligent) yang menitikberatkan pemecahan masalah dengan di dasarkan pada knowledge dari kasus-kasus sebelumnya secara umum, metode ini terdiri dari 4 langkah, yaitu:.

1. Retrieve, mendapatkan kasus-kasus yang mirip.
2. Reuse, menggunakan kembali kasus-kasus yang ada dan dicoba untuk menyelesaikan suatu masalah sekarang.
3. Revise, merubah dan mengadopsi solusi yang ditawarkan jika perlu.
4. 4. Retain, memakai solusi baru sebagai bagian dari kasus baru, kemudian kasus baru diupdate kedalam basis kasus.

2.3.2 Cara kerja *Cara Based Reasoning*

Case Based Reasoning menggunakan pendekatan kecerdasan buatan (artificial intelligent) yang mengutamakan pemecahan masalah dengan berdasarkan pada pengetahuan dari kasus-kasus sebelumnya, apabila ada kasus yang baru maka kasus tersebut akan tersimpan pada basis pengetahuan sehingga sistem akan melakukan pembelajaran dan pengetahuan terhadap kasus-kasus sebelumnya yang dimiliki. Secara umum ada 4 langkah dalam case based reasoning:

1. Retrieve (memperoleh kembali)

Pada proses Retrieve ini kita mendapatkan kembali kasus yang sama atau yang mirip dengan kasus baru yang baru kita temui. Dalam proses ini, tahapan yang dapat kita lakukan adalah identifikasi masalah, memulai pencocokan, dan seleksi.

2. Reuse (menggunakan kembali)

Pada proses Reuse ini, sistem akan melakukan pencarian masalah pada database melalui identifikasi masalah baru. Setelah itu, sistem akan menggunakan

kembali informasi permasalahan yang pernah terjadi tersebut yang memiliki kesamaan untuk menyelesaikan permasalahan yang baru. Proses Reuse dipusatkan pada dua aspek. Pertama, perbedaan antara kasus sebelumnya dengan kasus sekarang. Kedua, bagian dari kasus yang lama yang sudah diperoleh akan dikirimkan menjadi kasus baru. Ada dua cara yang dapat digunakan untuk melakukan reuse kasus yang sudah ada. Pertama, Reuse solusi (transformational reuse), Kedua, Reuse yang dapat membuat solusi (derivational reuse).

3. Revise (meninjau kembali / memperbaiki)

Pada proses Revise ini akan dilakukan tinjauan kembali/memperbaiki solusi-solusi yang sudah didapat pada masalah tersebut. Ada dua tugas pokok dari tahapan Revise ini, di proses ini solusi yang sudah diperoleh dari proses Rause akan dievaluasi kembali. Jika berhasil, maka akan langsung dilanjutkan ke proses selanjutnya yaitu proses retain. Jika tidak, sistem akan memperbaiki lagi solusi kasus yang diperoleh dari proses retain dengan menggunakan domain spesifik pengetahuan.

4. Retain (menyimpan)

Pada proses Retain, bisa dibilang proses ini adalah yang terakhir di dalam system Case Based Reasoning. Di dalam proses system ini akan menyimpan permasalahan yang baru lalu dimasukan ke dalam basis pengetahuan, setelah itu akan digunakan untuk menyelesaikan permasalahan yang akan datang.

Di dalam proses Case Based Reasoning, ada salah satu tahapan yang paling penting dalam proses penyelesaian masalah ini, yaitu proses pengambilan kasus (case retrieval). Sejak zama dahulu sudah banyak peneliti yang memfokuskan di tahapan ini (Pengambilan Keputusan). Di dalam proses pengambilan keputusan, persamaan antara kasus satu dan kasus lain dijadikan sebagai dasar dalam pengambilan sebuah kasus di dalam basis kasus. Semakin besar persamaan yang dimiliki oleh suatu kasus dengan kasus yang baru di dalam basis kasus, maka memungkinkan solusi yang terdapat pada kasus tersebut bisa digunakan untuk me-

nyelesaikan masalah di dalam kasus yang baru. Penalaran berbasis kasus ini biasanya dipakai untuk: Diagnosis, Manajemen Pengetahuan dan pendukung keputusan. Dalam hal ini tingkat kecocokan kasus akan sangat berpengaruh pada kerja sistem Case Based Reasoning, itu dikarenakan solusi-solusi yang ada di dalam kasus sebelumnya akan digunakan kembali sebagai patokan dalam penyelesaian masalah baru. Sudah banyak peneliti yang menggunakan berbagai macam algoritma di dalam kasus retrieval.

2.3.3 Kelebihan case base reasoning :

- Memecahkan masalah dengan mudah karena dapat mengambil solusi dengan cepat dan tepat.
- Semakin banyak pengalaman yang tersimpan di dalam system maka system akan semakin pintar dalam menemukan solusi untuk sebuah kasus.
- Biasanya langsung fokus pada fitur terpenting pada masalah tersebut.
- Dapat memecahkan masalah dalam domain yang hanya dapat dipahami sebagian.
- Dapat memberikan solusi jika tidak ada metode algoritmik yang tersedia.
- Dapat menafsirkan konsep terbuka dan tidak jelas.

2.3.4 Kelemahan case base reasoning :

- Tidak menjamin solusi yang didapat itu menjadi solusi terbaik atau maksimal, karena dalam system Case Based Reasoning ini sangat bergantung pada kasus yang pernah terjadi, maka dari itu jika solusi dari kasus yang pernah terjadi itu salah, maka dalam hal ini tahapan revise sangat diperlukan untuk mengurangi tingkat kesalahannya.
- Namun dalam hal ini, jika semakin banyak pengalaman yang tersimpan di basis data, maka tidak menutup kemungkinan dalam menyelesaikan suatu kasus baru akan menjadi lama. Itu dikarenakan, system ini akan mencari kasus-kasus yang paling mirip.

2.4 Kelapa Hibrida

2.4.1 Definisi Kelapa Hibrida

Pengertian Kelapa Hibrida pengertian Buah Kelapa Hibrida Atau Timun merupakan buah yang sering digunakan sebagai lalapan makanan. Buah ini juga memiliki cukup gizi yang bagus untuk dikonsumsi tubuh. Kelapa Hibrida bentuknya lonjong dan panjang dengan warna hijau tua pada kulitnya. Bagian dalam buah ini cenderung hijau dan sedikit putih pada bagian tengahnya. Selain lalapan, buah ini juga sering dijadikan pelengkap acar dan campuran makanan lainnya sebagai penambah kesegaran [8].

Pengertian Kelapa Hibrida Dan Manfaat Kelapa Hibrida - Kelapa Hibrida sangat gampang untuk ditemui di sekitar kita. Buah Kelapa Hibrida ini sering dikira sebagai sayur karena kebiasaan masyarakat yang sering mengolahnya dengan sayuran lain. Kandungan yang terdapat pada buah ini yaitu air, kalori, lemak, protein, lemak, karbohidrat, vitamin K, vitamin C, magnesium, kalium, mangan, vitamin A, thiamin, riboflavin, folat, zat besi, fosfor, seng, dan juga tembaga. Mari kita urai lebih jauh tentang Pengertian Kelapa Hibrida Dan Manfaat Kelapa Hibrida untuk kesehatan

Manfaat Buah Kelapa Hibrida Untuk Kesehatan

13. Mampu mencegah peradangan Mengonsumsi timun sangat berkhasiat untuk mencegah peradangan, cara mengkonsumsinya dapat secara langsung atau dijadikan sebagai campuran sirup atau dijadikan jus bersama buah-buahan lainnya.
14. kelapa mampu melancarkan pencernaan Bagi anda yang sedang memiliki masalah dengan pencernaan atau susah buang air besar, mengonsumsi timun dapat menjadi solusi yang tepat dalam melancarkan sistem pencernaan anda. Hal ini dikarenakan, buah timun memiliki kandungan serat alami yang sangat berkhasiat serta dapat memberikan efek dingin yang dapat meringankan rasa mulas atau sakit perut.
3. Dapat membantu menyembuhkan penyakit paru-paru, cacingan dan arthritis Mengonsumsi timun secara rutin terbukti sangat berkhasiat dalam

membantu masa penyembuhan bagi mereka yang menderita penyakit paru-paru, cacingan dan arthritis.

4. Mencegah dehidrasi dan meningkatkan daya tahan tubuh
Timun kaya akan kandungan air nya yang mencapai 95% , sehingga sangat ampuh dalam mencegah tubuh terkena dehidrasi serta membantu membuang racun di dalam tubuh. Selain itu, kandungan vitamin C nya mampu menangkal serangan virus dan bakteri yang dapat menyebabkan timbulnya penyakit, sehingga daya tahan tubuh akan tetap terjaga.
5. Menurunkan kadar gula dan kolesterol dalam tubuh
Mengonsumsi jus timun secara rutin dapat memenuhi hormon yang sangat dibutuhkan oleh sel-sel pankreas untuk memproduksi insulin di dalam tubuh. Hormon tersebut mampu mengontrol kadar gula dalam darah bagi seseorang yang menderita diabetes. Selain itu, kandungan senyawa sterol nya dapat membantu dalam menurunkan kadar kolesterol di dalam tubuh.
6. Menurunkan tekanan darah tinggi
Bagi anda yang memiliki masalah dengan tekanan darah tinggi (hipertensi), meminum jus timun merupakan solusi yang tepat untuk membantu menurunkan tekanan darah. Hal ini dikarenakan, timun mengandung kalium, magnesium dan serat alami yang sudah terbukti keampuhannya.
7. Bermanfaat untuk kecantikan kulit
Untuk merawat kecantikan seperti menghaluskan kulit, menghilangkan jerawat, mengatasi kulit berminyak, dan mengatasi masalah kesehatan rambut. Hal ini dikarenakan, timun kaya akan nutrisi dan mineral yang sangat dibutuhkan oleh kulit dan rambut kita.
8. Mengobati penyakit mata rabun
Mengobati penyakit mata seperti mata rabun. Caranya cukup mudah yaitu dengan menaruh irisan Kelapa Hibrida pada bagian atas alis dan diamkan sekitar 15 menit. Lakukan rutin setiap hari. Maka berangsur-angsur mata anda akan jelas kembali dalam melihat. Dengan diletakkannya timun diatas alis maka akan memperbaiki urat syaraf yang berhubungan ke mata.

2.4.2 Penyakit dan Gejala Serangan Hama

Penyakit **Busuk Pucuk Kelapa** disebabkan oleh jamur *Phytophthora palmivora*. Jamur ini bertahan pada jaringan sakit atau hidup sebagai saprofit pada tanah. Penularan penyakit melalui spora dari tanaman sakit dengan perantara air hujan, angin, atau serangga, khususnya **kumbang tanduk** (*Oryctes rhinoceros*). Selain menyerang kelapa, jamur tersebut dapat juga menyerang tanaman karet, kakao, lada, kina, jeruk dan pepaya.

Sexava nubila Dari hasil pengamatan di lapangan, didapati gejala serangan yang terlihat sama seperti yang diakibatkan oleh hama Sexava nubila, dimana pada bagian daun kelapa yang diamati terlihat seperti daun terpotongpotong tidak beraturan dan pelepah daun hanya tinggal lidinya saja, dan pada umumnya adalah daun kelapa yang sudah tua. Menurut Hosang (2005), bahwa hama Sexava nubila memakan anak daun yang dimulai dari pinggir ke bagian tengah, kadang-kadang dimakan sebagian atau sampai ke lidi. Bekas gigitan biasanya tidak rata, dan pada serangan berat terlihat pada pelepah daun bagian bawah tinggal lidi saja. Kalshoven (1991) juga mengemukakan, bahwa jenis belalang Sexava spp. biasanya merusak dengan cara memakan helaian daun, dan selain merusak tanaman kelapa, juga merusak tanaman lainnya seperti sagu, pinang, salak, dan lain sebagainya. Selanjutnya Setyamidjaja (1995) menambahkan, bahwa belalang Sexava nubila juga merusak bagian daun muda, kulit buah, dan bungabunganya.

Hasil pengamatan juga menemukan telur, nimfa dan imago Sexava nubila. Di Desa Kamarian ditemukan beberapa butir telur berwarna coklat kekuningan, berbentuk seperti bulir padi. Di Desa Waipirit ditemukan nimfa berwarna hijau-coklat, sedangkan imago jantan berwarna hijau, antenanya halus seperti rambut dan panjangnya 9 cm dan tanpa ovipositor yang ditemukan di desa Kamarian (Gambar 2). Ciri-ciri tersebut sesuai dengan yang dikemukakan oleh Hosang (2005), yaitu bentuk dan warna telur Sexava nubila. seperti buah padi masak (gabah), salah satu ujung telur lancip dan lainnya bulat, telur tua panjangnya sampai 13mm dan lebarnya 3mm. Nimfa muda dan tua berwarna hijau, tetapi kadang-kadang berwarna coklat. Imago berwarna hijau, antena merah muda dan matanya abu-abu. Alat peletak telur (ovipositor) berwarna hijau pada bagian pangkalnya

yaitu sepertiga dari panjang ovipositor, sepertiga lagi berwarna kemerahan dan bagian ujungnya berwarna hitam. Panjang imago betina (kepala + badan + ovipositor) antara 9.5 – 10.5 cm. Panjang ovipositor 3 – 4.5 cm dan panjang antena 16 cm. Panjang imago jantan 6 – 9.5 cm dan antenanya 14-16 cm.

Makanan kumbang dewasa adalah tajuk tanaman, dengan menggerek melalui pangkal batang sampai pada titik tumbuh. Daun yang telah membuka memperlihatkan bentuk seperti huruf V terbalik atau karakteristik potongan serrate (Sadakhatula & Ramachandran, 1990). Menurut Kartasapoetra (1990) bahwa *O. rhinoceros* selain memakan daun kelapa, juga bagian buah yang masih muda, sehingga buah mengalami kegagalan, kering terpaksa dan kalau dikupas sangat sukar, biasanya buah-buah demikian jatuh sebelum waktunya.

2.4.3 Pengembangan Sistem

2.4.3.1 Pengertian Sistem

Pengertian sistem adalah : “A sistem is group of elements that are integrated with the common porpose of achieving an objective.” Sistem adalah sekelompok elemen yang terintegritas dengan maksud yang sama untuk mencapai tujuan [9].

2.4.3.2 Anasisi Sistem

Anasisi system adalah sebagai komponen yang utuh dari suatu sistem dengan tujuna dapat mengevaluasi permasalahan, kesempatan yang misal terjadi sehingga untuk penanganannya dapat diketahui.

Jadi seandainya ditahap ini memiliki kesalahan, maka bisa juga akan menyebabkan kesalahan ditahap yang selanjutna.

a. Studi kelayakan

Untuk mendapatkan keberhasilan dari solusi yang di usulkan, kita bisa menggunakan studi kelayakan.

b. Analisis kebutuhan

untuk mendapatkan spesifikasi kebutuhan yang harus kita lakukan adalah mencari spesifikasi yang rinci tentang hal-hal yang dilakukn oleh sistem.

Sekaligus spesifikasi digunakan supaya memuat kepakatan antara pengembang.

2.4.3.3 Desain Sistem

Desain sistem adalah berupa output yang dikehendaki untuk menganalisa persyaratan yang diperlukan untuk menyiapkan data yang perlu disimpan dalam file. Tahap perencanaannya sebagai berikut :

1. Merancang penerapan file- file yang dibutuhkan datanya.
2. Sistem dirancang secara global.
3. Pembuatan program untuk mendesain sistem.

2.4.4 Pendukung perangkat lunak

Penulis menggunakan perangkat lunak pendukung dengan menggunakan PHP, XAMMP, MySQL, *Dreamweave*, dan *photoshop* [10]

2.4.4.1 PHP (PHP; Hypertext Preprocessor)

PHP (*Personal Home Page*) adalah pemograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”. Sedangkan menurut penulis lain “PHP (atau resminya PHP: *Hypertext Preprosesor*) adalah skrip bersifat Bersifat server-side yang di tambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari Personal Home Page Tools. Skrip ini akan membuat suatu aplikasi dapat di integrasikan ke dalam HTML sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat server side berarti pengerjaan script dilakukan di server, baru kemudian hasilnya dikirimkan ke browser”.



Gambar 2.3 : PHP

2.4.4.2 MySQL

mengungkapkan: “MySQL merupakan RDBMS atau server database yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat diakses oleh banyak user”.



Gambar 2.4 : MySQL

2.4.4.3 XAMPP

“XAMPP (*X Apache MySQL PHP Perl*) merupakan paket PHP dan Mysql berbasis *open source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP”.



Gambar 2.5 : XAMPP

2.4.4.4 Adobe Dreamweaver

, untuk mendesain dengan bagus yang perlu kita gunakan adalah *Adobe Dreamweaver*.



Gambar 2. 6 : Adobe Dreamweaver

2.4.4.5 Adobe Photoshop

Adobe Photoshop merupakan salah satu aplikasi untuk mengedit gambar, dengan aplikasi ini kita dapat mempercantik atau memperbaiki foto yang ingin kita edit, bahkan bias menambahkan efek dalam gambar tersebut. .



Gambar 2.7 : Adobephotoshop CS

2.4.5 Teknik Penujian Sistem

2.4.5.1 White Box

White Box merupakan cara untuk menguji aplikasi dengan cara meneliti kode program yang dibuat ada yang salah atau tidak.

White Box bias dilakukan dengan pengujian *basis path*, cara ini merupakan salah satu teknik struktur control untuk menjamin satu kali. Sebelum menghitung nilai, harus diterjemahkan ke grafik alir, kemudian dilanjutkan pembuatan flowgraphnya.

2.4.5.2 Black box

Black Box adalah pengujian tingkah laku, yang tertuju pada kebutuhan perangkat lunak. Persyaratan fungsional untuk suatu program ini menggunakan pengujian *Black Box* sehingga memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya.

Kesalahan yang ditemukan dalam kategori pengujian *Black Box*:

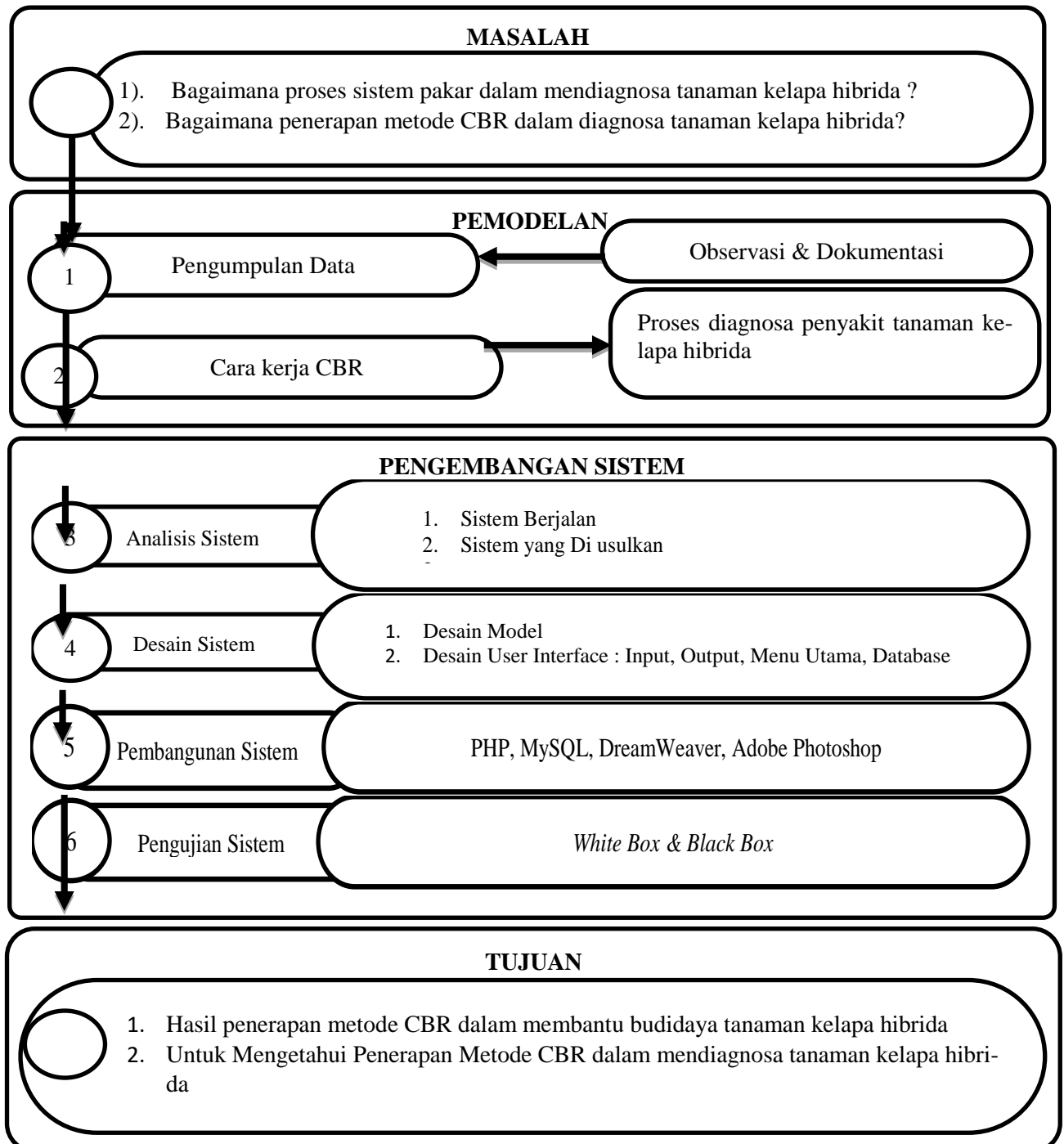
- a. Kesalahan performasi.
- b. Kesalahan inisialisasi.

- c. Kesalahan pada struktur data
- d. Kesalahan antar muka
- e. Fungsi tidak benar.

Nah, pengujian ini bisa tercapai karena melalui sebagai berikut :

1. Analisis nilai batas : berdasarkan nilai batas domain pengujian ini disebut pengujian back-to-back yang diterapkan pada suatu cersi perangkat lunak.
2. *Equibalence Partitioning* : untuk pengujian agar diperoleh kelas misal data atribut, atau karakter yang lain.
3. pengujian *graph-based* : membuat sekumpulan node yang merepsesentasikan onjek misal layar baru dengan atributnya.

2.4.6 Kerangka Pemikiran



Gambar 2. 8 : Kerangka pikir

BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Yang mendasari objek penelitian ini ialah kerangka pikir semacam yang telah di jelaskan pada uraian bab-bab sebelumnya, maka objek yang menjadi penelitian kali ini adalah Sistem Pakar Diagnosa Penyakit pada tanaman Kelapa Hibrida dengan Metode *Case Based Reasoning* (CBR). Pada Dinas Pertanian Kabupaten Pohuwato

3.2 Metode Penelitian

Yang dilakukan dalam metode kali ini adalah metode deskriptif, dan proses yang dilakukan yaitu untuk berusaha mencari solusi dari masalah yang berdasarkan data yang ada sekarang, serta menginterpretasikan dan menganalisa. Cara ini dilakukan agar supaya memperbaiki atau menggantikan sistem yang ada kekeliruan sebelumnya, contoh kesalahannya sebagai berikut :

- a. Adanya permasalahan-permasalahan (*problem*) yang timbul di sistem yang lama.
Permasalahan yang timbul dapat berupa ketidakberesan dan pertumbuhan organisasi.
- b. Untuk meraih kesempatan-kesempatan (*opportunities*)
- c. Adanya instruksi-instruksi (*directives*)

System life cycle adalah sederhana dan masuk akal. Di sistem *life cycle*, tiap-tiap bagian dari pengembangan sistem dibagi menjadi beberapa tahapan kerja. Tiap-tiap tahapan ini mempunyai karakteristik tersendiri. Tahapan utama siklus hidup pengembangan sistem dapat terdiri dari tahap[an perencanaan sistem (*system planning*), analisis sistem (*system analysis*), desain sistem (*system design*), seleksi sistem (*system selection*), implementasi sistem (*system implementation*), dan penawaran sistem (*system maintenance*). Tahapan-tahapan seperti ini sebenarnya merupakan tahapan didalam pengembangan sistem teknik (*engineering sys-*

tem).

3.2.1 Analisis Sistem Berjalan

Pada tahap ini dilakukan analisis kebutuhan dan masalah dalam merancang sistem yang akan dibuat, kemudian menetapkan sistem juga harus memperhatikan representasi parameter yang akan digunakan, sehingga sistem pakar untuk mendiagnose sesuai dengan kebutuhan pengguna, agar dapat membantu melakukan diagnose Penyakit Pada Tanaman Kelapa Hibrida sesuai dengan Pakarnya.

3.2.2 Analisis Sistem yang Diusulkan

Pada tahap ini dilakukan pendalaman tentang kejelasan sasaran, kejelasan tujuan dari sistem Pakar diagnose Penyakit Pada Tanaman Kelapa Hibrida, kejelasan sistem yang akan direkayasa serta bimbingan teknis penggunaan sistem. Secara umum dapat digambarkan bahwa sistem yang akan dibangun merupakan sebuah sistem pakar yang menggunakan *Case Based Reasoning*

3.2.3 Metode yang digunakan

Metode penelitian yang digunakan pada penelitian ini adalah metode *Case Based Reasoning*,

3.2.4 Tahapan Identifikasi Sistem

Dalam penelitian ini akan dibagi menjadi beberapa tahapan kegiatan yaitu Identifikasi sistem yang dilakukan untuk mendapatkan karakteristik sistem yang ada saat ini, dan identifikasi kebutuhan sistem yang terbagi sebagai berikut:

- a. Mengidentifikasi Penyebab masalah
- b. Mengidentifikasi Titik keputusan
- c. Mengidentifikasi Personel kunci
- d. Menentukan Jenis penelitian
- e. Merencanakan Jadwal penelitian
- f. Membuat penugasan penelitian/*survey*
- g. Membuat agenda wawancara

h. Mengumpulkan Hasil Penelitian

3.2.5 Jenis dan Sumber Data

1. Data primer, data yang diperoleh dari sumbernya dengan melakukan wawancara terhadap Pegawai yang paham terhadap Penyakit Pada Tanaman Kelapa Hibrida pada Kantor Dinas Pertanian Kabupaten Pohuwato
2. Data skunder, data yang diperoleh secara tidak langsung bersumber dari dokumentasi, literature, buku, jurnal, dan informasi lainnya yang ada hubungannya dengan masalah yang diteliti.

3.2.6 Metode dan Pengumpulan Data

Untuk mendapatkan data yang benar-benar akurat, relevan, valid dan dapat dipercaya maka pengumpulan data dilakukan sebagai berikut:

- a. *Observasi*, Pengumpulan data melalui pengamatan dan pencatatan terhadap berbagai bentuk laporan-laporan yang akan dihasilkan dari sistem pakar. Pengamatan dilakukan di Kantor Dinas Pertanian Kabupaten Pohuwato.
- b. *Interview*, Pengumpulan data dengan cara tatap muka dan Tanya jawab langsung dengan sumber data, yaitu Pegawai Dinas Pertanian Kabupaten Pohuwato sebagai pakar dari Sistem ini.
- c. *Survey*, Pengumpulan data dengan melakukan *survey* ke Dinas Pertanian Kabupaten Pohuwato
- d. *Studi pustaka*, Pengumpulan data dengan cara membaca serta mempelajari dokumen-dokumen, literature, buku, jurnal, video/gambar yang berhubungan dengan obyek penelitian guna mendapatkan teori ataupun konsep yang dapat digunakan sebagai landasan teori dan kerangka pemikiran dalam penelitian dan untuk mencari metodologi yang sesuai serta membandingkan teori yang ada dengan fakta yang ada di lapangan.

3.2.7 Prosedur Pengujian Sistem

Sebelum program diterapkan, maka program harus bebas terlebih dahulu dari kesalahan-kesalahan. Oleh sebab itu program harus diuji untuk menentukan kesalahan-kesalahan yang mungkin dapat terjadi. Program dites untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai. Kesalahan dari program yang mungkin terjadi dapat diklasifikasikan dalam tiga bentuk kesalahan, yaitu sebagai berikut:

1. Kesalahan bahasa (*language errors*) atau disebut juga dengan kesalahan penulis (*syntax errors*) atau kesalahan tata bahasa (*grammatical errors*) adalah kesalahan didalam penulisan *source program* yang tidak sesuai dengan yang telah disyaratkan. Kesalahan ini relatif ditemukan dan diperbaiki, karena kompiler akan memberitahukan letak dan sebab kesalahannya sewaktu program dikompilasi.
2. Kesalahan sewaktu proses (*run time errors*), adalah kesalahan yang terjadi sewaktu *executable program* dijalankan. Kesalahan ini akan menyebabkan proses program berhenti sebelum selesai pada saatnya, karena compiler menemukan kondisi-kondisi yang belum terpenuhi yang tidak bisa dikerjakan. Kesalahan ini juga relatif mudah ditemukan, karena juga ditunjukkan letak serta sebab kesalahannya.
3. Kesalahan logika (*logical errors*), adalah kesalahan dari logika program yang dibuat. Kesalahan seperti ini sulit ditemukan, karena tidak adapemberitahuan mengenai kesalahan dan tetap akan didapatkan hasil dari proses program, tetapi hasilnya salah. Kesalahan seperti ini merupakan kesalahan yang berbahaya karena bila tidak disadari dan tidak ditemukan, hasil yang salah dapat menyesatkan bagi yang menggunakannya. Cara mencari kesalahan logika dapat dilakukan dengan *test data*, yaitu dengan menjalankan program dengan menggunakan data tertentu dan membandingkan hasil pengolahannya dengan hasil yang sudah diketahui. Bila hasilnya berbeda, berarti mengalami kesalahan dan harus di lacak serta ditemukan sebab sebab kesalahannya. Proses melacak kesalahan ini dikenal dengan istilah mencari kutu (*debugging*). Hasil pelacaknya adalah didapatkan kutu tersebut (*bug yang berarti penyebab kesalahannya*).

Program dapat diuji untuk tiap-tiap modulnya dan dilanjutkan dengan pengujian untuk semua modul yang telah dirangkai. Dengan demikian terdapat tiga tingkat pengujian yang dilakukan, yaitu:

1. Pengujian modul

Pengujian untuk tiap-tiap modul program (dapat berupa program utama, sub routine, sub program) disebut dengan *stub testing*. Pengetesan suatu modul dapat saja dilakukan walaupun modul lainnya yang berhubungan dengannya belum ditulis. Hal ini mudah saja dilakukan, yaitu dengan cara mensimulasi modul yang dipanggil yang belum ditulis. Modul dipanggil yang disimulasi ini disebut dengan *stub*. Modul *stub* dapat berupa sub routine atau sub program yang tidak berisi dengan logika-logika program. Mungkin juga modul *stub* ini diisi dengan instruksi-instruksi yang akan mencetak parameter yang diterimanya untuk menunjukkan bahwa modul ini sudah dapat dipanggil dengan benar.

2. Pengujian Unit atau Pengujian Program

Setelah semua modul program selesai ditulis dan diuji secara independent sampai bebas dari kesalahan dan telah dirangkai menjadi satu unit program ini perlu diuji kembali. Pengetesan untuk ini disebut dengan *unit testing* atau *program testing* yang dimaksudkan untuk meyakinkan bahwa semua modul telah bekerja terintegrasi tanpa mengalami kesalahan.

3. Pengujian Sitem

Pengujian sistem biasanya dilakukan setelah pengujian program. Pengujian sistem dilakukan untuk memeriksa kekompakan antar komponen sistem yang diimplementasikan. Tujuan utama dari pengujian sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengujian perlu dilakukan untuk mencari kesalahan-kesalahan atau kelemahan-kelemahan yang mungkin terjadi. Pengujian sistem termasuk juga pengujian program secara menyeluruh. Pada pengujian program, masing-masing program

yang telah berjalan dengan benar dan baik bukan berarti tersebut juga akan dapat berjalan dengan program lainnya dalam sistem dengan baik. Kumpulan dari semua program yang telah diintegrasikan perlu diuji kembali untuk melihat apakah suatu program dapat menerima input data dengan baik, dapat memprosesnya dengan baik dan dapat memberikan output kepada program

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Hasil pengumpulan data dilakukan dengan cara wawancara dengan pakar yang berada di Dinas Pertanian Kabupaten Pohuwato yang beralamat di Kecamatan Marisa Kabupaten Pohuwato.

Matriks penyakit (hama) Kelapa Hibrida ter terdiri dari 6 macam jenis Hama yang ditunjukkan ole kode H001 sampai H06 yaitu terdiri dari Hama Brontispa Logissima, Kumbang Nyiur, Belalang Kembara, Hama Mayang, Ulat Api / Ulat Siput, Kumbang Sagu. Macam hama Tanaman ini ini terdiri dari 7 gejala penyakit yang ditunjukkan ole kode G01 sampai G7.

Hama yang terdiri dari 6 macam jenis hama dan 7 gejala penyakit akan diolah sehingga menghasilkan suatu kesimpulan dan solusi akhir dari masalah ini.

Tabel 4.1 : Datftar Macam Hama Kelapa

Kode Jenis Hama	Nama Macam Hama
H001	Brontispa Logissima
H002	Kumbang Nyiur
H003	Belalang Kembara
H004	Ulat Artona
H005	Ulat Api / Ulat Siput
H006	Kumbang Sagu

Tabel 4.2 : Daftar Gejala Penyakit Kelapa Hibrida

Kode	Nama Gejala Penyakit
G01	Busuk Pucuk
G02	Daun Pucuk layu
G03	Bercak Kelabu pada daun
G04	Bercak Coklat pada daun
G05	Daun Tinggal Lidi
G06	Daun Patah

G07	Gugur Buah
------------	------------

Tabel 4.3 : Daftar Solusi

Kode	Solusi
T001	mupukan , pengelolaan air dan sanitasi kebun untuk menunjang pertumbuhan tanaman yang baik sehingga produksinya maksimal.
T002	Menebang tanaman yang sudaah mati
T003	usahakan agar tanaman berada dalam kondisi baik seperti pemberian air yang cukup, pemupukan yang seimbang, daun sakit dipotong dan dibakar agar tidak menyebar, eradikasi / pemusnahan tanaman terserang.
T004	membersihkan kebun kelapa hibrida dari sisa-sisa batang kelapa yang sudah ditebang
T005	bersihkan mahkota daun kelapa supaya tidak jadi sarang tikus
T006	Singkirkan atau bersihkan sisa-sisa tumbuhan pada waktu persemian dan pembukaan tanah.
T007	membersihkan kebun

Tabel 4.4 : Matriks Penyakit Tanaman Kelapa Hibrida

H/G	H1	H2	H3	H4	H5	H6
G01	*	*				
G02	*	*				
G03					*	
G04					*	
G05			*		*	
G06		*		*		
G07	*			*		

Setiap gejala penyakit memiliki bobot, untuk nilai bobot dilihat dari seberapa besar tingkat kepentingan gejala terhadap penyakit, untuk bobot terdiri dari tiga parameter yaitu:

Tabel 4.5 : Tabel Bobot Parameter

Tingkat Gejala	Bobot/ Parameter
Gejala Penting	5
Gejala Sedang	3
Gejala Biasa	1

Tabel 4.6 : Daftar bobot dari setiap gejala penyakit

Kode	Nama Gejala Penyakit	Bobot
G01	Busuk Pucuk	5
G02	Daun Pucuk layu	5
G03	Bercak Kelabu pada daun	1
G04	Bercak Coklat pada daun	1
G05	Daun Tinggal Lidi	3
G06	Daun Patah	1
G07	Gugur Buah	5

4.2 Hasil Pemodelan Data

Berikut hasil pemodelan menggunakan metode CBR dari data yang telah didapatkan.

Tabel 4.7 : Kasus Lama

Hama	Gejala
H001	G001, G002, G007
H002	G001, G006, G002
H003	G005
H004	G007, G006
H005	G005, G003, G004
H006	G002

Tabel 4.8 : Kasus Baru

No	Kategori Gejala	Kode	Nama Gejala
1	Hama	G02	Daun Pucuk Layu
2	Hama	G03	Bercak Kelabu Pada Daun
3	Hama	G04	Bercak Coklat Pada Daun

Tabel 4.9 : Hasil Konsultasi

No	Kasus	Penyakit	Kecocokan(%)
1	Role Awal	Kumbang Sagu	100,00
2	Role Awal	Kumbang Nyiur	45,45
3	Role Awal	Ulat Api/Ulat Siput	40,00
4	Role Awal	Brontispa Logissima	33,33

Tabel 4.10 : Hasil Diagnosa

Hasil Diagnosa adalah Kumbang Sagu	
Solusi	
[T04]	Membersihkan kebun kelapa hibrida dari sisa-sisa batang kelapa yang sudah di tebang
[T07]	Pembersihan kebun

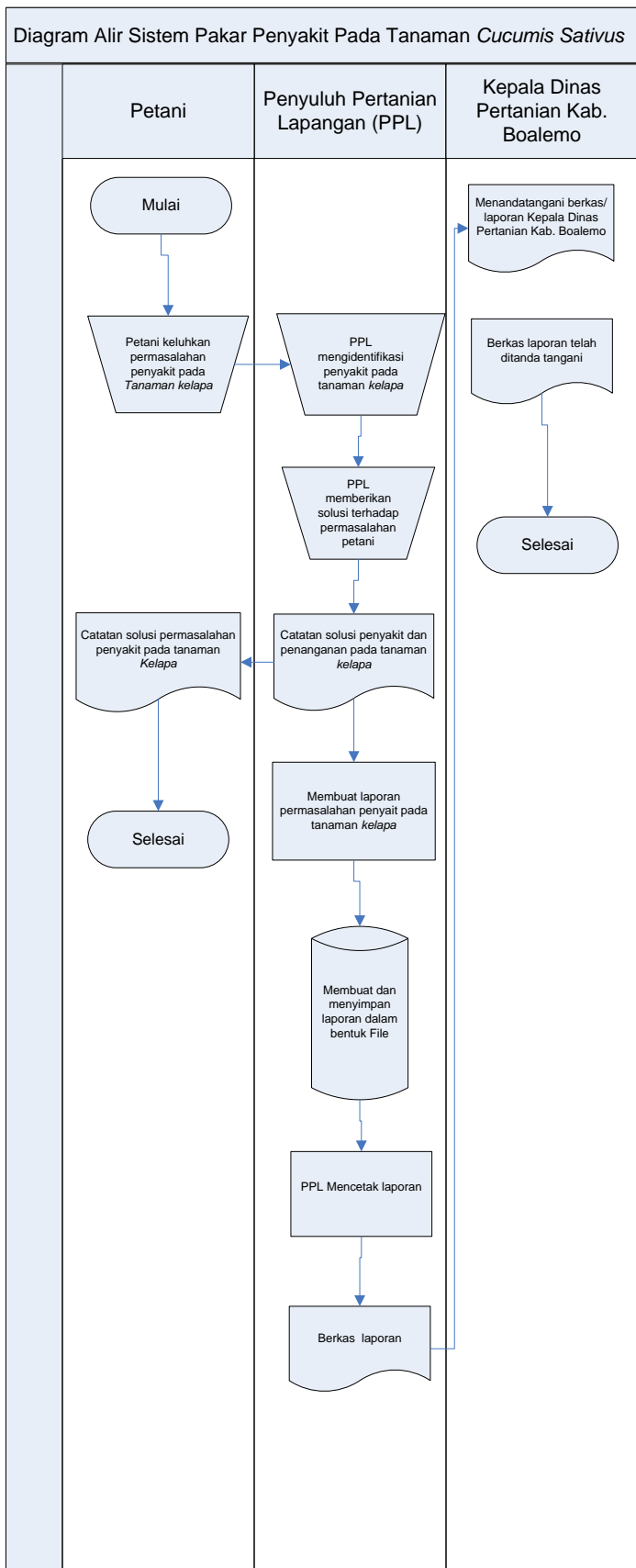
4.3 Hasil Pengembangan Sistem

4.3.1 Analisa Sistem

Analisa sistem adalah suatu teknik atau metode pemecahan masalah dengan cara menguraikan sistem ke dalam komponen-komponen pembentuknya untuk mengetahui bagaimana komponen-komponen tersebut bekerja dan saling berinteraksi satu sama lain untuk mencapai tujuan sistem

4.3.2 Analisis Sistem Berjalan

Analisa sistem berjalan adalah menganalisa sistem yang sedang berjalan/ sistem lama dalam diagnosa Penyakit Kelapa Hibrida.



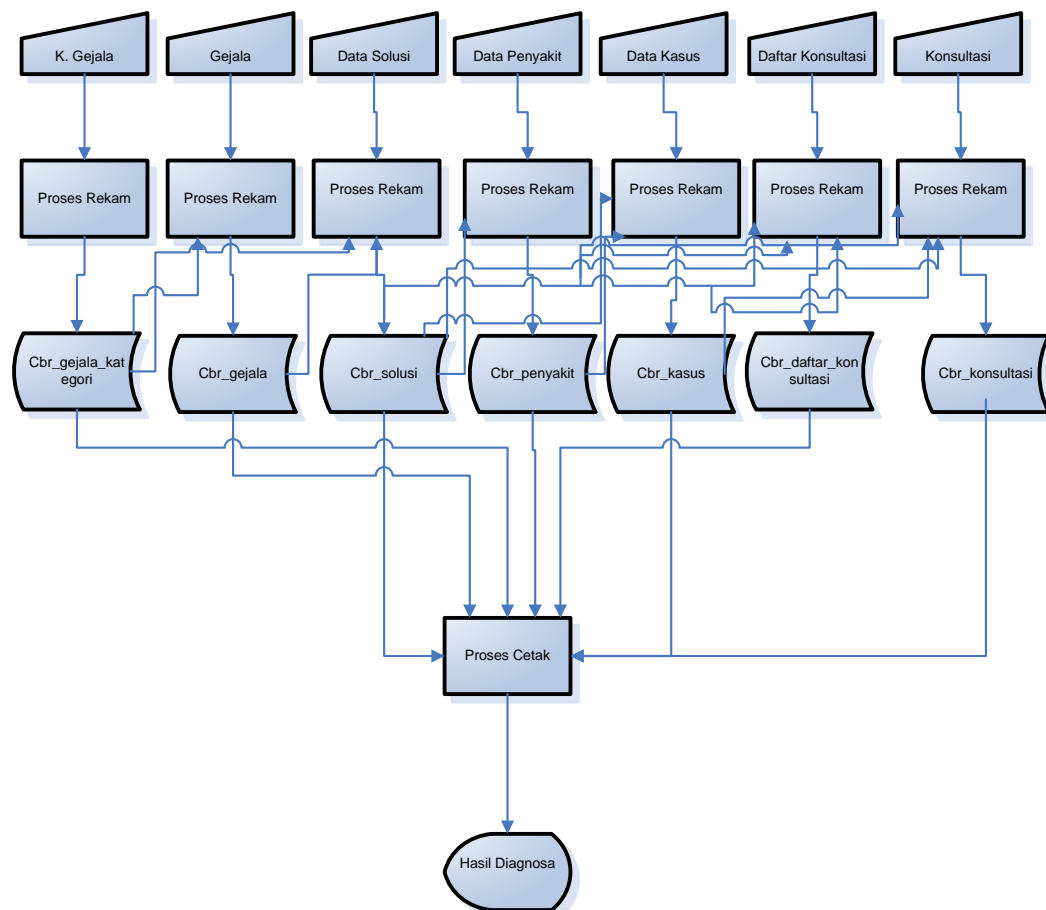
Gambar 4.1 : Sistem yang diusulkan

4.3.3 Analisa Sistem yang Diusulkan

Analisa sistem yang diusulkan adalah penguraian dari sistem utuh untuk setiap bagian komponen bertujuan untuk identifikasi dan evaluasi terhadap permasalahan setiap kesempatan dan hambatan yang ditimbulkan serta kebutuhan sehingga dapat di buat perbaikannya.

Setelah dilakukan analisa sistem yang berjalan/ sistem lama, tahapan selanjutnya ialah menganalisa sistem yang baru. Metode CBR serta penggunaan *Data Flow Diagram* (DFD) digunakan untuk menganalisa kebutuhan sistem. Data-data gejala penyakit yang dibutuhkan untuk pembuatan sistem ini dimasukkan kedalam analisa data sistem untuk mendiagnosa hama tanaman *Kelapa Hibrida*.

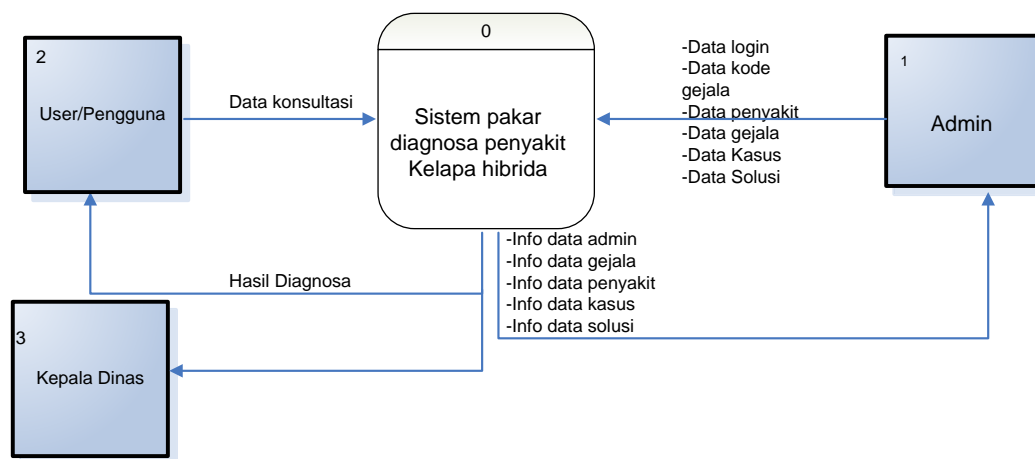
Setelah dilakukan analisa, diketahui sejauh mana kebutuhan sistem untuk diteliti darimana asal data, bagaimana aliran data ke sistem, bagaimana operasi sistem serta hasil akhirnya.



Gambar 4.2 : Analisis Sistem yang diusulkan

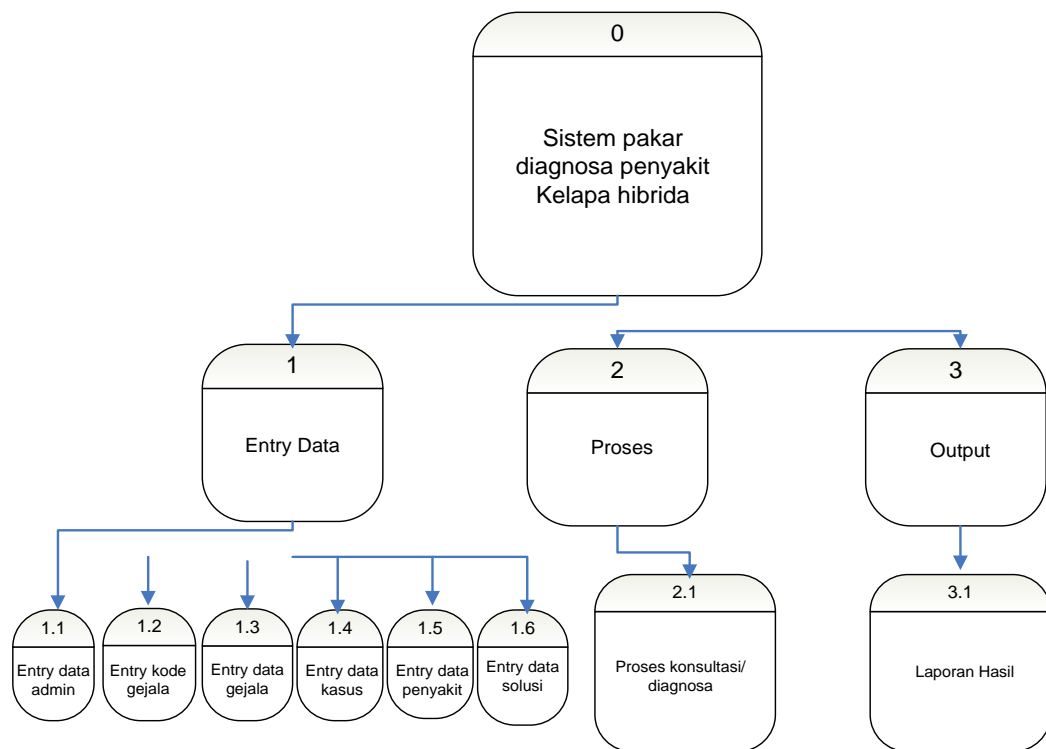
4.3.4 Diagram Konteks

Diagram konteks terdiri dari dua entitas yaitu Admin dan pengguna. Pengguna bisa langsung konsultasi tanpa melalui admin, tapi pengguna tidak bisa mengakses halaman admin. Sedangkan Admin menginput data-data penyakit, gejala, solusi serta basis pengetahuannya/ kasusnya yang telah didapatkan dari pakar. Sehingga nantinya akan mengeluarkan output kepada pengguna berupa hasil diagnosa.



Gambar 4.3 : Diagram Konteks

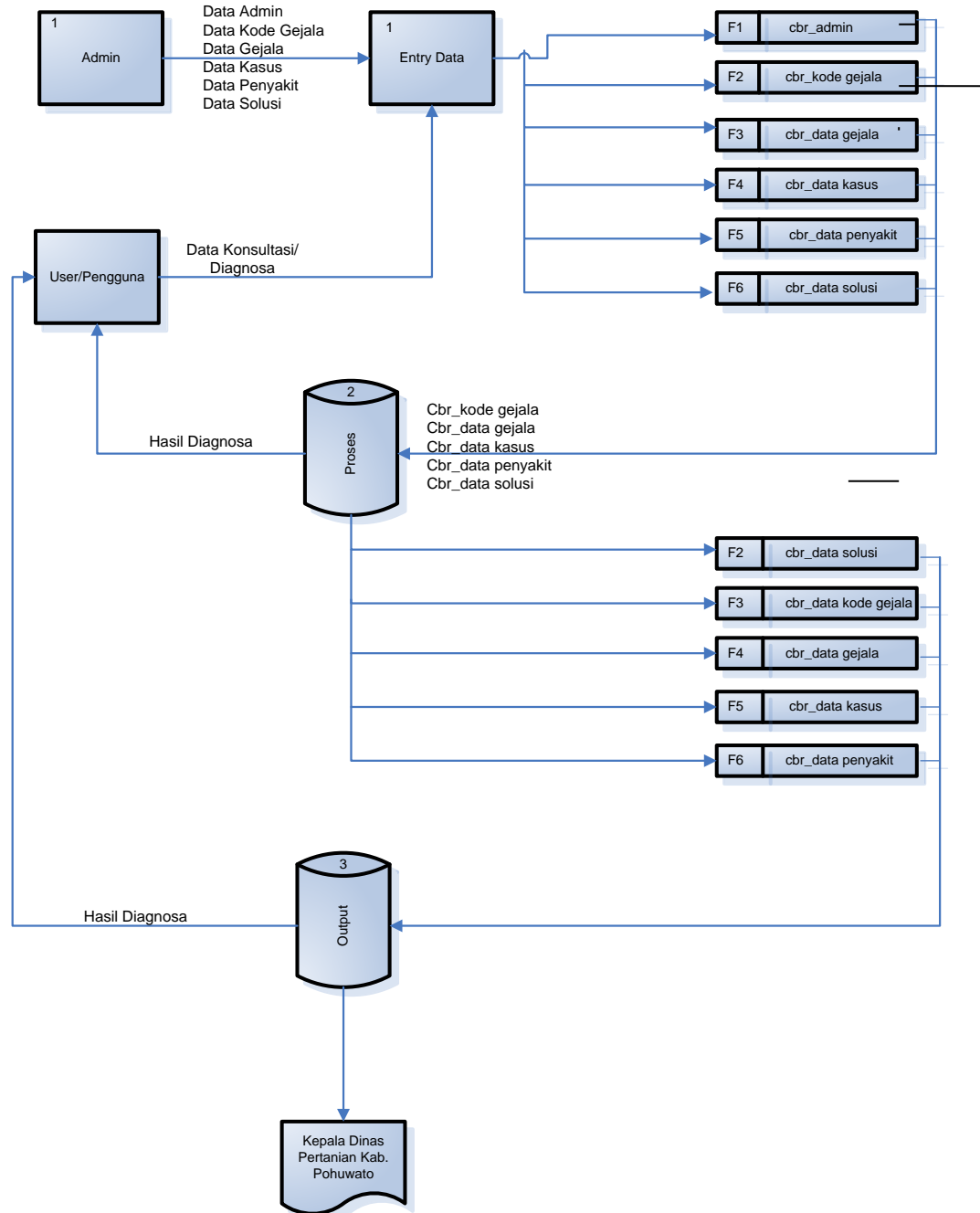
4.3.5 Diagram Berjenjang



Gambar 4.4 : Diagram Berjenjang

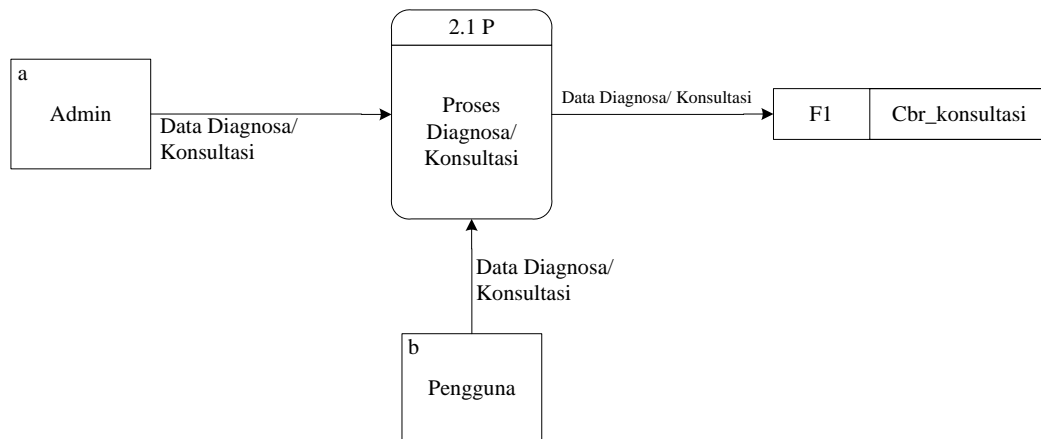
4.3.6 Diagram Arus Data (DAD)

4.3.6.1 Diagram Arus Data (DAD) Level 0



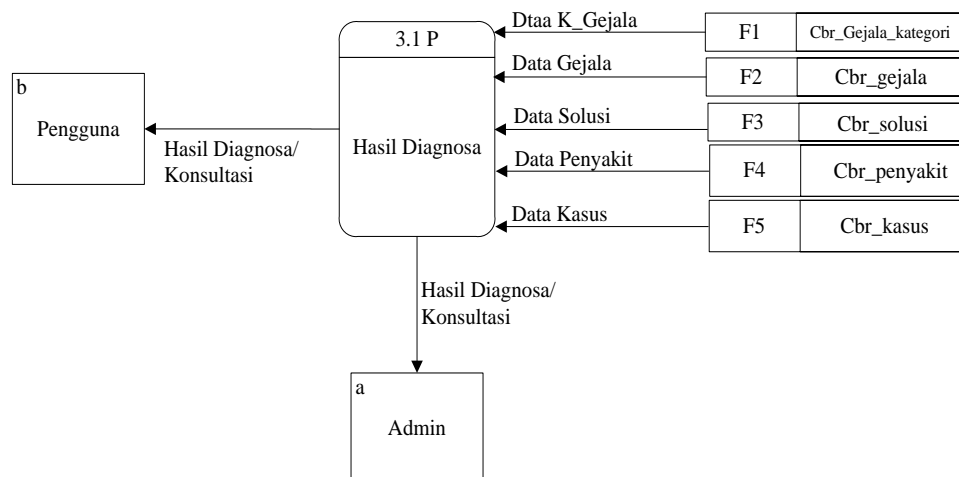
Gambar 4.5 : Diagram Arus Data Level 0

4.3.4.2 Diagram Arus Data Level 1 Proses 2



Gambar 4.6 : Diagram Arus Data Level 1 Proses 2

4.3.4.3 Diagram Arus Data Level 1 Proses 3



Gambar 4.7 : Diagram Arus Data Level 1 Proses 3

4.3.7 Kamus Data

Kamus data merupakan suatu penjelasan tertulis tentang suatu data yang berada di dalam database.

Tabel 4.11 : Kamus Data Pengguna

Nama Arus Data : Data Admin Penjelasan : Input Data Pengguna				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket

1	id_Admin	Int	11	id Admin
2	Nama	varchar	50	Admin
3	Username	varchar	20	Username Admin
4	Password	varchar	50	Password Admin
5	Tanggal	date		Tanggal

Tabel 4.12 : Kamus Data Gejala

Nama Arus Data : Data gejala Penjelasan : Input Data Gejala				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	id_gejala	int (primary)	11	Id gejala
2	kd_gejala	Varchar	5	Kode gejala
3	Nm_gejala	Varchar	250	Nama gejala
4	bobot_parameter	Int	1	Bobot parameter
5	id_gejala_kategori	Int	11	Id kategori gejala

Tabel 4.13 : Kamus Data Kategori Gejala

Nama Arus Data : Data Kategori Gejala Penjelasan : Input Data Alternatif				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	id_gejala_kategori	int	11	Id kategori Gejala
2	Gejala_kategori	Varchar	50	Nama Kategori Gejala
3	Keterangan	Text	50	Penjelasan gejala

Tabel 4.14 : Kamus Data Kasus

Nama Arus Data : Data Kasus Penjelasan : Kasus Antara Penyakit dengan gejala		Bentuk Data : Dokumen
--	--	-----------------------

No	Field Name	Type	Size	Ket
1	Id_kasus	Int	11	Id kasus
2	Nama	Varchar	50	Nama Kasus
3	Id_penyakit	Int	11	Id Penyakit
4	Tanggal	Date		Tanggal Inputan Kasus
5	Status	Int	1	Staus Kasus

Tabel 4.15 : Kamus Data Kasus Gejala

Nama Arus Data : Data Kasus Gejala Penjelasan : Kasus antara penyakit dengan gejala				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	Id_kasus_gejala	Int	11	Id kasus gejala
2	Id_kasus	Int	11	kasus
3	Id_gejala	Int	11	Nama gejala penyakit

Tabel 4.16 : Kamus Data Konsultasi

Nama Arus Data : Data Konsultasi Penjelasan : input data konsultasi				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	Id_konsultasi	Int	11	Id Konsultasi
2	Nama	Varchar	100	Nama petani
3	Tanggal	Datetime		Tanggal setiap konsultasi
4	Status	Int	1	Nilai setiap konsultasi

Tabel 4.17 : Kamus Data Konsultasi Gejala

Nama Arus Data : Data Konsultasi Gejala Penjelasan : Input data konsultasi gejala				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	Id_konsultasi_gejala	Bigint	20	Id Konsultasi gejala
2	Id_konsultasi	Int	11	Id Konsultasi
3	Id_gejala	Int	11	Id Gejala
4	Status	Int	1	Nilai setiap konsultasi

Tabel 4.18 : Kamus Data Konsultasi hasil

Nama Arus Data : Data Konsultasi Hasil Penjelasan : Hasil konsultasi setiap kasus penyakit				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	Id_konsultasi_hasil	Bigint	20	Id Konsultasi hasil
2	Id_konsultasi	Int	11	Id Konsultasi
3	Id_kasus	Int	11	Id Kasus
4	Id_penyakit	Int	11	Id Penyakit
5	Nilai	Double	11	Nilai dari setiap konsultasi
6	Status	Int	1	Nilai untuk setiap konsultasi hasil

Tabel 4.19 : Kamus Data Penyakit

Nama Arus Data : Data Penyakit Penjelasan : input data penyakit				Bentuk Data : Dokumen
--	--	--	--	-----------------------

No	Field Name	Type	Size	Ket
1	Id_penyakit	Int	11	Id Penyakit
2	Kd_penyakit	Varchar	5	Kode penyakit
3	Nm_penyakit	Varchar	50	Nama penyakit
4	Definisi	Text	11	Penjelasan tentang penyakit

Tabel 4.20 : Kamus Data Penyakit Solusi

Nama Arus Data : Data Penyakit Solusi Penjelasan : input data penyakit dan solusi				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	Id_penyakit_solusi	Int	11	Nama penyakit dan solusi
2	Id_penyakit	Int	11	Nama penyakit
3	Id_solusi	Int	11	Solusi untuk penyakit

Tabel 4.21 : Kamus Data Solusi

Nama Arus Data : Data Solusi Penjelasan : Input Data Solusi				Bentuk Data : Dokumen
No	Field Name	Type	Size	Ket
1	id_solusi	Int	11	Id Solusi
2	kd_solusi	Varchar	5	Kode solusi
3	Nm_solusi	Varchar	100	Nama Solusi
4	keterangan	Text		Keterangan Solusi

4.3.8 Arsitektur Sistem/ Kebutuhan Hardware dan Software

Dalam pengembangan sistem ini penulis menggunakan bahasa pemrograman PHP dan database MySQL. Untuk implementasi sistem membutuhkan perangkat lunak

atau perangkat keras dasar yang mesti digunakan untuk menjalankan program aplikasi, diantaranya :

1. Spesifikasi *Hardware dan Software*

Spesifikasi yang disarankan untuk komputer

- a. Processor setara AMD A4-3305M APU with Radeon(tm) HD Graphics 1.90 GHz atau lebih
- b. RAM (Memory) 2 GB atau lebih
- c. HDD 500 atau lebih.
- d. Monitor SVGA dengan Resolusi 1024 X 768
- e. Dan Peralatan I/O Lainnya
- f. Windows XP, Vista Windows 8 atau lebih
- g. Browser Mozilla Firefox, Internet Explorer dan Google Chrome untuk membuka Web
- h. Dreamwaver CS4 atau lebih

4.3.9 Interface Design

4.3.10 Desain Secara Umum

4.3.10.1 Desain Output Secara Umum

Untuk : Sistem Pakar Penyakit Kelapa Hibrida

Sistem : Penerapan Metode CBR untuk diagnose penyakit kelapa hibrida

Tahap : Desain Output Secara Umum

Tabel 4.22 : Desain Output Secara Umum

No	Nama	Tipe	Format	Media	Alat	Distribusi	Periode
1	Daftra Gejala yang dipilih	Internal/ Eksternal	Tabel	Layar	Monitor	Pengguna, admin	Non Periodik
2	Hasil Konsultasi	Internal/ Eksternal	Tabel	Layar	Monitor	Pengguna, admin	Non Periodik

4.3.10.2 Desain Input Secara Umum

Tujuan dari Desain Input secara umum itu sendiri yaitu untuk memberikan gam-

baran secara umum kepada user tentang sistem yang baru/ sistem yang diusulkan. Desain sistem secara umum mengidentifikaikan komponen-komponen sistem informasi yang akan didesain secara rinci. Desain terinci dimaksudkan untuk pemrograman komputer dan ahli yang mengimplementasikan sistem.

Untuk : Sistem Pakar Penyakit Kelapa Hibrida

Sistem : Penerapan Metode CBR Untuk Diagnosa Penyakit Kelapa Hibrida

Tahap : Desain Output Secara Umum

Tabel 4.23 : Desain Input Secara Umum

No	Nama	Tipe	Format	Media	Alat	Distribusi	Periode
1	Data Admin	Internal	Tabel	Layar	Monitor	Admin	Non periodik
2	Data Kategori Gejala	Internal	Tabel	Layar	Monitor	Admin	Non periodik
3	Data Gejala	Internal	Tabel	Layar	Monitor	Admin	Non periodik
4	Data Solusi	Internal	Tabel	Layar	Monitor	Admin	Non periodik
5	Data Penyakit	Internal	Tabel	Layar	Monitor	Admin	Non periodik
6	Data Kasus	Internal	Tabel	Layar	Monitor	Admin	Non periodik

4.3.11 Desain Secara Terinci

4.3.9.2.1 Desain Output Secara Terinci

Gambar berikut adalah desain hasil analisa yang dirancang untuk menampilkan data pengguna, hasil identifikasi baik data penyakit, gejala maupun pengobatan dari penyakit Kelapa hibrida.

Konsultasi- Hasil Diagnosis

Gejala yang dipilih

Hasil Konsultasi

>

x Selesai Konsultasi

Hasil diagnose anda adalah

Penjelasan Penyakit ...

Solusi :

.....

4.3.9.2.2 Desain Input Secara Terinci

a) Input Kategori Gejala

Gambar dibawah ini merupakan desain input untuk Kategori Gejala

Tambah Kategori Gejala

Kategori Gejala	
Keterangan	

Simpan

Batal

Gambar 4.9 : Desain Input Penyakit

b) Input Gejala

Gambar berikut adalah desain input gejala yang dirancang untuk menginput data gejala penyakit Kelapa hibrida ke tabel gejala dalam database.

Tambah Gejala

Kategori Gejala	
Kode Gejala	Ex. G01
Nama Gejala	
Bobot Parameter	Gejala Penting (5)

Simpan

Batal

Gambar 4.10 : Desain Input Gejala

c) Input Solusi

Gambar berikut adalah desain yang dirancang untuk menginput data Solusi untuk setiap penyakit.

Tambah Solusi

Kode Solusi	
Nama Solusi	
Keterangan	

Simpan Batal

Gambar 4.11 : Desain Input Solusi

d) Input Penyakit

Gambar berikut adalah desain input penyakit, dimana penyakit yang diinput yaitu penyakit tentang Kelapa hibrida.

Tambah Kasus

Nama Kasus	
Pilih Penyakit	Pilih
Status	Aktif

Simpan Batal

Gambar 4.12 : Desain Input Penyakit

e) **Input Kasus**

Gambar berikut yaitu gambar desain input Kasus, disini penginputan kasus penyakit.

Gambar 4.13 : Desain Input Kasus

4.3.9.2.3 Desain Database Secara Terinci

Tabel 4.24 : Tabel Penyakit

No.	Nama Field	Tipe	Ukuran	Kunci
1.	Id_penyakit	Int	11	Primary Key
2.	Kd_penyakit	Varchar	5	
3.	Nama_penyakit	Varchar	50	
4.	definisi	Text		

Tabel 4.25 : Tabel Gejala

No.	Nama Field	Tipe	Ukuran	Kunci
1.	Id_gejala	Char	11	Primary Key
2.	Kd_gejala	Varchar	5	
3	Nm_gejala	Varchar	250	

4	Bobot_parameter	Int	1	
5	Id_gejala_kategori	Int	11	

Tabel 4.26 : Tabel Kasus_Gejala

No.	Nama Field	Tipe	Ukuran	Kunci
1.	Id_kasus_gejala	Int	11	Primary Key
2.	Id_kasus	Int	11	
3.	Id_gejala	Int	11	

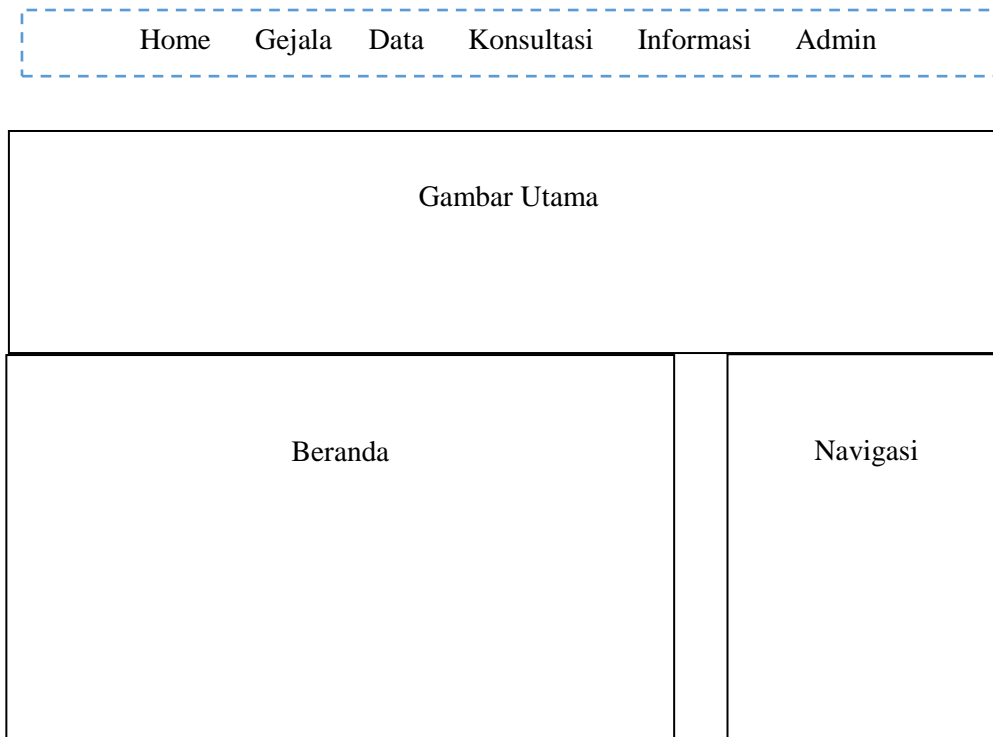
Tabel 4.27 : Tabel konsultasi

No.	Nama Field	Tipe	Ukuran	Kunci
1.	Id_knsultasi	integer	11	Primary Key
2.	Nama	Varchar	100	
4.	tanggal	Datetime		
5	status	Int	1	

Tabel 4.28 : Tabel Konsultasi Hasil

No.	Nama Field	Tipe	Ukuran	Kunci
1.	Id_konsultasi_hasil	Bigint	20	Primary Key
2.	Id_konsultasi	Int	11	
3.	Id_kasus	Int	11	
4	Id_penyakit	Int	11	
5	Nilai	Double		
6.	status	Int	int	

4.3.9.2.4 Desain Menu Utama



Gambar 4.14 : Desain menu Utama

4.3.12 Hasil Kontruksi Sistem

Pada tahap (81)

1. Bahasa Pemrograman yang digunakan yaitu PHP
2. Xampp digunakan sebagai Web Server
3. Dreamwaver digunakan sebagai Halaman xxxxx

4.3.13 Pengujian Sistem

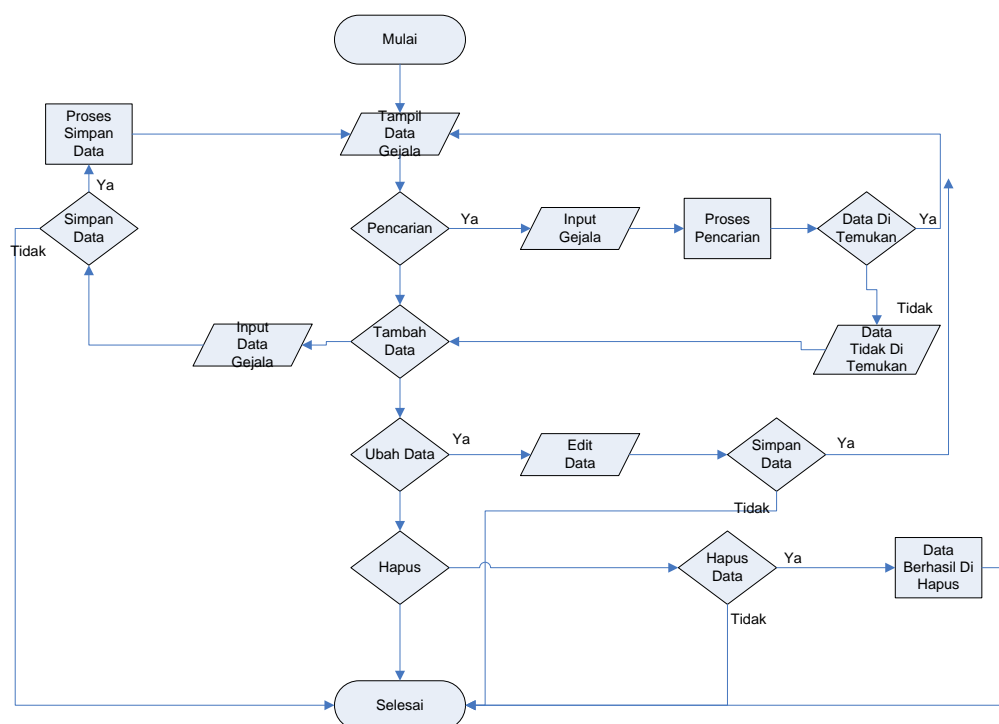
4.3.13.1 Pengujian White Box

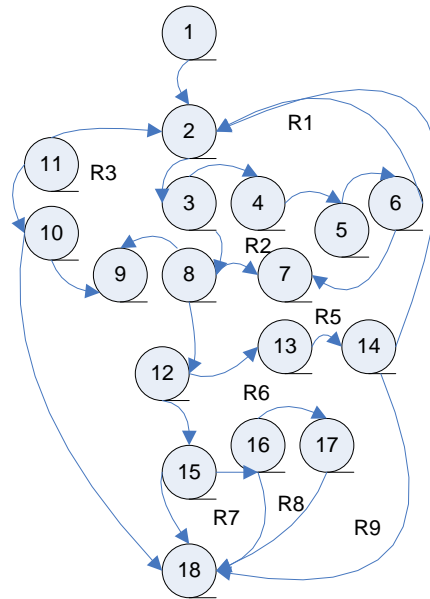
Tujuan penggunaan *White Box* untuk menguji semua statement program. Penggunaan metode pengujian *White Box* dilakukan untuk:

- Memberikan jaminan bahwa semua jalur independent suatu modul digunakan minimal satu kali.
- Menggunakan semua keputusan logis untuk semua kondisi *true* atau *false*
- Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
- Menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

Berikut pengujian *White Box* menggunakan *Flowchart* dan *Flowgraph*. Peneliti menggunakan *Flowchart* Alternatif.

a. *Flowchart* Gejala



Gambar 4.15 : Flowchart Gejala**b. Flowgraph Gejala****Gambar 4.16 : Flowgraph Gejala**

Dari *flowgraph* di atas pada gambar 5.5, didapatkan

- *Region (R)* = 9
- *Node (N)* = 18
- *Edge (E)* = 25
- *Predicate Node (P)* = 7

Dari *flow graph* diatas, cyclomatic complexity dari sebuah program dapat dibuat dengan menggunakan rumus dibawah ini :

$V(G)$: *cyclomatic complexity*

E : total jumlah *edge*

N : Total jumlah *node*

$$V(G) = E - N + 2$$

Pada *Flow graph* diatas (gambar 5.3), dapat dihitung *cyclomatic complexity* nya sebagai berikut :

$$\begin{aligned} V(G) &= 25 \text{ Edge} - 18 \text{ Node} + 2 \\ &= 8 \end{aligned}$$

Atau

$$\begin{aligned} V(G) &= 8 \text{ Predicate} + 1 \\ &= 9 \end{aligned}$$

Angka 5 dari hasil perhitungan *cyclomatic complexity* menunjukkan jumlah *independent path* dari *basis path testing*, atau dengan kata lain menunjukkan jumlah pengujian yang harus dijalankan untuk memastikan semua statement pada program dijalankan minimal sekali (semua *statement* telah diuji)

Hasil independent path pada contoh diatas dapat dijabarkan sebagai berikut:

R 1 : 1-2-3-4-5-6-2

R 2 : 1-2-3-4-5-7-8

R 3 : 1-2-3-8-9-10-11-2

R 4 : 1-2-3-8-9-10-18

R 5 : 1-2-3-8-12-13-14-2

R 6 : 1-2-3-8-12-13-14-18

R 7 : 1-2-3-8-12-15-16-18

R 8 : 1-2-3-8-12-15-16-17-8

R 9: 1-2-3-8-12-15-18

Catatan :

- Independent path adalah setiap *path* yang dilalui program yang menunjukkan satu set baru dari pemrosesan statement atau dari sebuah kondisi baru.
- *Independent path* pada *flow graph* harus melewati sedikitnya satu *edge* yang belum pernah dilewati oleh *path* sebelumnya.
- *Independent path* selalu dimulai dari *node* awal hingga ke *node* akhir

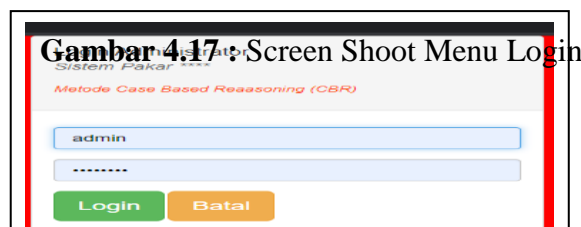
Independent path yang dibuat pertama kali adalah *independent path* terpendek.

4.3.13.2 Pengujian *Black Box*

Pengujian *Black Box* merupakan pendekatan komplementer dari teknik *White Box*, Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Dibawah ini merupakan pelaksanaan pengujian dengan menggunakan metode *black box* dari perangkat lunak yang dibuat.

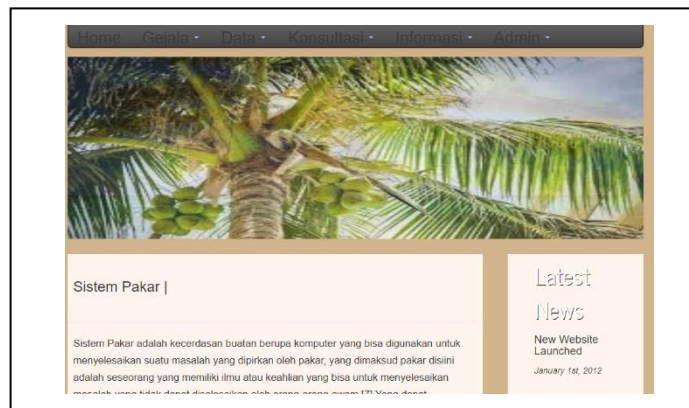
1. Menampilkan menu login

Test	Hasil	Keterangan
Mampu menampilkan menu Login	√	Berhasil menampilkan menu login



2. Pengujian Untuk Menampilkan Halaman Utama

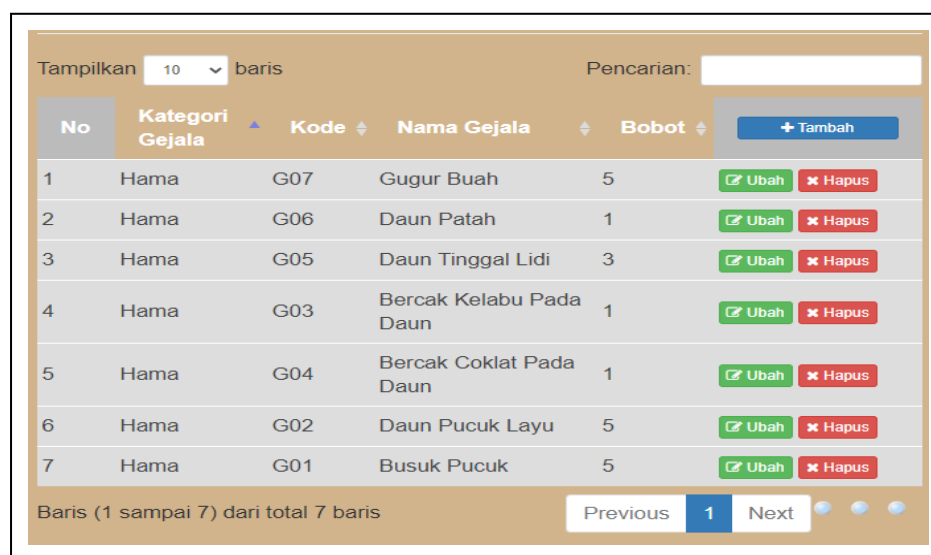
Test	Hasil	Keterangan
Mampu menampilkan Halaman Utama	√	Berhasil menampilkan Halaman Utama



Gambar 4.18 : Screen Shoot Halaman Utama

3. Pengujian untuk menampilkan Gejala

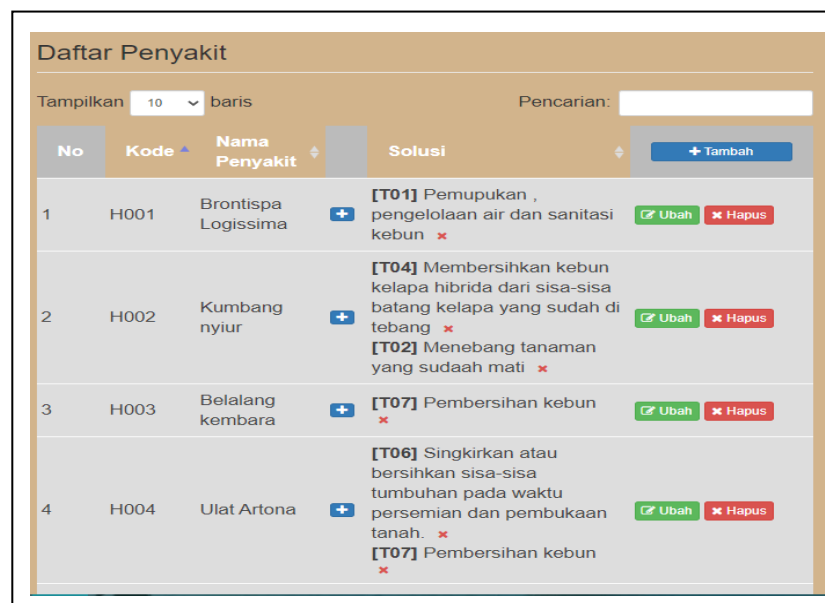
Test Faktor	Hasil	Keterangan
Mampu menampilkan Data Gejala	√	Berhasil menampilkan Halaman Gejala



Gambar 4.19 : Screen Shoot Data Gejala

4. Pengujian untuk menampilkan Data Penyakit

Test Faktor	Hasil	Keterangan
Mampu menampilkan Data Penyakit	√	Berhasil menampilkan Halaman Data Penyakit



Daftar Penyakit

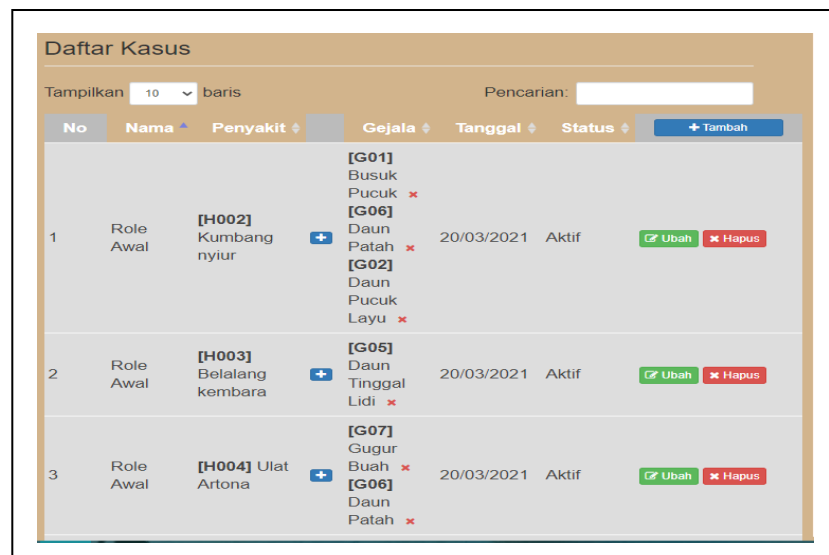
Tampilkan baris Pencarian:

No	Kode	Nama Penyakit	Solusi	
1	H001	Brontispa Logissima	[T01] Pemupukan , pengelolaan air dan sanitasi kebun ✖	<input checked="" type="checkbox"/> Ubah ✖ Hapus
2	H002	Kumbang nyiur	[T04] Membersihkan kebun kelapa hibrida dari sisa-sisa batang kelapa yang sudah di tebang ✖ [T02] Menebang tanaman yang sudaah mati ✖	<input checked="" type="checkbox"/> Ubah ✖ Hapus
3	H003	Belalang kembara	[T07] Pembersihan kebun ✖	<input checked="" type="checkbox"/> Ubah ✖ Hapus
4	H004	Ulat Artona	[T06] Singkirkan atau bersihkan sisa-sisa tumbuhan pada waktu perseminan dan pembukaan tanah. ✖ [T07] Pembersihan kebun ✖	<input checked="" type="checkbox"/> Ubah ✖ Hapus

Gambar 4.20 : Screen Shoot Data penyakit

5. Pengujian untuk menampilkan Data Kasus

Test Faktor	Hasil	Keterangan
Mampu menampilkan Data Kasus	√	Berhasil menampilkan Halaman Data Kasus



No	Nama	Penyakit	Gejala	Tanggal	Status	
1	Role Awal	[H002] Kumbang nyiur	[G01] Busuk [G06] Daun Patah [G02] Daun Pucuk Layu	20/03/2021	Aktif	Ubah Hapus
2	Role Awal	[H003] Belalang kembara	[G05] Daun Tinggal Lidi	20/03/2021	Aktif	Ubah Hapus
3	Role Awal	[H004] Ulat Artona	[G07] Gugur Buah [G06] Daun Patah	20/03/2021	Aktif	Ubah Hapus

Gambar 4.21 : Screen Shoot Data Kasus

Dari hasil pengujian dapat disimpulkan untuk uji *black box* yang meliputi uji input proses dan output dengan acuan rancangan perangkat lunak telah terpenuhi dengan hasil sesuai dengan rancangan. Uji juga dilakukan pada program utama dan program pendukung lainnya.

BAB V

PEMBAHASAN PENELITIAN

5.1 Pembahasan Model

tahap pembahasan model, disini akan diuraikan perhitungan dari metode yang digunakan dengan menggunakan data yang telah ada.

Rumus :

$$Similarity(p,q) = \frac{S_1 \times W_1 + S_2 \times W_2 + \dots + S_n \times W_n}{W_1 + W_2 + \dots + W_n}$$

Kode Jenis Hama	Nama Macam Hama
H001	Brontispa Logissima
H002	Kumbang Nyiur
H003	Belalang Kembara
H004	Ulat Artona
H005	Ulat Api / Ulat Siput
H006	Kumbang Sagu

Kasus I (H001) = Brontispa Logissima

Kasus Lama

Gejala	Bobot
G01	5
G02	5
G07	5

Konsultasi

Gejala	Bobot
G002	5
G003	1
G004	1

Kemiripan

Kasus Lama	Konsultasi	Kasus Baru
G01		0
G02	02	1

	03	0
	04	0
G07		0

$$\text{Similirity } (p,q) = \frac{(0 * 5) + (1 * 5) + (0*1) + (0*1) + (0*5)}{5+5+5}$$

$$= \frac{5}{15}$$

$$= 0,33333$$

$$\text{Dijadikan } \% = 0,3333 * 100 = 33,33\%$$

Kasus 2 (H002) = Kumbang Nyiur

Kasus Lama

Gejala	Bobot
G01	5
G02	5
G06	1

Konsultasi

Gejala	Bobot
G002	5
G003	1
G004	1

Kemiripan

sus Lama	nsultasi	sus Baru
01		0
02	02	1
	03	0
	04	0
06		0

$$\text{Similirity } (p,q) = \frac{(0 * 5) + (1 * 5) + (0*1) + (0*1) + (0*1)}{5+5+1}$$

$$= \frac{5}{11}$$

$$= 0,45$$

$$\text{Dijadikan } \% = 0,45 * 100 = 45,45\%$$

Kasus 3 (H005) = Ulat Api / Ulat Siput

Kasus Lama

Gejala	Bobot
G03	1
G04	1
G05	3

Konsultasi

Gejala	Bobot
G02	5
G03	1
G04	1

Kemiripan

Kasus Lama	Konsultasi	Ksus Baru
	G02	
G03	G03	1
G04	G04	1
G05		0

$$\begin{aligned}
 \text{Similarity } (p,q) &= \frac{(0*5) + (1*1) + (1*1) + (0*3)}{1+1+3} \\
 &= \frac{2}{5} \\
 &= 0,4 \\
 \text{Dijadikan } \% &= 0,4 * 100 = 40\%
 \end{aligned}$$

Kasus 4 (H006) = Kumbang Sagu

Kasus Lama

Gejala	Bobot
G02	5

Konsultasi

Gejala	Bobot
G02	5
G03	1
G04	1

Kemiripan

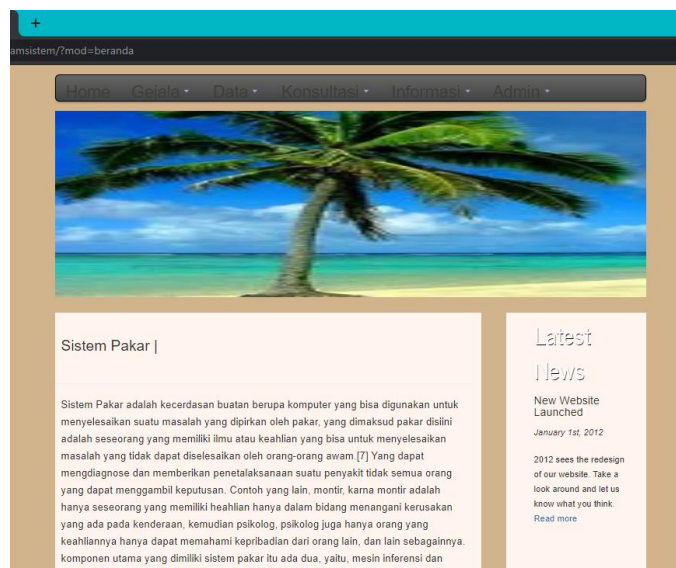
Kasus Lama	Konsultasi	Kasus Baru
02	02	1
	03	0
	04	0

$$\begin{aligned}
 \text{Similarity } (p,q) &= \frac{(1*5) + (0*1) + (0*1)}{5} \\
 &= \frac{5}{5} \\
 &= 0,1 \\
 \text{Dijadikan } \% &= 0,1 * 100 = 100\%
 \end{aligned}$$

5.2 Pembahasan Sistem

Pada tahap ini dilakukan penerapan hasil perancangan antarmuka ke dalam sistem yang dibangun dengan menggunakan perangkat lunak yang telah di paparkan sub bab implementasi perangkat lunak.

5.2.1 Halaman Menu Utama

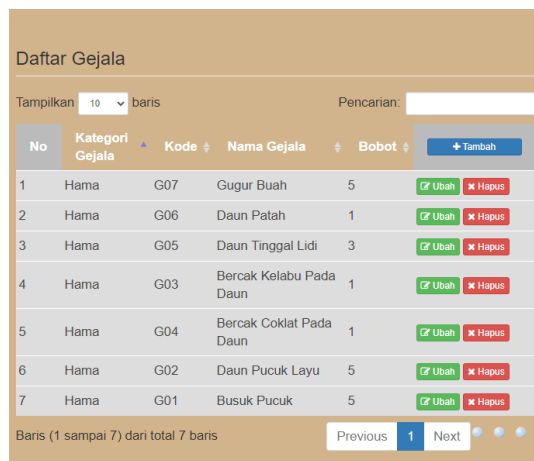


Gambar 5.1 : Halaman menu utama

Pada tampilan menu utama terdapat 6 menu yang digunakan pada system, yaitu menu Home, Gejala, Data, Konsultasi, Informasi dan Admin. Sedangkan pada tampilan dibawahnya terdapat link untuk mempermudah/ cara cepat untuk menampilkan gejala,

penyakit, Solusi, data kasus dan sebagainya.

5.3.2 Halaman Menu Gejala

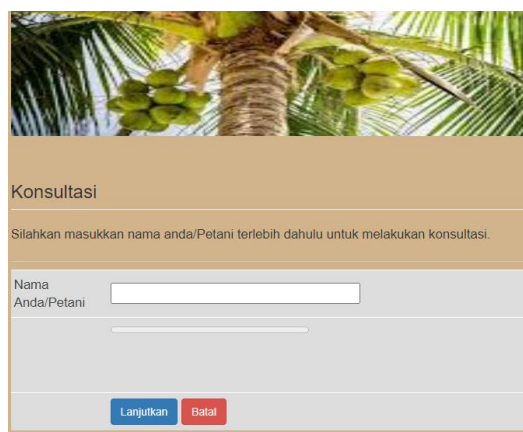


No	Kategori Gejala	Kode	Nama Gejala	Bobot	
1	Hama	G07	Gugur Buah	5	Ubah Hapus
2	Hama	G06	Daun Patah	1	Ubah Hapus
3	Hama	G05	Daun Tinggal Lidi	3	Ubah Hapus
4	Hama	G03	Bercak Kelabu Pada Daun	1	Ubah Hapus
5	Hama	G04	Bercak Coklat Pada Daun	1	Ubah Hapus
6	Hama	G02	Daun Pucuk Layu	5	Ubah Hapus
7	Hama	G01	Busuk Pucuk	5	Ubah Hapus

Gambar 5.2 : Halaman menu Gejala

Pada tampilan menu Gejala, digunakan untuk melihat data gejala-gejala pada tanaman kelapa hibrida. Pada Form ini juga digunakan untuk menginput/ menambah, mengubah dan menghapus gejala.

5.3.3 Halaman Menu Konsultasi



Gambar 5.3 : Halaman Menu Konsultasi

Pada tampilan menu konsultasi, pada tampilan di atas merupakan form untuk daftar konsultasi, maksudnya setiap user yang konsultasi datanya yang disimpan bisa dilihat di halaman ini, dihalaman ini juga admin bisa melihat detail gejala-gejala dan penyakit yang dialami tanaman kelapa hibrida user, dihalaman ini juga admin bisa menghapus data user yang telah konsultasikan tanaman kelapa hibridanya.

5.3.4 Halaman Menu Data Penyakit

Daftar Penyakit

Tampilkan 10 baris Pencarian:

No	Kode	Nama Penyakit	Solusi	
1	H001	Busuk Pucuk	[T01] Pemupukan , pengelolaan air dan sanitasi kebun	<input checked="" type="checkbox"/> Ubah <input checked="" type="checkbox"/> Hapus
2	H002	Daun Pucuk Layu	[T04] Membersihkan kebun kelapa hibrida dari sisa-sisa batang kelapa yang sudah di tebang [T02] Menebang tanaman yang sudah mati	<input checked="" type="checkbox"/> Ubah <input checked="" type="checkbox"/> Hapus
3	H003	Bercak Kelabu Pada Daun	[T07] Pembersihan kebun	<input checked="" type="checkbox"/> Ubah <input checked="" type="checkbox"/> Hapus
4	H004	Bercak Coklat Pada Daun	[T06] Singkirkan atau bersihkan sisa-sisa tumbuhan pada waktu persemanian dan pembukaan tanah [T07] Pembersihan kebun	<input checked="" type="checkbox"/> Ubah <input checked="" type="checkbox"/> Hapus

Gambar 5.4 : Halaman menu Data penyakit

Pada halaman ini admin bisa melihat data-data gejala yang sering terjadi pada tanaman kelapa hibrida. Di halaman ini admin juga bisa menambah, mengubah/ mengedit dan menghapus penyakit.

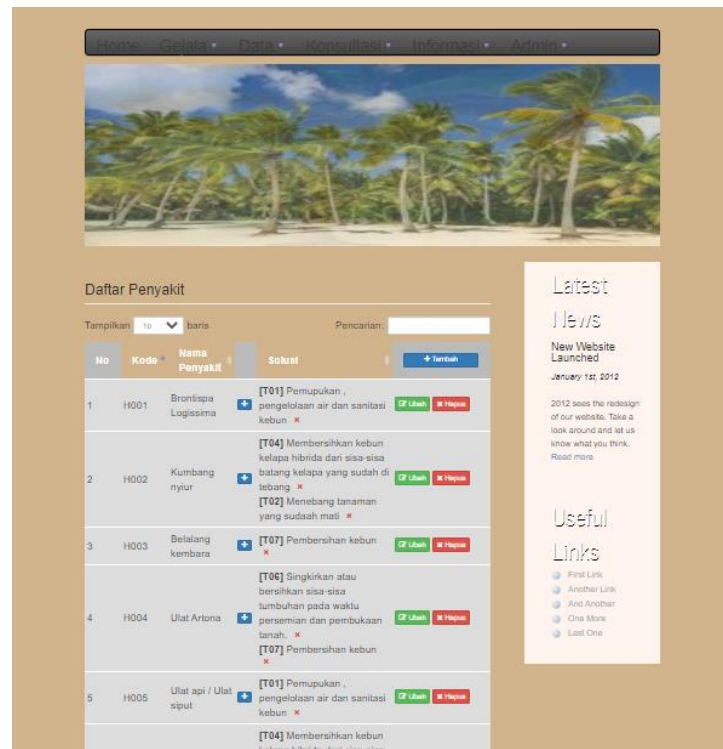
5.3.5 Halaman Menu Data Kasus

Daftar Kasus						
Tampilkan 10 baris		Pencarian: <input type="text"/>				
No	Nama	Penyakit	Gejala	Tanggal	Status	+ Tambah
1	Role Awal	[H002] Kumbang nyiur	[G01] Busuk Pucuk ✖ [G06] Daun Patah ✖ [G02] Daun Pucuk Layu ✖	20/03/2021	Aktif	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>
2	Role Awal	[H003] Belalang kembara	[G05] Daun Tinggal Lidi ✖	20/03/2021	Aktif	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>
3	Role Awal	[H004] Ulat Artona	[G07] Gugur Buah ✖ [G06] Daun	20/03/2021	Aktif	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>

Gambar 5.5 : Halaman menu data kasus

Pada halaman ini, admin bisa menambah solusi untuk setiap penyakit. Jadi solusi yang dimasukkan/ yang ditambahkan nantinya akan muncul pada hasil konsultasi tergantung dari Penyakit yang dialami tanaman kelapa hibrida. Pada halaman ini juga admin bisa mengubah dan menghapus data.

5.3.6 Halaman Menu Informasi Daftar Penyakit



Gambar 5.6 : Halaman menu informasi daftar penyakit

Pada halaman menu informasi daftar penyakit, pada halaman ini admin bisa melihat detail setiap penyakit beserta solusi yang akan ditampilkan.

BAB VI

PENUTUP PENELITIAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dan pembahasan yang telah diuraikan sebelumnya, maka dapat ditarik suatu kesimpulan bahwa :

1. Dapat diketahui cara merekayasa sistem pakar Diagona Penyakit Tanaman Kelapa Hibrida
2. Sistem pakar ini dapat di implementasikan untuk Mendiagnosa Penyakit Tanaman Kelapa Hibrida. Hal ini dapat dibuktikan dengan hasil pengujian yang dilakukan dengan metode *White Box Testing* dan *Bases Path Testing* yang menghasilkan nilai *Cyclomatic Complexity* = 8.

Sistem pakar ini dapat di implementasikan untuk Mendiagnosa Penyakit Tanaman Kelapa Hibrida. Hal ini dapat dibuktikan dengan hasil pengujian yang dilakukan dengan metode *White Box Testing* dan *Bases Path Testing* yang menghasilkan nilai *Cyclomatic Complexity* = 8

6.2 Saran

Penulis ingin menyampaikan beberapa saran atau masukan bagi siapapun yang ingin mengembangkan atau menyempurnakan dari Sistem Pakar ini adalah sebagai berikut :

1. Sistem pakar ini masih belum memberikan keakuratan data sebab metode perhitungan ketidak pastian yang digunakan hanya satu.
2. Gunakan metode lain untuk menjadi bahan perbandingan

DAFTAR PUSTAKA

- [1] AN Rahmi, I Verawati, M Kurniasih - INTECHNO Journal-Information Technology ..., 2019. Sistem Pakar Diagnosa Penyakit dan Hama Pada Tanaman Kelapa Hibrida Menggunakan Metode Forward Chaining
- [2] D. Kurniawan (2016). *sistem pakar untuk mendiagnosa penyakit tanaman Kelapa Hibrida dengan metode forward chaining*
- [3] L Pasaribu – Pelita (2019) Sistem Pakar Mendiagnosa Hama Penyakit Tanaman Kelapa Hibrida Metode Naïve Bayes. L ejurnal.stmik-budidarma.ac.id
- [4] Oktoviana Yanti, et all, 2012. *Pengaruh Lama Penyimpanan Dan Konsentrasi Natrium Benzoat Terhadap Kadar Vitamin C Kelapa Hibrida (Capsicum Annuum L)*. Jurnal Akademika Kimia Volume 1, No.4, 2012 : 193
- [5] Nurahmi E., T. Mahmud, dan Sylvia R.S . 2011. *Efektivitas Pupuk Organik terhadap Pertumbuhan dan Hasil Kelapa Hibrida*. Jurnal: Jurusan Agroteknolgi. Universitas Syiah Kuala Darrusalam Banda Aceh.
- [6] Madcoms, 2013. *Adobe Dreamweaver CS6 Dengan Pemograman PHP Dan MySQL*. Yogyakarta: Andi Yogyakarta.
- [7] Pressman, Roger S. 2010. *Rekayasa Perangkat Lunak (Buku Satu)*. Yogyakarta: Andi.
- [8] Nidhra, Srinivas, and Dondeti, Jagruthi, 2012, *Blackbox and Whitebox Testing Techniques - A Literature Review*, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012

LAMPIRAN

LISTING PROGRAM

1. Index

```

<?php
error_reporting(0);
?>

<?php
include "config/library.php";
include "config/koneksi.php";
include "config/fungsi_indotgl.php";
include "menu.php";
//include "../config/fungsi_seo.php";
//error_reporting(0);
/*
if(@$_SESSION['last_activity'] < time()-@$_SESSION['expire_time'] ) { //have
we expired?
//redirect to logout.php
header('Location: logout.php');
} else { //if we haven't expired:
$_SESSION['last_activity'] = time(); //this was the moment of last activity.
}
//$_SESSION['logged_in'] = true; //set you've logged in
$_SESSION['last_activity'] = time(); //your last activity was now, having logged
in.
$_SESSION['expire_time'] = 2*60*60; //expire time in seconds: three hours (you
must change this)
opendb();
$mod=antiinjec(@$_GET['mod']);
$sid_admin=antiinjec(@$_SESSION['ses_admwsadevuid']);

```

```

$stipe=0;
if($sid_admin!="") {
$h_admin=querydb("SELECT id_admin, nama, username, password FROM
cbr_admin WHERE id_admin='$sid_admin'");
$d_admin=mysql_fetch_array($h_admin);
$stipe=1;
}
?>
<!DOCTYPE HTML>
<html>
<head>
<title>CSS3_seascape</title>
<meta name="description" content="website description" />
<meta name="keywords" content="website keywords, website keywords" />
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<!-- modernizr enables HTML5 elements and feature detects -->
<script type="text/javascript" src="js/modernizr-1.5.min.js"></script>
<link rel="icon" type="image/x-icon" href="dist/img/admin_page.png" />
<!-- Bootstrap Core CSS -->
<link href="bower_components/bootstrap/dist/css/bootstrap.min.css"
rel="stylesheet">
<!-- MetisMenu CSS -->
<link href="bower_components/metisMenu/dist/metisMenu.min.css"
rel="stylesheet">
<!-- DataTables CSS -->
<link href="bower_components/datatables-
plugins/integration/bootstrap/3/dataTables.bootstrap.css" rel="stylesheet">
<!-- DataTables Responsive CSS -->
<link href="bower_components/datatables-
responsive/css/dataTables.responsive.css" rel="stylesheet">

```

```

<!-- Custom CSS -->
<link href="dist/css/sb-admin-2.css" rel="stylesheet">
<link href="dist/css/tabel.css" rel="stylesheet">
<!-- Morris Charts CSS -->
<!--<link href="../bower_components/morrisjs/morris.css" rel="stylesheet">-->
<!-- Custom Fonts -->
<link href="bower_components/font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css">
<script type="text/javascript" src="js/jquery.form.min.js"></script>
<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
<script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
<script type="text/javascript" src="tinymce/tinymce.min.js"></script>
<!-- place in header of your html document -->
<script>
tinymce.init({
selector: "textarea.info",
theme: "modern",
relative_urls: false,
height: 280,
plugins: [
"advlist autolink link image lists charmap print preview hr anchor pagebreak
spellchecker",
"searchreplace wordcount visualblocks visualchars code fullscreen insertdatetime
media nonbreaking",
"save table contextmenu directionality emoticons template paste textcolor respon-

```

```

sivefilemanager"
],
content_css: "css/content.css",
toolbar1: "insertfile undo redo | styleselect | fontselect | fontselect | bold italic
underline | alignleft aligncenter alignright alignjustify | bullist numlist outdent in-
dent | link unlink image | print preview media fullpage | forecolor backcolor emot-
icons | responsivefilemanager ",
image_advtab: true ,
style_formats: [
{title: 'Bold text', inline: 'b'},
{title: 'Red text', inline: 'span', styles: {color: '#ff0000'}},
{title: 'Red header', block: 'h1', styles: {color: '#ff0000'}},
{title: 'Example 1', inline: 'span', classes: 'example1'},
{title: 'Example 2', inline: 'span', classes: 'example2'},
{title: 'Table styles'},
{title: 'Table row 1', selector: 'tr', classes: 'tablerow1'}
]
});
</script>
</head>
<body style="background: #D2B48C;">
<div id="main">
<header>
<div id="logo">
<div id="logo_text">
<!-- class="logo_colour", allows you to change the colour of the text -->
<h1></h1>
<h2></h2>
</div>
</div>
<nav>

```

```

<?php menu_atas();?>
<ul class="sf-menu" id="nav">
</ul>
</nav>
</header>
<div id="site_content">
<div class="gallery">
<ul class="images">
<li class="show"></li>
<li></li>
<li></li>
</ul>
</div>
<div id="sidebar_container" style="background: #FFF5EE;">
<div class="sidebar">
<h3>Sistem Pakar</h3>
<h4>Case-Based Reasoning (CBR)</h4>
<h5>February 28, 2021</h5>
<p>CBR Merupakan sebuah pendekatan penyelesaian masalah dengan
menekankan peran pengalaman sebelumnya.<br /><a href="#">Read
more</a></p>
</div>
<div class="sidebar"></div>
</div>
<div class="content">
<?php
if($mod==" " || $mod=="beranda") { include "pages/modul/home.php"; }
elseif($mod=="gejala" && $stipe==1) { include "pag-
es/modul/mod_pakar/1st_gejala.php"; }
elseif($mod=="gejala-input" && $stipe==1) { include "pag-

```

```

es/modul/mod_pakar/inp_gejala.php"; }
elseif($mod=="gejala-kategori" && $stipe==1) { include "pag-
es/modul/mod_pakar/lst_gejala_kategori.php"; }
elseif($mod=="gejala-kategori-input" && $stipe==1) { include "pag-
es/modul/mod_pakar/inp_gejala_kategori.php"; }
elseif($mod=="solusi" && $stipe==1) { include "pag-
es/modul/mod_pakar/lst_solusi.php"; }
elseif($mod=="solusi-input" && $stipe==1) { include "pag-
es/modul/mod_pakar/inp_solusi.php"; }
elseif($mod=="penyakit" && $stipe==1) { include "pag-
es/modul/mod_pakar/lst_penyakit.php"; }
elseif($mod=="penyakit-input" && $stipe==1) { include "pag-
es/modul/mod_pakar/inp_penyakit.php"; }
elseif($mod=="penyakit-solusi-input" && $stipe==1) { include "pag-
es/modul/mod_pakar/inp_penyakit_solusi.php"; }
elseif($mod=="kasus" && $stipe==1) { include "pag-
es/modul/mod_pakar/lst_kasus.php"; }
elseif($mod=="kasus-input" && $stipe==1) { include "pag-
es/modul/mod_pakar/inp_kasus.php"; }
elseif($mod=="kasus-gejala-input" && $stipe==1) { include "pag-
es/modul/mod_pakar/inp_kasus_gejala.php"; }
elseif($mod=="konsultasi") { include "pag-
es/modul/mod_konsultasi/konsultasi_awal.php"; }
elseif($mod=="konsultasi-gejala") { include "pag-
es/modul/mod_konsultasi/konsultasi_gejala.php"; }
elseif($mod=="konsultasi-hasil") { include "pag-
es/modul/mod_konsultasi/konsultasi_hasil.php"; }
elseif($mod=="konsultasi-selesai") { include "pag-
es/modul/mod_konsultasi/konsultasi_selesai.php"; }
elseif($mod=="daftar-konsultasi") { include "pag-
es/modul/mod_konsultasi/konsultasi_list.php"; }

```

```

elseif($mod=="daftar-konsultasi-view") { include "pages/modul/mod_konsultasi/konsultasi_list_view.php"; }
elseif($mod=="info-penyakit") { include "pages/modul/mod_informasi/info_penyakit.php"; }
elseif($mod=="info-penyakit-view") { include "pages/modul/mod_informasi/info_penyakit_view.php"; }
elseif($mod=="info-terapi") { include "pages/modul/mod_informasi/info_solusi.php"; }
elseif($mod=="info-terapi-view") { include "pages/modul/mod_informasi/info_solusi_view.php"; }

//Pengaturan
elseif($mod=="admin" && $stipe==1) { include "pages/modul/mod_pengguna/lst_pengguna.php"; }
elseif($mod=="admin-input" && $stipe==1) { include "pages/modul/mod_pengguna/inp_pengguna.php"; }

//User
elseif($mod=="ubah-password") { include "pages/modul/mod_pengguna/z_user_password.php"; }

?>
</div>
</div>
<footer>
<p>Copyright &copy; Sistem pakar | <a
href="http://www.css3templates.co.uk">design from
css3templates.co.uk</a></p>
</footer>
</div>
<p>&nbsp;</p>
<!-- javascript at the bottom for fast page loading -->
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.easing-sooper.js"></script>

```

```

<script type="text/javascript" src="js/jquery.sooperfish.js"></script>
<script type="text/javascript" src="js/image_fade.js"></script>
<script type="text/javascript">
$(document).ready(function() {
$('ul.sf-menu').sooperfish();
});
</script>
<script src="bower_components/jquery/dist/jquery.min.js"></script>
<!-- Bootstrap Core JavaScript -->
<script src="bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
<!-- Metis Menu Plugin JavaScript -->
<script src="bower_components/metisMenu/dist/metisMenu.min.js"></script>
<!-- DataTables JavaScript -->
<script
src="bower_components/datatables/media/js/jquery.dataTables.min.js"></script>
<script src="bower_components/datatables-
plugins/integration/bootstrap/3/dataTables.bootstrap.min.js"></script>
<!-- Custom Theme JavaScript -->
<script src="dist/js/sb-admin-2.js"></script>
<!-- Page-Level Demo Scripts - Tables - Use for reference -->
<!--responsive: true -->
<?php
if($mod=="daftar-konsultasi") { $col_sort=1; $sort_tipe="desc"; }
else { $col_sort=1; $sort_tipe="asc"; }
?>
<script>
$(document).ready(function() {
var t = $('#tabel_data').DataTable( {
"oLanguage": {
"sSearch": "Pencarian:",
"sLengthMenu": "Tampilkan _MENU_ baris",

```

```

"sZeroRecords": "Data tidak ditemukan (kosong).",
"sInfo": "Baris (_START_ sampai _END_) dari total _TOTAL_ baris",
"sInfoEmpty": "Nol (0) Baris",
"sInfoFiltered": " - dari _MAX_ baris"
},
"columnDefs": [ {
"searchable": false,
"orderable": false,
"targets": 'no_sort'
} ],
"order": [[ <?php echo $col_sort; ?>, '<?php echo $sort_tipe; ?>' ]]
} );
t.on( 'order.dt search.dt', function () {
t.column(0, { search:'applied', order:'applied'}).nodes().each( function (cell, i) {
cell.innerHTML = i+1;
} );
} ).draw();
} );
</script>
</body>
</html>
<?php closedb(); ?>

```

2. Koneksi

```

<?php
//koneksi.php
$dbhost="localhost";
$dbname="yayan";
$dbuser="root";
$dbpassword="";
function opendir()
{
    global $dbhost, $dbuser, $dbpassword, $dbname, $dbconnection;
    $dbconnection=mysql_connect($dbhost, $dbuser, $dbpassword)
    or die ("gagal membuka database");
    $dbselect=mysql_select_db($dbname);
}

```

```

}
function closedb()
{
    global $dbconnection;
    mysql_close($dbconnection);
}
function querydb($query)
{
    $result=mysql_query($query) or die ("gagal melakukan Query=$query");
    return $result;
}
?>

```

3. Library

```

<?php
session_start();
date_default_timezone_set('Asia/Jakarta');
$root_url="http://localhost/Programsistem"; //Ubah ini sesuai url localhost di
komputer Anda
$seminggu = array("Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu");
$hari = date("w");
$hari_ini = $seminggu[$hari];
$tgl_sekarang = date("Ymd");
$tgl_skrng = date("d");
$bln_sekarang = date("m");
$thn_sekarang = date("Y");
$jam_sekarang = date("H:i:s");
$nama_bln=array(1=> "Januari", "Februari", "Maret", "April", "Mei",
                "Juni", "Juli", "Agustus", "September",
                "Oktober", "November", "Desember");
function antiinjec($data){
    $filter_sql =
mysql_real_escape_string(stripslashes(strip_tags(htmlspecialchars($data,ENT_Q
UOTES))));
    return $filter_sql;
}

```

```

function antiinjec_f($data){
    $filter_sql =
mysql_real_escape_string(stripslashes(strip_tags(htmlspecialchars($data,ENT_QUOTES))));
    return $filter_sql;
}
$tgl_full=date("Y-m-d H:i:s");
$sesinf_adminid=1;
function tgl_waktu($data){
    $tgl_waktu=date("d-m-Y H:i:s", strtotime($data));
    return $tgl_waktu;
}
function buat_text($text) {
    $TMPBAGIAN = array();
    $TMP=explode("-", strip_tags(html_entity_decode($text)));
    $jml=count($TMP);
    for($i=0;$i<=$jml;$i++)
    {
        $TMPBAGIAN[$i] = $TMP[$i];
    }
    $VIEW = implode(" ",$TMPBAGIAN);
    return ucfirst($VIEW);
}
function ukuran_file($size)
{
    if ($size >= 1073741824) {
        $fileSize = round($size / 1024 / 1024 / 1024,1) . 'GB';
    } elseif ($size >= 1048576) {
        $fileSize = round($size / 1024 / 1024,1) . 'MB';
    } elseif($size >= 1024) {
        $fileSize = round($size / 1024,1) . 'KB';
    }
}

```

```
    } else {  
        $fileSize = $size . ' bytes';  
    }  
    return $fileSize;  
}  
?>
```

DAFTAR RIWAYAT HIDUP



Nama : Yayan Kiraman
 TTL : Marisa, 04 September 1997
 Alamat : Desa Buntulia Tengah
 Jenis Kelamin : Laki-Laki
 Status Kawin : Belum Kawin
 Agama : Islam
 E-Mail : yayankiraman97@gmail.com

ORANG TUA

Ayah : Marton Kiraman
 Ibu : Saripa Saburi

PENDIDIKAN

1. Tahun 2011, Menyelesaikan Pendidikan di SDN 07 Buntulia.
2. Tahun 2014, Menyelesaikan Pendidikan di MTS AL-Khairaat Buntulia.
3. Tahun 2017, Menyelesaikan Pendidikan di SMK N 1 Duhiadaa
4. Tahun 2017, Mendaftar dan diterima menjadi Mahasiswa di Fakultas Ilmu Komputer, Jurusan Teknik Informatika, Universitas Ichsan Gorontalo