



**PENERAPAN METODE CERTAINTY FACTOR PADA SISTEM PAKAR
DIAGNOSA PENYAKIT IKAN GURAMI BERBASIS WEBSITE
(STUDI KASUS UPTD. BALAI BENIH IKAN KOTA BINJAI)**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

**NAMA : RIZKY YULI LLSTARI
NPM : 1514370543
PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

ABSTRAK

RIZKY YULI LESTARI

**Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Ikan Gurami Berbasis Website
(Studi Kasus UPTD. Balai Benih Ikan Kota Binjai)
2019**

Salah satu jenis ikan air tawar yang sering dibudidayakan oleh para pengusaha perikanan ialah ikan gurami. Hal ini karena ikan gurami memiliki kelebihan yaitu rasa daging yang enak, serta harga relatif stabil. Namun dalam proses budidaya masih sering terjadi kendala yaitu gagal panen, kurangnya pengetahuan para pembudidaya juga kurangnya jumlah para pakar perikanan hal tersebut menjadi salah satu pemicu gagal panen karena ikan sakit dan tingkat kematian yang tinggi. Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, dan merupakan salah satu alternatif untuk mendiagnosa suatu penyakit berdasarkan gejala-gejala yang terjadi. Melalui sistem pakar maka memudahkan dalam mendiagnosa penyakit ikan gurami. Untuk mengidentifikasi diagnosa penyakit ikan gurami dengan metode yang digunakan adalah Metode *Certainty Factor*. Kegunaan dari penelitian ini adalah untuk membantu para pembudidaya ikan gurami untuk menentukan kemungkinan penyakit yang di derita ikan sehingga dapat dilakukan penanganan sejak dini. Dari hasil pembuatan, aplikasi sistem pakar diagnosa awal penyakit ikan gurami diperoleh bahwa sistem dapat berjalan dengan baik dan mampu mendiagnosa penyakit ikan dengan baik.

Kata Kunci: *Penyakit Ikan gurami, Sistem Pakar, Metode Certainty Factor.*

DAFTAR ISI

Halaman

LEMBAR JUDUL	
LEMBAR PENGESAHAN	
ABSTRAK	
KATA PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	vii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB II LANDASAN TEORI.....	5
2.1 Konsep Dasar Sistem	5
2.2 Sistem Pakar	5
2.2.1 Pengertian Sistem Pakar	5
2.2.2 Keuntungan Sistem Pakar	6
2.2.3 Kelemahan Sistem Pakar	7
2.2.4 Konsep Dasar Sistem Pakar	7
2.2.5 Struktur Sistem Pakar	9
2.2.6 Basis Pengetahuan	10
2.3 Certainty Factor	11
2.4 Ikan Gurami	13
2.4.1 Pengertian Ikan Gurami	13
2.4.2 Jenis Ikan Gurami	14
2.4.3 Penyakit Ikan Gurami	14
2.5 Unified Modeling Language (UML).....	15
2.5.1 Pengertian UML.....	15
2.5.2 Konsep Dasar UML	16
2.6 Use Case.....	16
2.7 Class Diagram	18

2.8 Activity Diagram.....	20
2.9 PHP	23
2.10 Perangkat Lunak Yang Digunakan	23
2.10.1 MySQL	23
2.10.2 Adobe Dreamweaver CS6	26
2.10.3 JQuery	27
2.10.4 CSS	27
BAB III METODOLOGI PENELITIAN	28
3.1 Metode Pengumpulan Data	28
3.2 Tahapan Penelitian	29
3.3 Analisis Sistem Yang sedang Berjalan	30
3.3.1 Analisis Proses Sistem Yang sedang Berjalan	30
3.3.2 Kelemahan Proses Sistem Yang sedang Berjalan	30
3.4 Sistem Yang Diusulkan.....	31
3.4.1 Analisis Kebutuhan Sistem	31
3.4.2 Analisis Prosedur	32
3.4.3 Penerapan Metode <i>Certainty Factor</i>	32
3.4.4 Rancangan Sistem	43
3.4.5 Perancangan Pemodelan Sistem	44
3.4.6 Rancangan Interface.....	51
BAB IV HASIL DAN PEMBAHASAN	57
4.1 Pengujian Aplikasi dan Pembahasan	57
4.1.1 Halaman Menu Utama	57
4.1.2 Halaman Menu Diagnosa.....	58
4.1.3 Halaman Hasil Diagnosa.....	59
4.1.4 Halaman Menu Login Admin	60
4.1.5 Halaman Menu Utama Admin	61
4.1.6 Halaman Menu <i>Input</i> Data Gejala	61
4.1.7 Halaman <i>Input</i> Data Penyakit	63
4.1.8 Halaman <i>Input</i> Data Nilai Gejala Penyakit.....	64
4.1.9 Laporan	64

BAB V PENUTUP	66
5.1 Kesimpulan	66
5.2 Saran.....	66

DAFTAR PUSTAKA

DAFTAR GAMBAR

	Halaman
Gambar 3.1 Tahapan penelitian	29
Gambar 3.2 <i>use case</i> diagram	45
Gambar 3.3 <i>Activity</i> diagram Diagnosa	46
Gambar 3.4 <i>Activity</i> diagram Login	46
Gambar 3.5 <i>Activity</i> diagram Tambah Admin	47
Gambar 3.6 <i>Activity</i> diagram Data Gejala.....	47
Gambar 3.7 <i>Activity</i> diagram Data Penyakit.....	48
Gambar 3.8 <i>Activity</i> diagram Data Nilai Gejala Penyakit.....	48
Gambar 3.9 <i>Activity</i> diagram Laporan	49
Gambar 3.10 Rancangan Tampilan Awal	51
Gambar 3.11 Rancangan Menu Diagnosa.....	52
Gambar 3.12 Rancangan Form Hasil Diagnosa.....	52
Gambar 3.13 Rancangan Menu About US.....	53
Gambar 3.14 Rancangan Form Login Admin.....	53
Gambar 3.15 Rancangan <i>Home</i> Admin	54
Gambar 3.16 Rancangan Form Admin	54
Gambar 3.17 Rancangan Menu Data Gejala.....	55
Gambar 3.18 Rancangan Menu Data Penyakit	55
Gambar 3.19 Rancangan Form Nilai Gejala Penyakit	56
Gambar 3.20 Rancangan Form Laporan	56
Gambar 4.1 Tampilan Halaman Utama Pengguna	57
Gambar 4.2 Tampilan Menu Diagnosa	58
Gambar 4.3 Form Tampilan Hasil Diagnosa	59
Gambar 4.4 Halaman Login Admin.....	60
Gambar 4.5 Halaman Utama Admin.....	61
Gambar 4.6 Halaman <i>Input</i> Data Gejala	62
Gambar 4.7 Halaman Data Gejala	52
Gambar 4.8 Halaman <i>Input</i> Data Penyakit	63
Gambar 4.9 Halaman Data Penyakit.....	63
Gambar 4.10 Halaman <i>Input</i> Data Nilai Gejala Penyakit.....	64
Gambar 4.11 Halaman Laporan	65

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol <i>Use Case</i>	17
Tabel 2.2 Simbol <i>Class Diagram</i>	19
Tabel 2.3 Simbol <i>Activity Diagram</i>	21
Tabel 3.1 Data Gejala Penyakit Pada Ikan Gurami.....	32
Tabel 3.2 Pengkodean Gejala.....	33
Tabel 3.3 Pengkodean Penyakit	34
Tabel 3.4 Hubungan Gejala Dan Penyakit.....	34
Tabel 3.5 <i>Rule</i> Penyakit	35
Tabel 3.6 Solusi Pengendalian Penyakit	35
Tabel 3.7 Nilai CF <i>USer</i>	37
Tabel 3.8 Nilai CF Pakar	37
Tabel 3.9 Persentase Kesimpulan	37
Tabel 3.10 Admin.....	49
Tabel 3.11 Gejala	49
Tabel 3.12 Penyakit	49
Tabel 3.13 Gejala Penyakit	50
Tabel 3.14 Diagnosa.....	50

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ikan merupakan hewan yang hidup di air yang menjadi salah satu dari sekian banyak sumber protein yang dibutuhkan manusia. Ikan sangat bermanfaat bagi manusia sebab mengandung bermacam zat yang dibutuhkan oleh tubuh manusia. Selain itu apabila dibandingkan dengan sumber penghasil protein lain seperti daging, susu dan telur harga ikan relatif lebih murah. Mengingat pentingnya ikan bagi manusia, tak heran bila manusia berusaha mendapatkan ikan dalam jumlah yang mencukupi, antara lain dengan mengusahakan melakukan pencarian di sumbernya yakni sungai dan laut, ada pula yang memeliharanya dengan sebaik-baiknya yang lazim disebut dengan usaha perikanan

Salah satu jenis ikan air tawar yang sering dibudidayakan oleh para pengusaha perikanan ialah ikan gurami. Gurami (*Osphronemus gouramy*) merupakan ikan asli perairan Indonesia yang sudah menyebar ke wilayah Asia Tenggara dan Cina. Ikan ini merupakan salah satu jenis ikan konsumsi yang sudah cukup dikenal dan banyak diminati di Indonesia. Hal ini karena ikan gurami memiliki kelebihan yaitu rasa daging yang enak, serta harga relatif mahal.

Seiring dengan meningkatnya permintaan pasar, perlu diimbangi dengan peningkatan produksi. Namun dalam proses budidaya masih sering terjadi kendala yaitu gagal panen dan kerugian karena penyakit ikan yang terlambat ditangani, kurangnya pengetahuan para pembudidaya juga kurangnya jumlah para pakar perikanan hal tersebut menjadi salah satu pemicu gagal panen karena ikan sakit

dan tingkat kematian yang tinggi, oleh karena itu kebutuhan informasi tentang penyakit ikan gurami sangat penting, alternatif yang dapat dilakukan yaitu dengan membuat sistem pakar diagnosa penyakit ikan gurami

sistem pakar yang dibangun diharapkan dapat membantu pembudidaya agar lebih mudah dan cepat dalam melakukan konsultasi dan mengetahui penyakit ikan gurami lebih dini. Sistem pakar adalah program komputer yang mengatasi masalah di dunia nyata untuk membutuhkan interpretasi suatu pakar. Sistem pakar yang dibangun bukan untuk menggantikan posisi pakar perikanan tapi untuk membantu pekerjaan pakar dan memudahkan pembudidaya dalam berkonsultasi tanpa terkendala jarak dan waktu. Metode yang digunakan adalah *Certainty factor* untuk mengatasi ketidakpastian dalam data

Berdasarkan uraian di atas, maka penulis tertarik melakukan penelitian yang berjudul “**Penerapan Metode *Certainty Factor* Untuk Diagnosa Penyakit Ikan Gurami Berbasis Website (Studi Kasus UPTD. Balai Benih Ikan Kota Binjai)**”

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat disusun rumusan masalah sebagai berikut:

- a. Bagaimana menerapkan metode *Certainty Factor* pada aplikasi sistem pakar dalam mendiagnosa penyakit ikan gurami?
- b. Bagaimana merancang dan membangun sistem pakar dalam mendiagnosa penyakit ikan gurami dengan menggunakan metode *Certainty Factor*?

1.3 Batasan Masalah

Pokok pembahasan dalam penelitian ini dibatasi sebagai berikut:

- a. Sistem pakar ini hanya mendiagnosa benih ikan gurami
- b. Aplikasi yang dibangun berbasis website dengan menggunakan bahasa pemrograman *PHP*, menggunakan *database MySQL* dengan metode *Certainty Factor*
- c. Pengujian masih secara offline
- d. Lokasi penelitian dilakukan di UPTD. Balai Benih Ikan Kota Binjai yang beralamat di Jl. Madura No. 34 A Kel. Kebun Lada Kec. Binjai Utara

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

- a. Untuk menerapkan Metode *Certainty factor* pada aplikasi sistem pakar diagnosa penyakit ikan gurami
- b. Untuk merancang dan membangun sistem pakar dalam mendiagnosa penyakit ikan gurami

1.5 Manfaat Penelitian

Perancangan aplikasi sistem pakar untuk mengetahui penyakit ikan gurami ini bermanfaat bagi penulis dan masyarakat luas antara lain :

- a. Menambah pengetahuan pengguna untuk mengambil keputusan dalam mendiagnosa penyakit ikan gurami
- b. Membantu pembudidaya ikan gurami dalam mengetahui penyakit ikan gurami agar tidak merugi
- c. Membantu pekerjaan para pakardalam bidang perikanan

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Menurut Hartono (2013:10), Sistem yakni suatu benda atau entitas (yaitu himpunan dari berbagai bagian atau komponen), dan sekaligus juga suatu proses atau metode cara untuk mencapai tujuan (yaitu saling berhubungan secara terstruktur berdasar fungsi-fungsinya). Menurut Rosa dan shalahudin (2013:291), Sistem Kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai. Berdasarkan pendapat dua ahli tersebut maka penulis menyimpulkan bahwa “Sistem adalah kumpulan komponen saling terkait, saling berhubungan secara terorganisasi dan mempunyai tujuan yang ingin dicapai”.

2.2 Sistem Pakar

2.2.1 Pengertian sistem pakar

Sistem pakar adalah program komputer yang merupakan cabang dari penelitian ilmu komputer yang disebut AI (*Artificial Intelligence*). Tujuan ilmu AI adalah membuat sesuatu menjadi cerdas dalam hal pemahaman melalui program komputer yang ditunjukkan dengan tingkah laku cerdas. Hal ini berkenaan dengan suatu konsep dan metode inferensi simbolik atau penalaran yang dilakukan komputer, dan berkenaan juga dengan bagaimana suatu pengetahuan digunakan untuk membuat

suatu kesimpulan yang akan direpresentasikan ke dalam suatu mesin (Desiani, Arhami : 2006)

Professor Edward Feigenbaum dari Universitas Stanford yang merupakan seorang pelopor awal dari teknologi sistem pakar, yang mendefinisikan sistem pakar sebagai “Suatu program komputer cerdas yang menggunakan *knowledge* (pengetahuan) dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang ahli untuk menyelesaikannya” (Feigenbaum: 1982). Suatu sistem pakar adalah suatu sistem komputer yang menyamai (*emulates*) kemampuan pengambilan keputusan dari seorang pakar. Istilah *emulates* berarti bahwa sistem pakar diharapkan dapat bekerja dalam semua hal seperti seorang pakar (Arhami : 2005)

2.2.2 Keuntungan Sistem Pakar

keuntungan sistem pakar menurut Arhami (2005 : 10) adalah:

1. menjadikan pengetahuan dan nasihat lebih mudah di dapat
2. meningkatkan output dan produktivitas
3. menyimpan kemampuan dan keahlian pakar
4. meningkatkan penyelesaian masalah, menerusi paduan pakar, penerangan, sistem pakar khas
5. meningkatkan reliabilitas
6. memberikan *response* (jawaban) yang cepat
7. merupakan panduan *intelligence* (cerdas)

8. dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian
9. *intelligence database* (basis datacerdas), bahwa sistem pakar dapat digunakan untuk mengakses basis data dengan cara cerdas (Kerschberg : 86, Schur : 88)

2.2.3 Kelemahan Sistem Pakar

Sistem pakar seperti halnya sistem lainnya juga memiliki kelemahan, diantaranya adalah :

1. Masalah dalam mendapatkan pengetahuan di mana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan walaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaannya
3. Boleh jadi sistem tak dapat membuat keputusan
4. Sistem pakar tidaklah 100% menguntungkan, walaupun seorang tetap tidak sempurna atau tidak selal benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan. Dalam hal ini peran manusia tetap merupakan faktor dominan

2.2.4 Konsep Dasar Sistem Pakar

Pengetahuan dari suatu sistem pakar mungkin dapat direpresentasikan dalam sejumlah cara. Salah satu metode yang paling umum untuk merepresentasikan pengetahuan adalah dalam bentuk tipe aturan (*rule*) IF...THEN (Jika...maka). Konsep dasar dari suatu sistem pakar mengandung beberapa unsur atau elemen, yaitu (Kusrini, 2006):

1. Kepakaran

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar.

2. Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya.

3. Pengalihan Kepakaran

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

- a. Akuisisi pengetahuan (dari pakar atau sumber lain)
- b. Representasi pengetahuan (pada komputer)
- c. Inferensi pengetahuan
- d. Pemindahan pengetahuan ke pengguna
- e. Inferensi

Inferensi adalah sebuah prosedur yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

4. Aturan

Kebanyakan aplikasi sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based system*), yaitu pengetahuan disimpan terutama dalam bentuk *rule* , sebagai prosedur-prosedur pemecahan masalah.

5. Kemampuan Menjelaskan

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi- operasinya.

2.2.5 Struktur Sistem Pakar

Sistem pakar dibagi menjadi empat bentuk, yaitu:

1. Berdiri Sendiri.

Sistem pakar jenis ini merupakan perangkat lunak yang berdiri sendiri tidak bergabung dengan *software* lainnya.

2. Tergabung,

Sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana di dalamnya memanggil algoritma subrutin lain (konvensional).

3. Menghubungkan ke perangkat lunak lain

Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya DBMS.

4. Sistem Mengabdikan

Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

2.2.6 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan dalam menyelesaikan masalah, tentu di dalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang umum digunakan, yaitu:

1. Penalaran berbasis aturan (*Rule-Based Reasoning*)

Pengetahuan direpresentasikan dengan menggunakan aturan berbentuk: *IFTHEN*. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan si pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu juga digunakan apabila dibutuhkan penjelasan tentang langkah-langkah pencapaian solusi.

2. Penalaran berbasis kasus (*Case-Based Reasoning*)

Basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang sekarang. Bentuk ini akan digunakan apabila *user* menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama. Selain itu, bentuk ini juga digunakan apabila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan (Kusumadewi, 2003: 105).

2.3 Certainty Factor

Menurut Fitri Wulandari dan Ihsan Yuliandri (2014 : 306) *Certainty factor* adalah suatu metode untuk membuktikan apakah suatu fakta itu pasti atau tidak pasti yang berbentuk *metric* yang biasanya digunakan dalam sistem pakar. Metode ini sangat cocok untuk sistem pakar yang mendiagnosis sesuatu yang belum pasti. Tahapan dalam merepresentasikan data-data kualitatif :

1. Kemampuan untuk mengekspresikan derajat keyakinan sesuai dengan metode yang sudah dibahas sebelumnya.

2. Kemampuan untuk menempatkan dan mengkombinasikan derajat keyakinan tersebut dalam sistem pakar

Dalam mengekspresikan derajat keyakinan digunakan suatu nilai yang disebut *Certainy Factor* (CF) untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. Berikut adalah formulasi dasar dari *Certainy Factor*:

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

Keterangan :

CF = *Certainy Factor* (faktor kepastian) dalam hipotesis H yang dipengaruhi oleh fakta E.

MB = *Measure of Belief* (tingkat keyakinan), adalah ukuran kenaikan dari kepercayaan *hipotesis* H dipengaruhi oleh fakta E.

MD = *Measure of Disbelief* (tingkat ketidakpercayaan), adalah kenaikan ketidakpercayaan hipotesis H dipengaruhi fakta E.

E = *Evidence* (peristiwa atau fakta).

H = *Hipotesis* (Dugaan).

Untuk mengkombinasikan dua atau lebih aturan, sistem berbasis pengetahuan dengan beberapa aturan, masing-masing darinya menghasilkan kesimpulan yang sama tetapi faktor ketidakpastiannya berbeda, maka setiap aturan dapat ditampilkan sebagai potongan bukti yang mendukung kesimpulan bersama. Untuk menghitung CF (keyakinan) dari kesimpulan diperlukan bukti pengkombinasian sebagai berikut :

$$CF (R1,R2) = CF (R1) + [CF (R2)] \times [1-CF(R1)]$$

Jika kita hanya menambahkan CF R1 dan R2, kepastian kombinasinya akan lebih dari 1. Memodifikasikan jumlah kepastian melalui penambahan factor kepastian kedua dan mengalikannya (1 dikurangi faktor kepastian pertama). Jadi, semakin besar CF pertama semakin kecil kepastian penambah kedua. Tetapi faktor tambahan selalu menambahkan beberapa kepastian. Untuk aturan ketiga yang ditambahkan, dapat digunakan aturan sebagai berikut

(Hermawati, 2011).

$$CF(R1, R2, R3) = CF(R1, R2) + [CF(R3)] [1 - CF(R1, R2)] = CF(R1, R2) + CF(R3) - [CF(R1, R2)] \cdot [CF(R3)]$$

Untuk solusi dengan lebih banyak aturan dapat menggunakan persamaan yang secara bertingkat seperti pada persamaan diatas.

2.4 Ikan Gurami

2.4.1 Pengertian Ikan Gurami

Gurame merupakan jenis ikan konsumsi air tawar, bentuk badan pipih lebar, bagian punggung berwarna merah sawo dan bagian perut berwarna kekuning-kuningan/ keperak-perakan. Ikan gurame merupakan keluarga Anabantidae, keturunan Helostoma dan bangsa Labyrinthici. Ikan gurami berasal dari perairan daerah Sunda (Jawa Barat, Indonesia), dan menyebar ke Malaysia, Thailand, Ceylon dan Australia. Pertumbuhan ikan gurame agak lambat dibanding ikan air tawar jenis lain. Di Indonesia, orang Jawa menyebutnya gurami, Gurameh, orang Sumatra ikan kalau, kala, kalui, sedangkan di Kalimantan disebut Kalui. Orang Inggris

menyebutnya “Giant Gouramy”, karena ukurannya yang besar sampai mencapai berat 5 kg.

2.4.2 Jenis Ikan Gurame

Klasifikasi ikan gurame adalah sebagai berikut :

Klas : Pisces
 Sub Kelas : Teleostei
 Ordo : Labyrinthici
 Sub Ordo : Anabantoidae
 Famili : Anabantidae
 Genus : Osphronemus
 Species : Osphronemus goramy (Lacepede)

Jenis gurami yang sudah dikenal masyarakat diantaranya: gurami angsa, gurami jepun, blausafir, paris, bastar dan porselen. Empat terakhir banyak dikembangkan di Jawa Barat, khususnya Bogor. Dibanding gurame jenis lain, porselen lebih unggul dalam menghasilkan telur. Jika induk bastar dalam tiap sarangnya hanya mampu menghasilkan 2000-3000 butir telur, porselen mampu 10.000 butir. Karena itu masyarakat menyebutnya sebagai *top of the pop*, dan paling banyak diunggulkan.

2.4.3 Penyakit Ikan Gurami

Gangguan yang dapat menyebabkan matinya ikan adalah penyakit yang disebut penyakit non parasiter dan penyakit yang disebabkan parasit. Gangguan-gangguan non parasiter bisa berupa pencemaran air seperti adanya gas-gas beracun berupa asam belerang atau amoniak; kerusakan akibat penangkapan atau kelainan tubuh karena keturunan. Penanggulangannya adalah dengan mendeteksi keadaan kolam dan perilaku ikan-ikan tersebut. Memang diperlukan pengetahuan dan pengalaman yang cukup untuk mengetahuinya. Ikan-ikan yang sakit biasanya menjadi kurus dan lamban gerakannya. Gangguan lain yang berupa penyakit parasiter, yang diakibatkan oleh bakteri, virus, jamur dan berbagai mikroorganisme lainnya

2.5 *Unified Modeling language (UML)*

2.5.1 *Pengertian Unified Modeling language (UML)*

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Menurut Sri Dharwiyanti dan Sri Dharwiyanti (2012 : h.2) “*Unified Modelling Language (UML)* adalah sebuah "bahasa" yg telah menjadi standar dalam

industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak”. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntaxmendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

2.5.2 Konsep Dasar *Unified Modeling language*



Abstraksi konsep dasar UML yang terdiri dari structural classification, dynamic behavior, dan model management, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari Diagrams. Main concepts bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan *view* adalah kategori dari diagram tersebut.


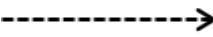






2.6 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke

sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang diincludeakan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain

Tabel 2.1 Simbol Use Case

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Generalization</i>	Hubungan dimana objek anak (descendant) berbagi perilaku dan struktur data dari objek yang

		ada di atasnya objek induk (ancestor)
	<i>Include</i>	Menspesifikasikan use case sumber eksplisit
	<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
	<i>Use Case</i>	deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen elemennya (sinergi)
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : Rosa A.S, M.Shalahuddin, 2013.

2.7 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan



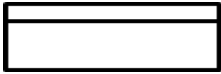
layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, packagedan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lainlain. *Class* memiliki tiga area pokok :





- 1) Nama (dan *stereotype*)
- 2) Atribut
- 3) Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a) *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
- b) *Protected*, hanya dapat dipanggil oleh classyang bersangkutan dan anak-anak yang mewarisinya
- c) *Public*, dapat dipanggil oleh siapa saja

Tabel 2.2 Simbol Class Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimmananobjek anak (descendant)berbagi perilaku dan struktur data dari objekyang ada di atasnya objekinduk (ancestor)
	<i>Navy Associaton</i>	Upaya untuk menghindariasosia dengan lebih dari 2objek
	<i>Class</i>	Himpunanan dari objekobjek yang berbagi atributserta operasi yang sama

	<i>Collaboration</i>	Interaksi aturan-aturandan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen elemennya (sinergi)
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber : Yuni Sugiarti, 2013.

Class dapat merupakan implementasi dari sebuah interface, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time. Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar Class

- 1) Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah *query* antar class
- 2) Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

- 3) Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asal nya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
- 4) Hubungan dinamis, yaitu rangkaian pesan (message) yang di-passing dari satu class kepada class lain.

2.8 Activity Diagram


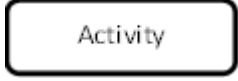
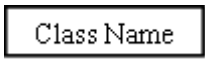




Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.


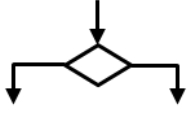
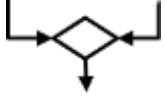
Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk

menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork dan join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Tabel 2.3 Activity Diagram

NAMA ELEMEN	FUNGSI	NOTASI
<i>Action</i>	Untuk menggambarkan perilaku sederhana dan bersifat non-decomposable	
<i>Activity</i>	Untuk mewakili kumpulan aksi (action)	
<i>Object Node</i>	Untuk mewakili objek yang terhubung dengan kumpulan objek flow	
<i>Control Flow</i>	Menunjukkan rangkaian pelaksanaan	
<i>Object Flow</i>	Menunjukkan aliran sebuah objek dari sebuah aktivitas atau aksi lainnya	
<i>Initial Node</i>	Menandakan awal dari kumpulan aksi atau aktivitas	
<i>Final-Activity Node</i>	Untuk menghentikan sebuah control flow atau object flow pada sebuah aktivitas (atau aksi)	
	Untuk menghentikan control flow	

<i>Final-Flow-Node</i>	atau objectflow tertentu	
<i>Decision Node</i>	Untuk mewakili suatu kondisi pengujian yang bertujuan untuk memastikan bahwa controlflow suatu object hanya menuju ke satu arah	
<i>Merge Node</i>	Untuk menyatukan kembali decision path yang di buat dengan menggunakan decision Node	

Sumber : Rosa A.S, M.Shalahuddin, 2013

2.9 PHP

Edy Winarno (2013 : 1) PHP adalah bahasa pemrograman web bersifat server side, yang tujuannya untuk menghasilkan skrip yang akan di *generate* dalam kode HTML yang merupakan bahasa standar web. PHP adalah bahasa pemrograman open source, PHP adalah singkatan dari *PHP Hypertext Pre-processor*.

2.10. Perangkat Lunak Yang Digunakan

2.10.1 MySQL

MYSQL adalah DBMS yang di distribusikan secara gratis dibawah lisensi dari *General Public License*(GPL), dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh untuk dijadikan program induk turunan bersifat *closer source*(komersial).

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basisdata sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah

konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, perubahan, dan penghapusan data yang mungkin dapat di kerjakan dengan mudah dan otomatis Berikut beberapa Keunggulan dari MYSQL, diantaranya :

1. *Portability*

Dapat berjalan stabil pada berbagai sistem operasi, diantaranya:
Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Asigma.

2. *Opensource*

sistem pengembangan yang tidak dikoordinasi oleh suatu individu / lembaga pusat, tetapi oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber (*source-code*) yang tersebar dan tersedia bebas (biasanya menggunakan fasilitas komunikasi internet).

3. Didistribusikan secara gratis dibawah lisensi dari *general public lisencc(GPL)* dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh menggunakan MYSQL untuk dijadikan turunan yang bersifat *close source* (komersial)

4. *Multi user*

Dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan

5. *Performance Tuning*

Mempunyai kecepatan yang tinggi dalam menangani *query*

6. *Column Types*

Memiliki tipe data yang sangat kompleks, seperti: *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year dan enum*.

7. *Command and Function*

Memiliki operator dan fungsi penuh yang mendukung *select* dan *where* dalam *query*.

8. *Security*

Memiliki beberapa lapisan keamanan seperti tingkat *subnetwork, hostname, privilege user* dengan sistem perijinan yang mendetail serta *password* yang ter-enkripsi. i) *Scalability dan Limits* Mampu menangani basis data dalam jumlah besar dengan jumlah *field* lebih dari 50 juta, 60 ribu *table* dan 5 milyar *record*. Batas indeks mencapai 32 buah per tabel

9. *Localization*

Data dapat mendeteksi pesan kesalahan (*error code*) pada klien dengan menggunakan lebih dari 20 bahasa

10. *Connectivity*

Dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix, socket Named Pipes*

11. *Interface*

Memiliki antarmuka terhadap berbagai aplikasi dan Bahasa pemrograman dengan menggunakan fungsi API

12. *Clients tools*

Dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basisdata sekaligus dokumen penunjuk *online*

13. Struktur table

Memiliki struktur *table* yang lebih fleksibel dalam menangani *alter table* dibandingkan dengan *PostgreSQL* dan *Oracle*.

2.10.2 Adobe *Dreamweaver CS6*

Dreamweaver merupakan software utama yang digunakan oleh desainer web dan programmer web untuk mengembangkan suatu situs web. *Dreamweaver* memiliki ruang kerja, fasilitas, dan kemampuan yang mampu meningkatkan produktivitas dan efektivitas, baik dalam desain maupun pembuatan situs web

dalam perkembangannya, adobe *dreamweaver* telah mencapai versinya yang terbaru atau lebih dikenal dengan adobe *dreamweaver CS6*. Menurut ELCOM (2013:1) *Dreamweaver CS6* memiliki peningkatan dalam kemampuan toolbar, yaitu dapat digunakan untuk memodifikasi tampilan toolbar atau menambahkan fungsi baru, selain antarmuka muka pengguna baru, *dreamweaver CS6* memiliki kemampuan untuk menyunting kode dengan lebih baik. *Dreamweaver CS6* juga dapat melakukan print kode pada jendela code view, serta memiliki fasilitas code Hints yang membantu dalam urutan tag-tag, serta tag inspector yang sangat berguna dalam menangani tag-tag HTML

Dreamweaver adalah sebuah HTML editor profesional untuk mendesign web secara visual dan mengelola situs atau halaman web. *Adobe Dreamweaver CS6*

memiliki beberapa kemampuan bukan hanya sebagai *software* untuk *design web* saja tetapi juga untuk menyunting kode serta pembuatan aplikasi *web* dengan menggunakan berbagai bahasa pemrograman *web*, antara lain: HTML, *ColdFusion*, PHP, CSS, *Javascript* dan XML.

2.10.3 JQuery

Menurut Hakim (2010 : 3) “JQuery adalah Javascript Library kumpulan kode/fungsi Javascript siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode Javascript”. Secara standar, apabila membuat kode Javascript, maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami.

2.10.4 CSS

Menurut ElCOME (2013 : 7) CSS kependekan dari *cassading Style Sheet* yang berfungsi untuk mengatur seluruh tampilan halaman web. CSS dapat mengatur posisi *layout*, jenis, dan tipe huruf dengan mudah dan fleksibel, bahkan CSS *style* juga dapat digunakan untuk mengatur properti yang tidak dapat diatur jika hanya menggunakan HTML.

BAB III

METODE PENELITIAN

3.1 Metode Pengumpulan Data

1. Studi Literatur

Metode ini memperoleh informasi dengan mengumpulkan referensi yang dapat mendukung sebagai landasan teori dan dasar pedoman dari pembuatan laporan.

2. Studi Lapangan

Memperoleh data atau informasi secara langsung ke lapangan dengan teknik pengumpulan data sebagai berikut :

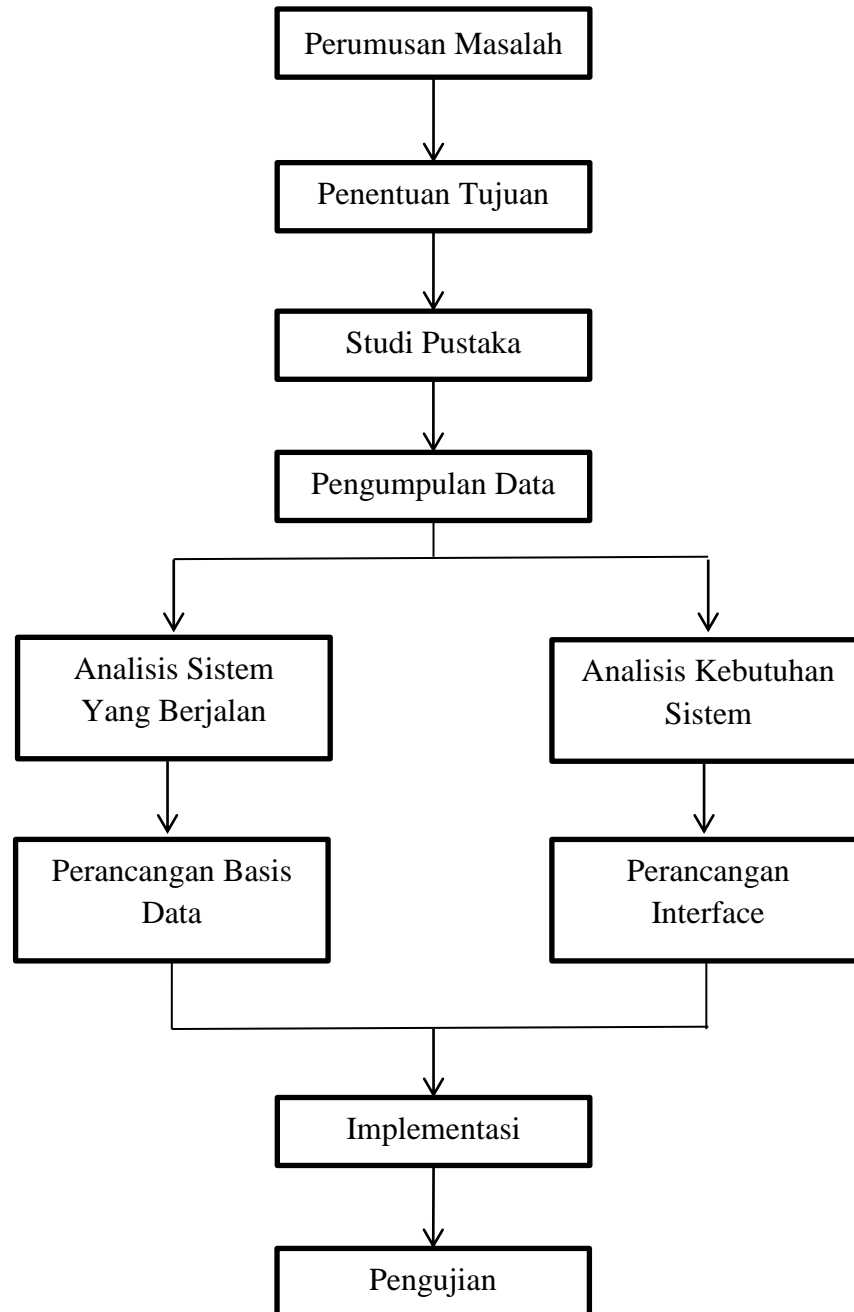
a. Wawancara (*Interview*)

Wawancara dilakukan untuk mencari sumber kajian, landasan teori, tata cara dalam perumusan masalah, pengumpulan data dan informasi, pengolahan data dan penarikan kesimpulan, serta saran sebagai acuan penyusun penulisan.

b. Observasi

Metode ini dilakukan dengan observasi ke lapangan dengan cara melakukan pengamatan langsung ke tempat dijadikan objek penelitian secara sistematis terhadap unsur yang telah diteliti.

3.2 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

3.3 Analisis Sistem Yang Sedang Berjalan

3.3.1 Analisis Proses Sistem Yang Sedang Berjalan

Dalam proses konsultasi para pembudidaya ikan dengan para pakar di UPTD. Balai Benih Ikan Kota Binjai masih dilakukan secara manual yaitu dengan cara pembudidaya datang ke balai untuk melakukan konsultasi tentang kondisi ikannya, lalu para pakar akan mengambil kesimpulan dari informasi gejala kondisi yang diberikan pembudidaya tentang jenis penyakit serta cara penanggulangannya.

3.3.2 Kelemahan Proses Sistem Yang Sedang Berjalan

Dari tahapan analisis kelemahan dapat diketahui dengan jelas masalah-masalah apa saja yang sering muncul dalam menentukan penyakit ikan gurami. Berdasarkan hasil penelitian yang dilakukan didapatkan permasalahan dan kelemahan terhadap budidaya ikan gurami sebagai berikut:

1. Pengecekan penyakit ikan gurami harus dilakukan oleh pakar perikanan atau dokter hewan secara berkala.
2. Penentuan diagnosa penyakit yang kurang teliti mengakibatkan kurang tepat dalam pemilihan penanganan terhadap penyakit ikan gurami.
3. Proses penanganan terhadap penyakit ikan tidak dapat dilakukan secara cepat.
4. Penyimpanan data hasil diagnosa penyakit ikan gurami masih menggunakan kertas untuk pencatatan datanya.

3.4 Sistem Yang Diusulkan

Setelah melakukan analisis terhadap sistem yang berjalan, maka penulis dapat menarik kesimpulan atas sistem yang berjalan yakni dengan mengetahui kelemahan sistem yang ada, adapun kelemahan sistem yang telah diamati adalah dalam proses pendiagnosaan penyakit dengan cara datang secara langsung ke sumber pakar ataupun pakar yang meninjau langsung keadaan yang ada di kolam budidaya, hal tersebut akan memakan banyak waktu yang lama terlebih banyaknya pembudidaya ikan di Kota Binjai yang membutuhkan bantuan para pakar, disertai ketersediaan para pakar perikanan di kota Binjai yang masih sedikit, maka penulis mengusulkan untuk menerapkan metode *certainty factor* pada sistem pakar untuk mendiagnosa penyakit ikan gurami berbasis website dengan menggunakan sistem pakar maka pekerjaan para pakar akan terbantu serta para pembudidaya dapat melakukan konsultasi dan mendapatkan informasi tanpa terkendala jarak dan waktu

3.4.1 Analisis Kebutuhan Sistem

Dari hasil pengamatan, wawancara dan observasi yang dilakukan di UPTD. Balai Benih Ikan Kota Binjai, maka di dapat beberapa hal yang dianggap penting dalam kebutuhan penerapan metode *certainty factor* untuk mendiagnosa penyakit ikan gurami. Untuk implementasi sistem dibutuhkan perangkat komputer didalam UPTD. Balai Benih Ikan Kota Binjai untuk digunakan oleh user, user harus mempunyai kemampuan dalam mengoperasikan perangkat tersebut. Kebutuhan dari hasil analisis ini harus dapat dilaksanakan diukur, diuji, terkait

dengan kebutuhan yang teridentifikasi, serta didefinisikan sampai tingkat detail yang memadai untuk desain sistem.

3.4.2 Analisis Prosedur

Analisis prosedur sistem yang berjalan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem pakar tersebut, sehingga kelebihan dan kekurangan sistem pakar dapat diketahui dengan cara menerapkan metode *Certainty Factor* (CF)

3.4.3 Penerapan Metode *Certainty Factor*

Dalam penerapan metode *certainty factor* (CF) dalam diagnosa penyakit ikan gurami yang baik diperlukan kriteria, nilai dan bobot untuk melakukan perhitungannya agar di dapat *alternative* terbaik.

1. Kriteria, Nilai dan bobot

Penentuan diagnosa penyakit ikan gurami dengan menggunakan metode *certainty factor* (CF) terdapat kriteria yang dibutuhkan untuk menentukan bahwa ikan gurami terjangkit penyakit apa sesuai dengan kriteria yang telah di tentukan. Adapun kriterianya adalah sebagai berikut :

Tabel 3.1 Data Gejala Penyakit Pada Ikan Gurami

No	Nama Gejala	Nama Penyakit			
		Strepto coccus	Lerniasis	Argulus	Sapro legniasis
1	Berenang tidak normal (berputar-putar)	✓			
2	Nafsu makan menurun	✓			
3	Ikan kurus / terlihat lemah	✓			
4	Tubuh bewarna gelap	✓			

5	Mata menonjol (mata belo)	✓			
6	Menggosok-gosokkan badan		✓		
7	Luka berlumut		✓		
8	Pendarahan & bernanah pada tubuh		✓		
9	Terdapat cacing yang menempel pada tubuh ikan seperti panah		✓		
10	Berenang zig-zag atau meloncat-loncat			✓	
11	Peradangan pada kulit (tubuh)			✓	
12	Terdapat kutu pada sirip / sisik / insang			✓	
13	Luka berwarna putih atau kecoklatan				✓
14	Ada benang halus seperti kapas menempel pada luka atau telur				✓

Tabel 3.2 Pengkodean Gejala

Kode Gejala	Nama Gejala
G001	Berenang tidak normal (berputar-putar)
G002	Nafsu makan menurun
G003	Ikan kurus / terlihat lemah
G004	Tubuh bewarna gelap
G005	Mata menonjol (mata belo)
G006	Menggosok-gosokkan badan
G007	Luka berlumut
G008	Pendarahan & bernanah pada tubuh
G009	Terdapat cacing yang menempel pada tubuh ikan seperti panah

G010	Berenang zig-zag atau meloncat-loncat
G011	Peradangan pada kulit (tubuh)
G012	Terdapat kutu pada sirip / sisik / insang
G013	Luka berwarna putih atau kecoklatan
G014	Ada benang halus seperti kapas menempel pada luka atau telur

Tabel 3.3 Pengkodean Penyakit

Kode Penyakit	Nama Penyakit
P1	Streptococcus
P2	Lerniasis
P3	Argulus
P4	Saprolegniasis

Tabel 3.4 Hubungan Gejala dan Penyakit

No	Nama Gejala	Nama Penyakit			
		Strepto Coccus	Lerniasis	Argulus	Sapro legniasis
1	Berenang tidak normal (berputar-putar)	0.4			
2	Nafsu makan menurun	0.4			
3	Ikan kurus / terlihat lemah	0.4			
4	Tubuh bewarna gelap	0.6			
5	Mata menonjol (mata belo)	0.8			
6	Menggosok-gosokkan badan		0.4		
7	Luka berlumut		0.4		
8	Pendarahan & bernanah pada tubuh		0.8		
9	Terdapat cacing yang menempel pada tubuh ikan seperti panah		0.8		
10	Berenang zig-zag atau			0.4	

	meloncat-loncat				
11	Peradangan pada kulit (tubuh)			0.6	
12	Terdapat kutu pada sirip / sisik / insang			0.8	
13	Luka berwarna putih atau kecoklatan				0.4
14	Ada benang halus seperti kapas menempel pada luka atau telur				0.8

Tabel 3.5 Rule Penyakit

Rule	IF	Then
1	G001, G002, G003, G004 and G005 true	P1
2	G006, G007, G008 and G009 true	P2
3	G010, G011 and G012 true	P3
4	G013 and G014 true	P4

Tabel 3.6 Solusi Pengendalian Penyakit

No	Kode Penyakit	Nama Penyakit	Penyebab	Pengendalian
1	P1	Streptococcus	Bakteri Streptococcus agalactiae, S. Iniae	<ol style="list-style-type: none"> 1. Ikan diangkat dan dikarantina, wadah/ kolam dibersihkan 2. Ikan rendam dengan air garam dengan dosis 10ppm selama 10 menit 3. Lalu Rendam selama 30 menit dengan vaksin vaprivac dosis 30ppm 4. Untuk indukan suntik dengan vaksin vaprivac dosis 0.5ppm 5. Lakukan secara rutin 3 x 1 selama 3 hari berturut-turut 6. Selama pengobatan kurangi makan

2	P2	Lerneasis	Parasit lerneae cyprinaceae	<ol style="list-style-type: none"> 1. angkat ikan dan rendam dengan larutan garam 10ppm selama 30menit 2. obati dengan bubuk abate di kolam sebanyak 1.5 ppm, kalau di aquarium 1ppm 3. jika sudah akut, cabut kutu dari badan ikan menggunakan pinset
3	P3	Argulosis	Parasit Argulus sp	<ol style="list-style-type: none"> 1. angakt ikan dan rendam dengan larutan garam 10ppm selama 10 menit 2. kemudian rendam dengan PK 2.5 ppm dengan aerasi yang kuat selama 10 menit
4	P4	Saprolegniasis	Jamur	<ol style="list-style-type: none"> 1. rendam dengan larutan PK sebanyak ½ ppm selama 10 menit

Tabel 3.7 Nilai CF Pakar

Uncertainty Term	CF
Tidak	0
Tidak Tahu	0.2
Mungkin	0.4
Kemungkinan Besar	0.6
Hampir Pasti	0.8
Pasti / Ya	1

Tabel 3.8 Nilai CF User

Ketentuan	CF User
Tidak	0
Ya / Pasti	1

Pada sesi konsultasi sistem, user diberikan jawaban yang masing-masing memiliki bobot sebagai berikut :

Pilihan jawaban “Tidak” = 0

Pilihan jawaban “Ya” = 1

Tabel 3.9 Persentase Kesimpulan

No	Tingkat Presentase	Nilai Keyakinan
1	0% - 50%	Sedikit Kemungkinan atau Kemungkinan Kecil
2	51% - 79%	Pasti
3	80% - 99%	Kemungkinan Besar
4	100%	Sangat Yakin

Contoh Kasus :

Seekor gurami memiliki gejala, kemudian dilanjutkan dengan nilai bobot pengguna, misalnya pengguna memilih jawaban sebagai berikut :

1. Nafsu makan menurun = YA = 1
2. Menggosok-gosokkan badan = YA = 1
3. Luka berlumut = Tidak = 0
4. Pendarahan dan bernanah pada tubuh ikan = YA = 1
5. Terdapat cacing yang menempel pada tubuh
ikan Seperti panah = YA = 1

kemudian hitung nilai CF(pakar) dengan mengalikannya dengan CF(user)

Untuk perhitungan nilai CF pada penyakit streptococcus

$$CF_{Gejala1} = CF(Pakar) * CF(User)$$

$$= 0.4 * 1$$

$$= 0.4$$

$$CF_{Gejala2} = CF(Pakar) * CF(User)$$

$$= 0 * 1$$

$$= 0$$

$$CF_{Gejala3} = CF(Pakar) * CF(User)$$

$$= 0 * 0$$

$$= 0$$

$$CF_{Gejala4} = CF(Pakar) * CF(User)$$

$$= 0 * 1$$

$$= 0$$

$$CF_{Gejala5} = CF(Pakar) * CF(User)$$

$$= 0 * 1$$

$$= 0$$

Mengkombinasikan nilai CF pada penyakit streptococcus

$$CF_{combine1}(Cf_{gejala1}, Cf_{gejala2}) = Cf_{gejala1} + Cf_{gejala2} * (1 - Cf_{gejala1})$$

$$= 0.4 + 0 (1 - 0.4)$$

$$Cf_{old1} = 0.4$$

$$CF_{combine2}(Cf_{old1}, Cf_{gejala3}) = Cf_{old1} + Cf_{gejala3} * (1 - Cf_{old1})$$

$$= 0.4 + 0 (1 - 0.4)$$

$$Cf_{old2} = 0.4$$

$$CF_{combine3}(Cf_{old2}, Cf_{gejala4}) = Cf_{old2} + Cf_{gejala4} * (1 - Cf_{old2})$$

$$= 0.4 + 0 (1 - 0.4)$$

$$Cf_{old3} = 0.4$$

$$CF_{combine4}(Cf_{old3}, Cf_{gejala5}) = Cf_{old3} + Cf_{gejala5} * (1 - Cf_{gold3})$$

$$= 0.4 + 0 (1 - 0.4)$$

$$Cf_{old4} = 0.4$$

Jadi nilai CF dari penyakit streptococcus adalah

$$\text{Presentase keyakinan} = CF * 100\%$$

$$= 0.4 * 100\% = 40\%$$

Untuk perhitungan nilai CF pada penyakit lerniasis

$$CF_{Gejala1} = CF(\text{Pakar}) * CF(\text{User})$$

$$= 0 * 1$$

$$= 0$$

$$CF_{Gejala2} = CF(\text{Pakar}) * CF(\text{User})$$

$$= 0.4 * 1$$

$$= 0.4$$

$$CF_{Gejala3} = CF(\text{Pakar}) * CF(\text{User})$$

$$= 0.4 * 0$$

$$= 0$$

$$CF_{Gejala4} = CF(\text{Pakar}) * CF(\text{User})$$

$$= 0.8 * 1$$

$$= 0.8$$

$$CF_{Gejala5} = CF(\text{Pakar}) * CF(\text{User})$$

$$= 0.8 * 1$$

$$= 0.8$$

Mengkombinasikan nilai CF pada penyakit lerniasis

$$\begin{aligned} CF_{combine1}(Cf_{gejala1}, Cf_{gejala2}) &= Cf_{gejala1} + Cf_{gejala2} * (1 - Cf_{gejala1}) \\ &= 0 + 0.4 (1 - 0) \end{aligned}$$

$$Cf_{old1} = 0.4$$

$$\begin{aligned} CF_{combine2}(Cf_{old1}, Cf_{gejala3}) &= Cf_{old1} + Cf_{gejala3} * (1 - Cf_{old1}) \\ &= 0.4 + 0 (1 - 0.4) \end{aligned}$$

$$Cf_{old2} = 0.4$$

$$\begin{aligned} CF_{combine3}(Cf_{old2}, Cf_{gejala4}) &= Cf_{old2} + Cf_{gejala4} * (1 - Cf_{old2}) \\ &= 0.4 + 0.8 (1 - 0.4) \end{aligned}$$

$$Cf_{old3} = 0.88$$

$$\begin{aligned} CF_{combine4}(Cf_{old3}, Cf_{gejala5}) &= Cf_{old3} + Cf_{gejala5} * (1 - Cf_{old3}) \\ &= 0.88 + 0.8 (1 - 0.88) \end{aligned}$$

$$Cf_{old4} = 0.976$$

Jadi nilai CF dari penyakit leishiasis adalah

$$\begin{aligned} \text{Presentase keyakinan} &= CF * 100\% \\ &= 0.976 * 100\% \\ &= 97.6 \% \end{aligned}$$

Untuk perhitungan nilai CF pada penyakit argulus

$$\begin{aligned} CF_{Gejala1} &= CF(\text{Pakar}) * CF(\text{User}) \\ &= 0 * 1 \\ &= 0 \end{aligned}$$

$$\begin{aligned} CF_{Gejala2} &= CF(\text{Pakar}) * CF(\text{User}) \\ &= 0 * 1 \end{aligned}$$

$$= 0$$

$$CF_{Gejala3} = CF(Pakar) * CF(User)$$

$$= 0 * 0$$

$$= 0$$

$$CF_{Gejala4} = CF(Pakar) * CF(User)$$

$$= 0 * 1 = 0$$

$$CF_{Gejala5} = CF(Pakar) * CF(User)$$

$$= 0 * 1$$

$$= 0$$

Mengkombinasikan nilai CF pada penyakit argulus

$$CF_{combine1}(Cf_{gejala1}, Cf_{gejala2}) = Cf_{gejala1} + Cf_{gejala2} * (1 - Cf_{gejala1})$$

$$= 0 + 0 (1 - 0)$$

$$Cf_{old1} = 0$$

$$CF_{combine2}(Cf_{old1}, Cf_{gejala3}) = Cf_{old1} + Cf_{gejala3} * (1 - Cf_{old1})$$

$$= 0 + 0 (1 - 0)$$

$$Cf_{old2} = 0$$

$$CF_{combine3}(Cf_{old2}, Cf_{gejala4}) = Cf_{old2} + Cf_{gejala4} * (1 - Cf_{old2})$$

$$= 0 + 0 (1 - 0)$$

$$Cf_{old3} = 0$$

$$CF_{combine4}(Cf_{old3}, Cf_{gejala5}) = Cf_{old3} + Cf_{gejala5} * (1 - Cf_{old3})$$

$$= 0 + 0 (1 - 0)$$

$$Cf_{old4} = 0$$

Jadi nilai CF dari penyakit argulus adalah

$$\begin{aligned}
 \text{Presentase keyakinan} &= CF * 100\% \\
 &= 0 * 100\% \\
 &= 0\%
 \end{aligned}$$

Untuk perhitungan nilai CF pada penyakit saprolegniasis

$$\begin{aligned}
 CF_{\text{Gejala1}} &= CF(\text{Pakar}) * CF(\text{User}) \\
 &= 0 * 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 CF_{\text{Gejala2}} &= CF(\text{Pakar}) * CF(\text{User}) \\
 &= 0 * 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 CF_{\text{Gejala3}} &= CF(\text{Pakar}) * CF(\text{User}) \\
 &= 0 * 0 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 CF_{\text{Gejala4}} &= CF(\text{Pakar}) * CF(\text{User}) \\
 &= 0 * 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 CF_{\text{Gejala5}} &= CF(\text{Pakar}) * CF(\text{User}) \\
 &= 0 * 1 \\
 &= 0
 \end{aligned}$$

Mengkombinasikan nilai CF pada penyakit saprolegniasis

$$\begin{aligned}
 CF_{\text{combine1}}(Cf_{\text{gejala1}}, Cf_{\text{gejala2}}) &= Cf_{\text{gejala1}} + Cf_{\text{gejala2}} * (1 - Cf_{\text{gejala1}}) \\
 &= 0 + 0 (1 - 0)
 \end{aligned}$$

$$Cf_{\text{old1}} = 0$$

$$\begin{aligned} CF_{combine2}(Cf_{old1}, Cf_{gejala3}) &= Cf_{old1} + Cf_{gejala3} * (1 - Cf_{old1}) \\ &= 0 + 0 (1 - 0) \end{aligned}$$

$$Cf_{old2} = 0$$

$$\begin{aligned} CF_{combine3}(Cf_{old2}, Cf_{gejala4}) &= Cf_{old2} + Cf_{gejala4} * (1 - Cf_{old2}) \\ &= 0 + 0 (1 - 0) \end{aligned}$$

$$Cf_{old3} = 0$$

$$\begin{aligned} CF_{combine4}(Cf_{old3}, Cf_{gejala5}) &= Cf_{old3} + Cf_{gejala5} * (1 - Cf_{old3}) \\ &= 0 + 0 (1 - 0) \end{aligned}$$

$$Cf_{old4} = 0$$

Jadi nilai CF dari penyakit saprolegniasis adalah

$$\begin{aligned} \text{Presentase keyakinan} &= CF * 100\% \\ &= 0 * 100\% \\ &= 0\% \end{aligned}$$

Berdasarkan dari hasil perhitungan CF, maka nilai keyakinan tertinggi terdapat pada Serangan Parasit Lerniasis dengan nilai 0,976 atau 97.6 %. Dari hasil yang diperoleh maka sistem mendignosa bahwa gurami tersebut kemungkinan besar terkena penyakit lerniasis.

3.4.4 Rancangan Sistem

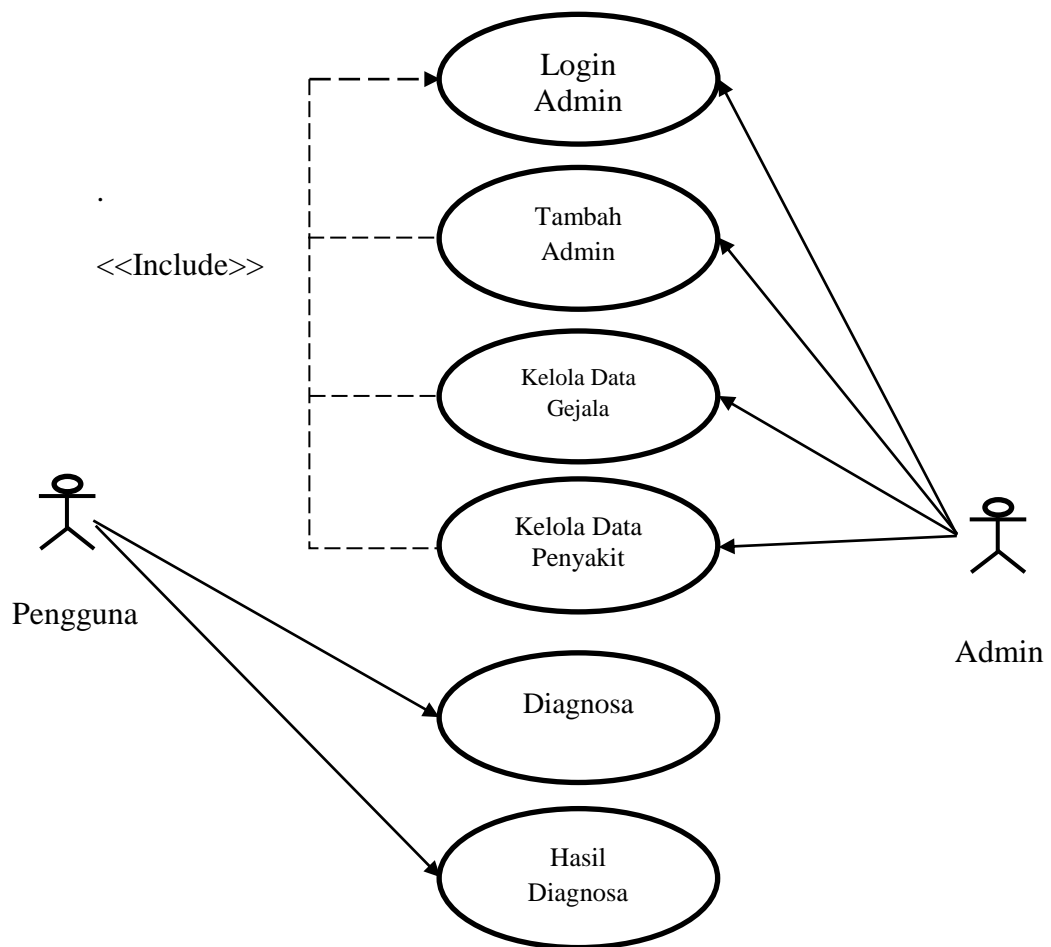
Dari Analisa diatas maka penulis membuat langkah-langkah perancangan sistem pakar untuk memudahkan pembudidaya ikan gurami dalam penentuan penyakit ikan gurami, membantu dalam pembuatan aplikasi pengolahan data, agar lebih mudah memahami alur perancangan sistem ini maka dibuatlah perancangan

secara global meliputi uml, *use case*, *activity diagram*, dan rancangan *database*. Kemudian perancangan secara detail yaitu desain rancangan *input* dan rancangan *ouput*.

3.4.5 Perancangan Pemodelan Sistem

Berikut ini adalah pemodelan sistem yang akan dirancang dengan tujuan untuk menggambarkan kondisi bagian – bagian yang berperan dalam sistem yang dirancang. Pemodelan sistem yang dilakukan dengan membuat perancangan *use case diagram*, *activity diagram* dan struktur tabel

1. Use case diagram



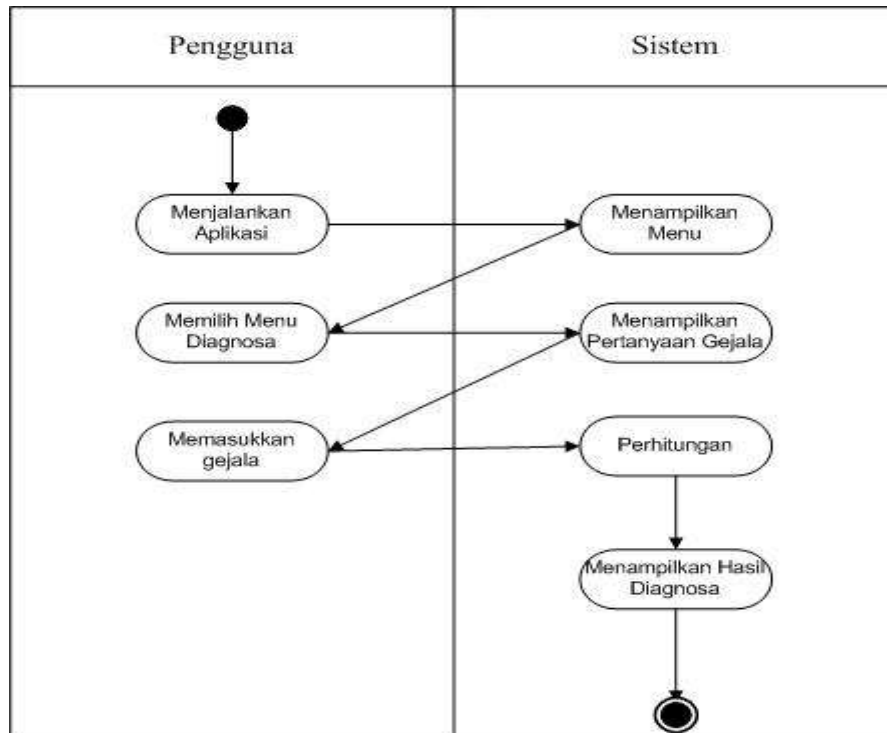
Gambar 3.2 Use Case Diagram

Use case diagram adalah sebuah kegiatan yang dilakukan oleh sistem biasanya menanggapi permintaan dari pengguna sistem

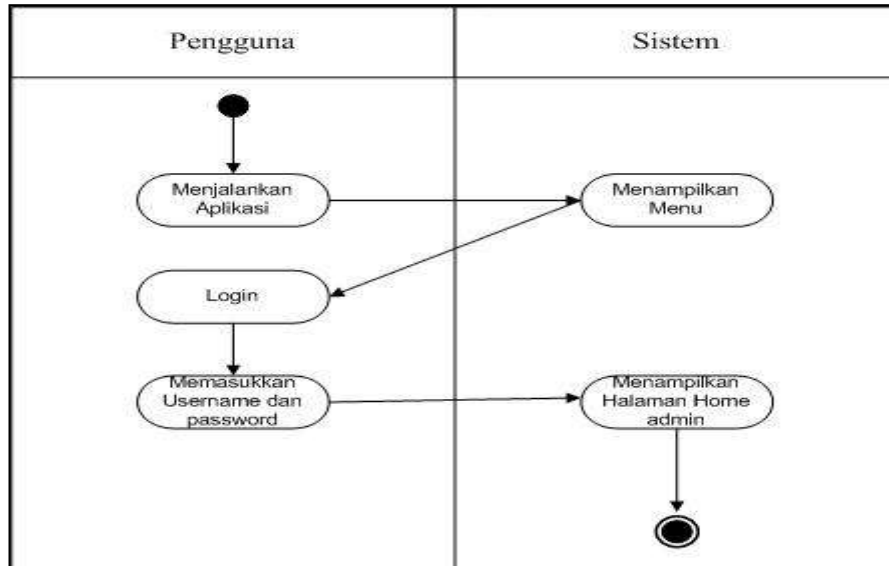
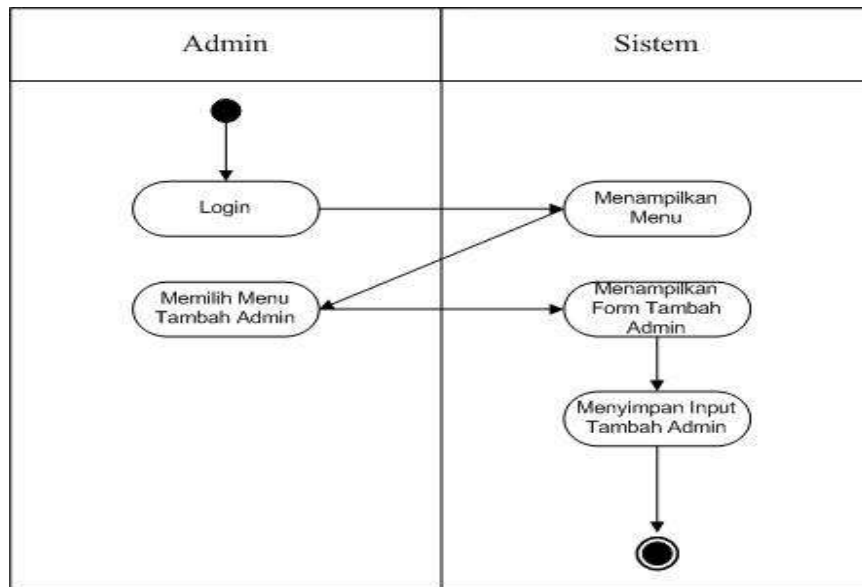
2. Activity Diagram

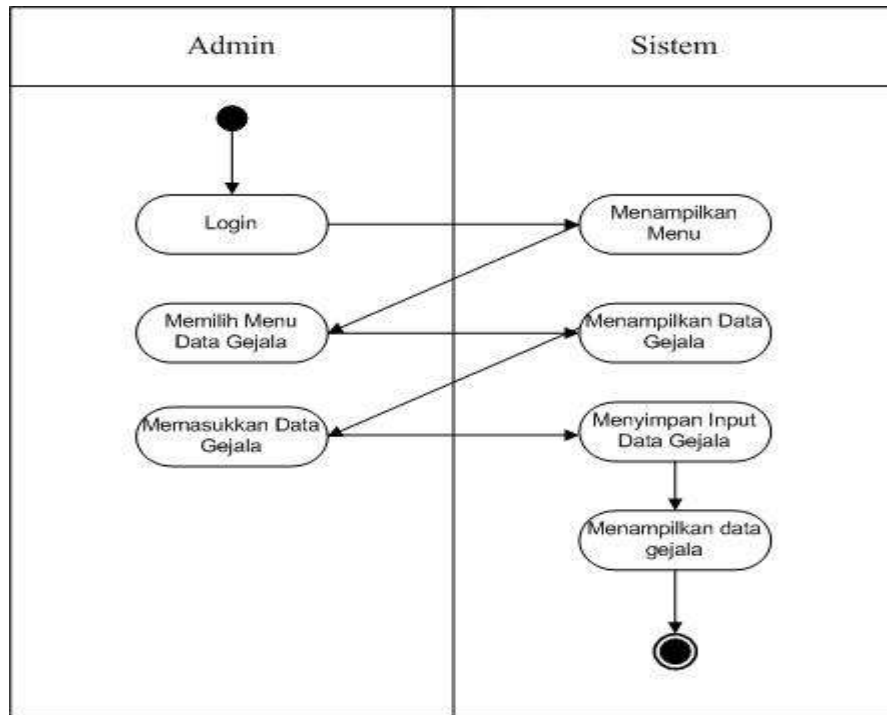
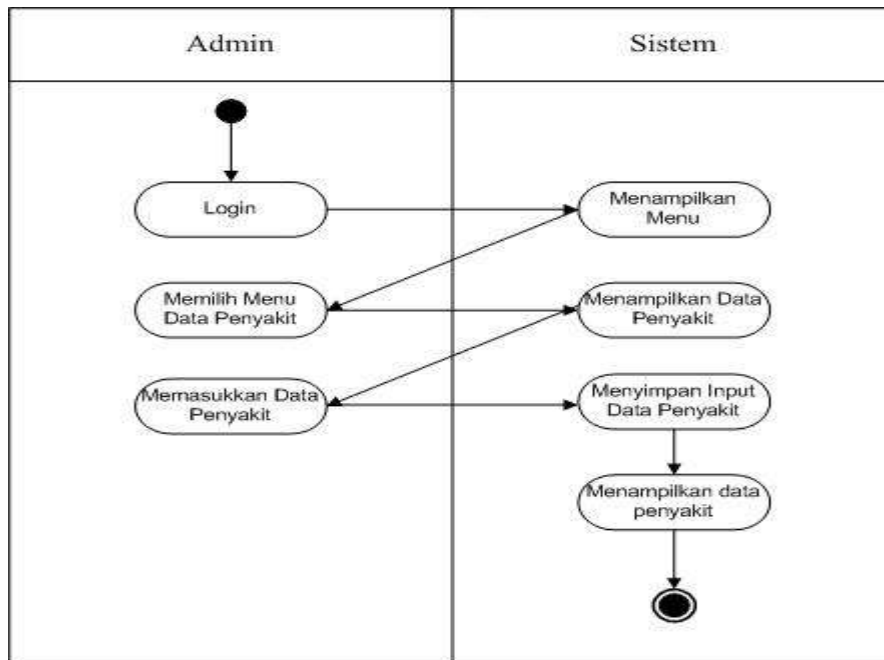
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang yang terdiri darisebagai berikut :

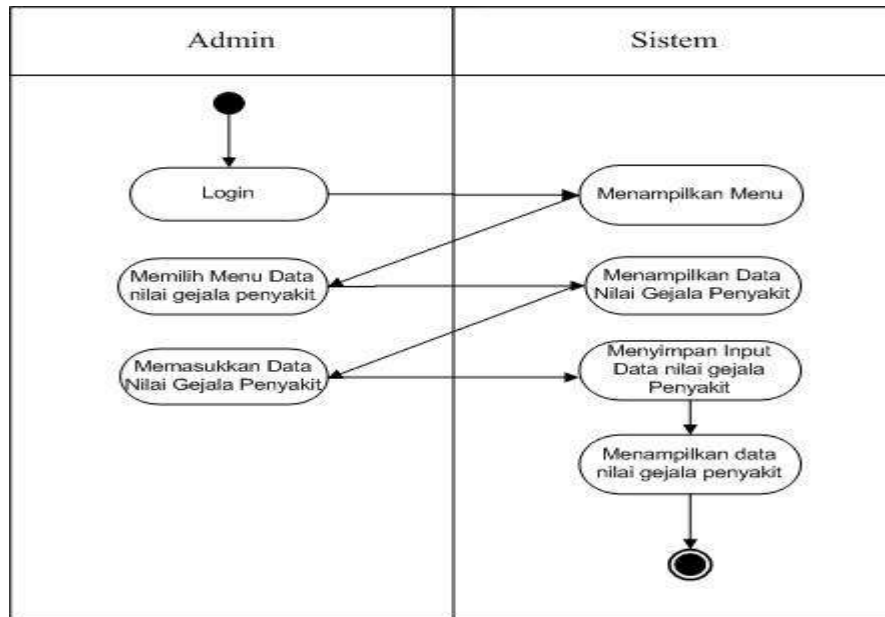
a. Activity diagram Diagnosa



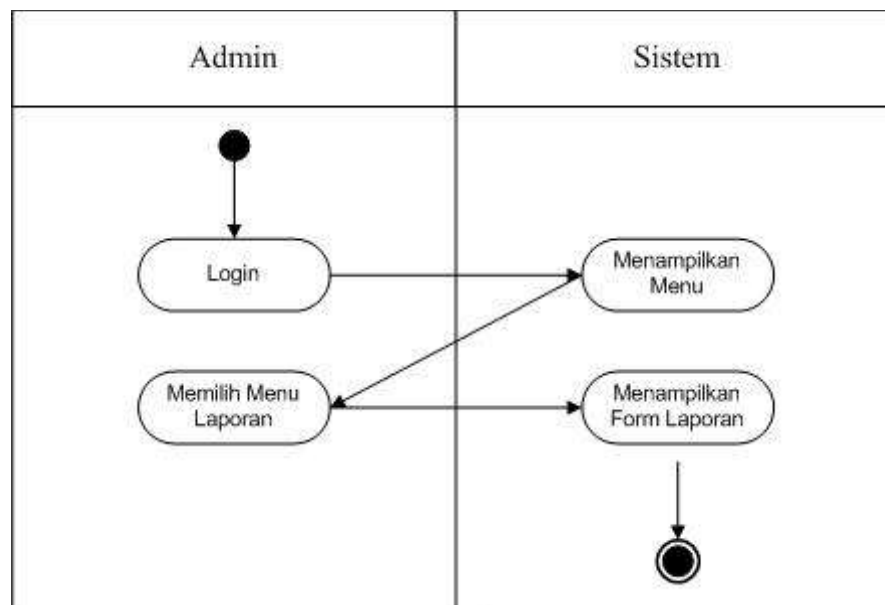
Gambar 3.3 Activity Diagram Diagnosa

b. *Activity diagram Login*Gambar 3.4 *Activity Diagram Login*c. *Activity diagram Tambah Admin*Gambar 3.5 *Activity diagram Tambah Admin*

d. *Activity diagram Data Gejala*Gambar 3.6 *Activity diagram Data Gejala*e. *Activity diagram Data Penyakit*Gambar 3.7 *Activity diagram Data Penyakit*

f. *Activity diagram Data Nilai Gejala Penyakit*

Gambar 3.8 Activity diagram Data Nilai Gejala Penyakit

g. *Activity diagram Laporan*

Gambar 3.9 Activity Diagram Laporan

3. Struktur table

Struktur table adalah penggambaran tentang file – file dalam table sehingga dapat dilihat bentuk-bentuk file tersebut baik field – fieldnya, tipe datanya serta ukuran data tersebut. Adapun struktur table yang ada pada database MySQL dari system pakar diagnosa penyakit bibit ikan gurami yang akan dibuat dapat digambarkan sebagai berikut.

Tabel 3.10 Admin

No	Nama Field	Type Data	Size	Keterangan
1	ID_Admin	Int	11	<i>Primari key</i>
2	Username	Varchar	50	
3	Password	Varchar	50	

Tabel 3.11 Gejala

No	Nama Field	Type Data	Size	Keterangan
1	ID_Gejala	Int	11	<i>Primary key</i>
2	Kode_gejala	Varchar	10	
3	Nm_gejala	Varchar	50	

Tabel 3.11 Penyakit

No	Nama Field	Type Data	Size	Keterangan
1	ID_Penyakit	Int	11	<i>Primary key</i>
2	Kode_penyakit	Varchar	10	
3	Nm_penyakit	Varchar	50	

4	Penyebab	Varchar	50	
5	Solusi	Text	0	

Tabel 3.12 Gejala Penyakit

No	Nama Field	Type Data	Size	Keterangan
1	ID_gejalapenyakit	Int	11	<i>Primary key</i>
2	Id_Gejala	Int	11	
3	Id_Penyakit	Int	11	
4	Nilai	Varchar	50	

Tabel 3.13 Diagnosa

No	Nama Field	Type Data	Size	Keterangan
1	ID_Diagnosa	Int	11	<i>Primary key</i>
2	Nama	Varchar	50	
3	Alamat	Varchar	50	
4	Data_Gejala	Text	0	
5	Hasil	Varchar	50	
6	Id_Penyakit	Int	10	
7	Tanggal	Datename	10	

3.4.6 Rancangan *Interface*

Perancangan antarmuka merupakan bagian yang paling penting dalam merancang sistem. adapun bentuk rancangan antarmuka pada sistem pakar diagnosa penyakit bibit ikan gurami dengan menggunakan metode *certainty factor* di UPTD. Balai Benih Ikan Kota Binjai adalah sebagai berikut

1. Tampilan Awal

The screenshot shows a web interface layout. At the top is a 'Header' section containing four navigation buttons: 'Home', 'Diagnosa', 'About Us', and 'Login Admin'. Below the header is a large rectangular area labeled 'Text'. Underneath the text area are four square placeholder boxes, each labeled 'Gambar'. At the bottom of the page is a 'Footer' section.

Gambar 3.10 Rancangan Tampilan Awal

2. Menu Diagnosa

The screenshot shows the 'Diagnosa Penyakit Ikan Gurami' menu. It features a 'Header' with navigation links: 'Home', 'Diagnosa', 'About Us', and 'Login Admin'. The main content area is titled 'Diagnosa Penyakit Ikan Gurami' and includes two input fields for 'Nama' and 'Alamat'. Below these is a table for symptom questions:

Pertanyaan Gejala	Ya	Tidak
...		
....		
.....		
.....		

At the bottom right of the main content area is a 'Proses' button. The page concludes with a 'Footer' section.

Gambar 3.11 Rancangan Menu Diagnosa

3. Form Hasil Diagnosa

Header	
Home Diagnosa About Us Login Admin	
HASIL DIAGNOSA	
Nama	:
Alamat	:
Tanggal Konsultasi	:
Berikut Adalah Nilai CF / Keyakinan Penyakit dari Gejala Yang Anda Inputkan	
Gurami Anda Mengidap Penyakit	: <input type="text"/>
Tingkat Keyakinan	: <input type="text"/>
Penyebab	: <input type="text"/>
Solusi	: <input type="text"/>
<input type="button" value="Cetak"/>	
Footer	

Gambar 3.12 Rancangan Form Hasil Diagnosa

4. Menu About Us

Header	
Home Diagnosa About Us Login Admin	
About Us	
<div style="border: 1px solid black; width: 80%; margin: 0 auto; padding: 20px;"> <p style="text-align: center;">Text</p> </div>	
Footer	

Gambar 3.13 Rancangan Menu About Us

5. Login Admin

Header			
Home	Diagnosa	About Us	Login Admin


Username

Password

Footer			
--------	--	--	--

Gambar 3.14 Rancangan Form Login Admin

6. Home Admin

Header			
Home	Input Data 	Laporan	Log Out

Tambah Admin

Data Gejala

Data Penyakit

Nilai Gejala Penyakit

Footer			
--------	--	--	--

Gambar 3.15 Rancangan Home Admin

7. Tambah Admin

Header																							
Home	Input Data	Laporan	Log Out																				
<div style="border: 1px solid black; padding: 10px; margin: 20px auto; width: 80%;"> <p style="text-align: center;">Tambah Admin</p> <p>Username <input style="width: 100%;" type="text"/></p> <p>Password <input style="width: 100%;" type="password"/></p> <p style="text-align: right;"><input type="button" value="Simpan"/></p> </div>																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Username</th> <th>Password</th> <th>Edit</th> <th>Hapus</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>				Username	Password	Edit	Hapus																
Username	Password	Edit	Hapus																				
Footer																							

Gambar 3.16 Rancangan Form Admin

8. Data Gejala

Header																		
Home	Input Data	Laporan	Log Out															
<div style="border: 1px solid black; padding: 10px; margin: 20px auto; width: 80%;"> <p style="text-align: center;">Input Data Gejala</p> <p>Kode Gejala : <input style="width: 50%;" type="text"/></p> <p>Nama Gejala : <input style="width: 80%;" type="text"/></p> <p style="text-align: right;"><input type="button" value="Simpan"/></p> </div>																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Kode Gejala</th> <th>Nama Gejala</th> <th>Action</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td>Edit/hapus</td></tr> <tr><td> </td><td> </td><td>Edit/hapus</td></tr> <tr><td> </td><td> </td><td>Edit/hapus</td></tr> <tr><td> </td><td> </td><td>Edit/hapus</td></tr> </tbody> </table>				Kode Gejala	Nama Gejala	Action			Edit/hapus			Edit/hapus			Edit/hapus			Edit/hapus
Kode Gejala	Nama Gejala	Action																
		Edit/hapus																
		Edit/hapus																
		Edit/hapus																
		Edit/hapus																
Footer																		

Gambar 3.17 Perancangan Menu Data Gejala

9. Data Penyakit

Header				
Home		Input Data	Laporan	Log Out
Input Data Penyakit				
Kode Penyakit	:	<input type="text"/>		
Nama Penyakit	:	<input type="text"/>		
Penyebab	:	<input type="text"/>		
Solusi	:	<input type="text"/>		
<input type="button" value="Simpan"/>				
Kode Penyakit	Nama Penyakit	Penyebab	Solusi	Action
				Edit/hapus
				Edit/hapus
				Edit/hapus
				Edit/hapus
Footer				

Gambar 3.18 Rancangan Menu Data Penyakit

10. Form Nilai Gejala Penyakit

Header					
Home		Input Data	Laporan	Log Out	
Input Data Nilai Gejala Penyakit					
Kode Gejala	Nama Gejala	Streptococcus	Lernaeasis	Argulus	Saprolegniasis
<input type="button" value="Simpan"/>					
Footer					

Gambar 3.19 Form Nilai Gejala Penyakit

11. Laporan

Header				
Home	Input Data	Laporan	Log Out	
Laporan Pengguna				
Tanggal	Nama Pengguna	Alamat	Diagnosa Penyakit	Nilai CF
<input type="button" value="Cetak"/>				
Footer				

Gambar 3.20 Rancangan Form Laporan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Aplikasi dan Pembahasan

Pada tahapan ini berisi tentang hasil pengujian program ketika dalam keadaan dijalankan

4.1.1 Halaman Menu Utama

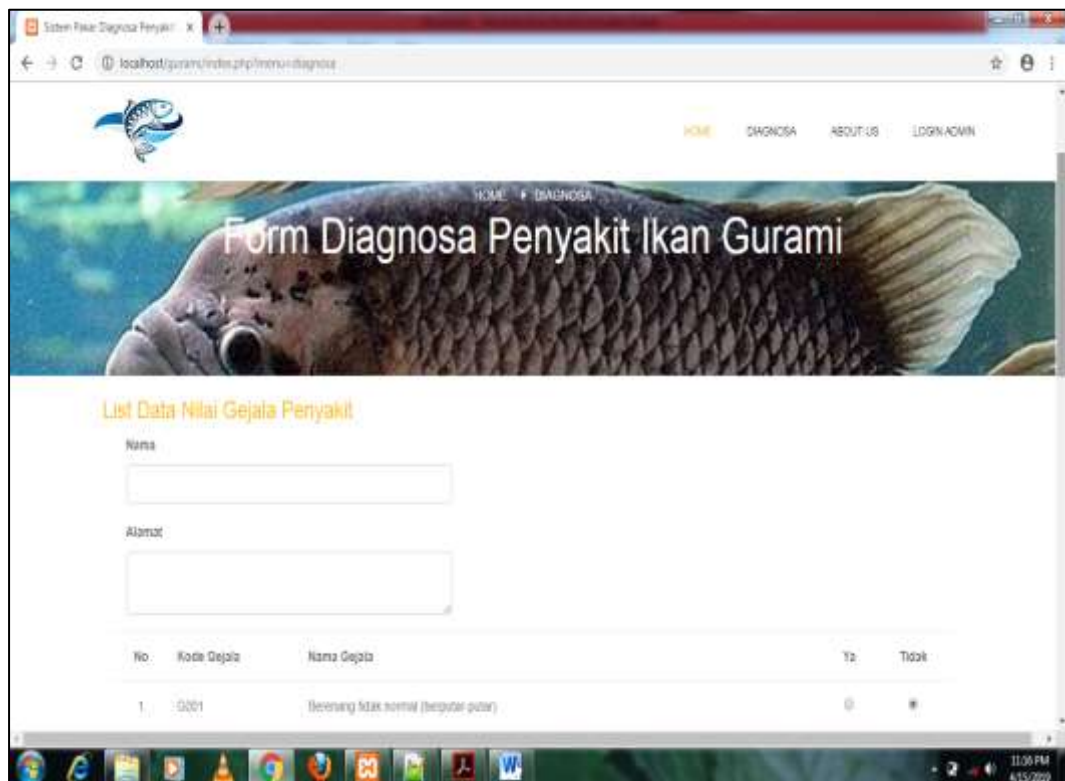
Pada halaman utama sistem pakar diagnosa penyakit ikan gurami menggunakan metode *certainty factor*, pengguna dapat melihat menu pada bagian atas kanan yaitu terdapat menu diagnosa, *about us*, dan login admin yang dapat dipilih oleh pengguna



Gambar 4.1 Tampilan Halaman Utama Pengguna

4.1.2 Halaman Menu Diagnosa

Pada halaman diagnosa pada sistem pakar diagnosa penyakit ikan QS2Wgurami dengan menggunakan *Certainty Factor* (CF), *user* dapat melakukan diagnosa perhitungan penyakit dengan cara memilih gejala-gejala yang tertera di dalam halaman. Selanjutnya *user* diminta memilih gejala-gejala apa saja yang dipilih kemudian di hitung sampai semua data telah dipilih, kemudian tunggu sampai semua proses perhitungan selesai. Selanjutnya data yang telah dihitung akan ditampilkan pada sebuah form hasil diagnosa dan *user* dapat mencetak data tersebut dengan mengklik tombol cetak



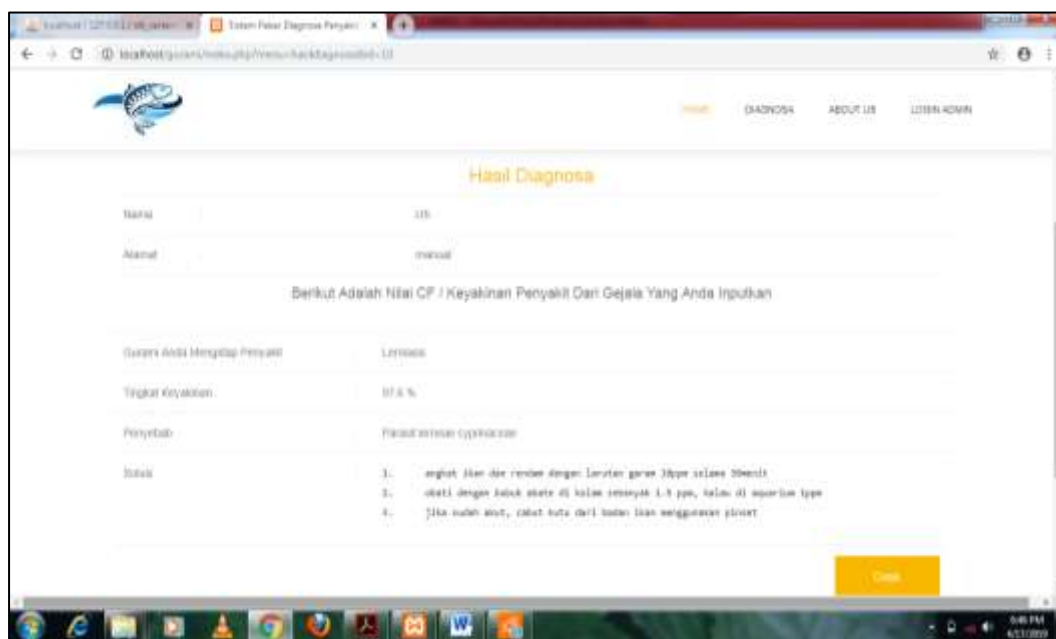
Gambar 4.2 Tampilan Menu Diagnosa

4.1.3 Halaman Hasil Diagnosa

Berikut ini adalah pengujian aplikasi penerapan metode *certainty factor* untuk mendiagnosa penyakit ikan gurami, dalam pengujian berikut ini pengguna memilih gejala penyakit yaitu

1. Nafsu makan menurun
2. Menggosok-gosokkan badan
3. Pendarahan dan bernanah pada tubuh ikan
4. Terdapat cacing yang menempel pada tubuh seperti panah

Setelah dilakukan proses dan dilakukan perhitungan oleh sistem di dapat hasil diagnosanya yaitu ikan gurami terkena penyakit Lerniasis dengan perhitungan nilai CF sebesar 97.6%, hasil dari pengujian ini sudah sesuai dengan hasil hitungan manual yang sudah dilakukan di bab sebelumnya.

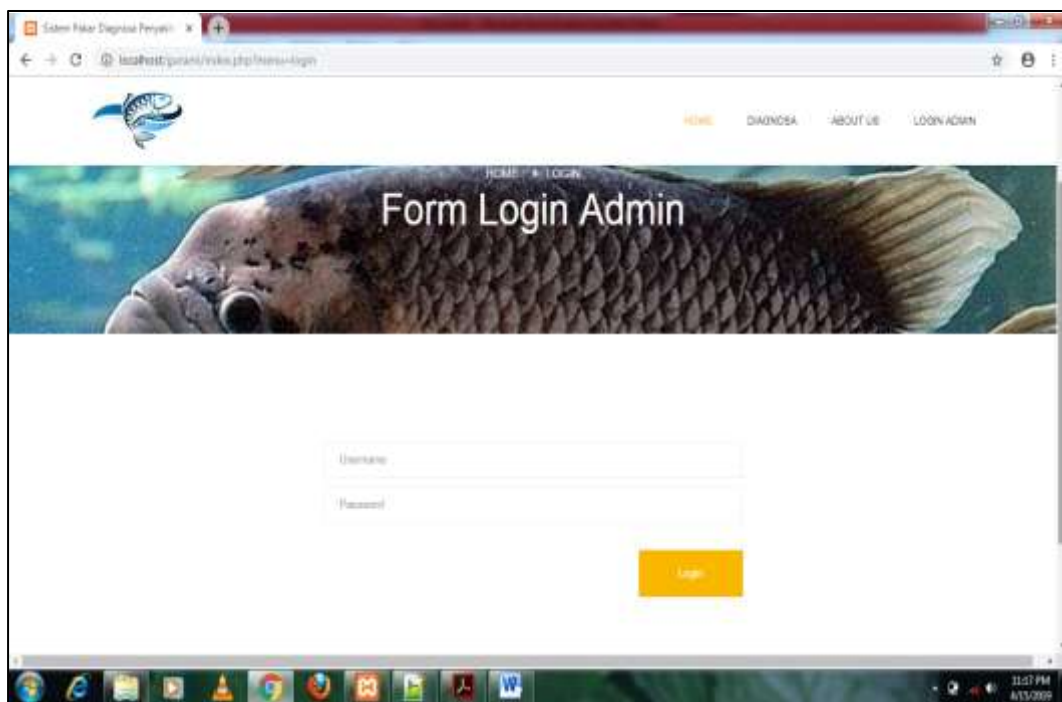


Gambar 4.3 Form Tampilan Hasil Diagnosa

Gambar 4.3 adalah tampilan dari form hasil diagnosa pada sistem pakar diagnosa penyakit ikan gurami menggunakan metode *certainty factor*. pengguna juga dapat mencetak hasil dari diagnosa penyakit ikan gurami mereka dengan mengklik tombol cetak.

4.1.4 Halaman Menu Login Admin

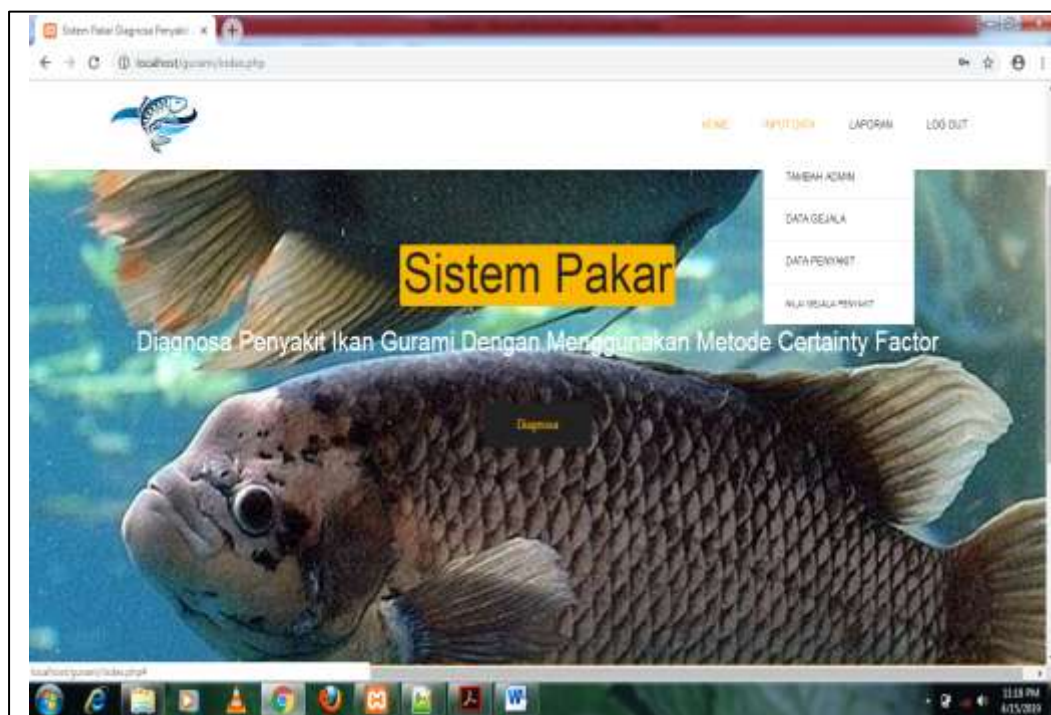
Pada halaman *login* admin, admin dapat mengakses atau masuk kedalam sistem untuk melakukan input data dan mengelola data sistem, terlebih dahulu harus memasukkan *username* dan *password* setelah itu *klik login*.



Gambar 4.4 Halaman Login

4.1.5 Halaman Menu Utama Admin

Pada halaman utama admin dapat dilihat pada bagian atas kanan aplikasi terdapat beberapa menu yang dapat dipilih admin yaitu menu Input data yang terdiri dari sub menu tambah admin, data gejala, data penyakit, data nilai gejala penyakit dan menu laporan



Gambar 4.5 Halaman Utama Admin

4.1.6 Halaman Menu *Input* Data Gejala

Pada halaman *input* data gejala, administrator dapat menambah dan menghapus data gejala pada sistem, pada halaman ini juga terdapat *list* data gejala yang sudah di masukkan

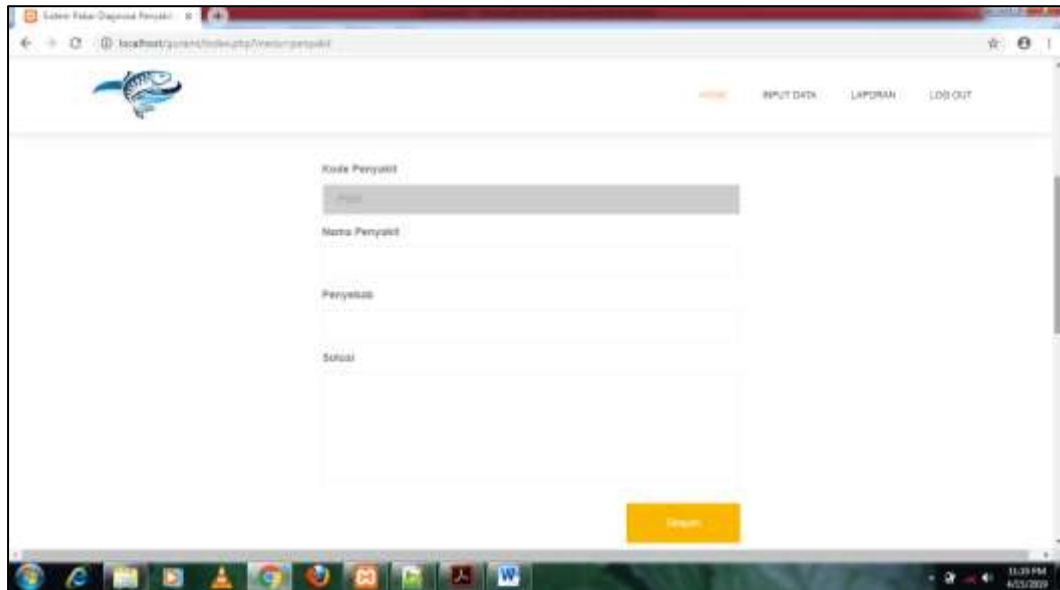
Gambar 4.6 Halaman Input Data Gejala

No	Kode Gejala	Nama Gejala	Akses
1	0001	Berenang tidak normal (berputar-putar)	Edit / Hapus
2	0002	Nafsu makan menurun	Edit / Hapus
3	0003	Ikan malas / terbalak banyak	Edit / Hapus
4	0004	Tubuh berwarna putih	Edit / Hapus
5	0005	Mata menonjol (juga bengkak)	Edit / Hapus
6	0006	Menggosok-gosokkan badan	Edit / Hapus
7	0007	Luka berdarah	Edit / Hapus
8	0008	Pencabutan & berdarah pada tubuh	Edit / Hapus
9	0009	Terdapat cairan yang menempel pada tubuh ikan seperti pasir	Edit / Hapus

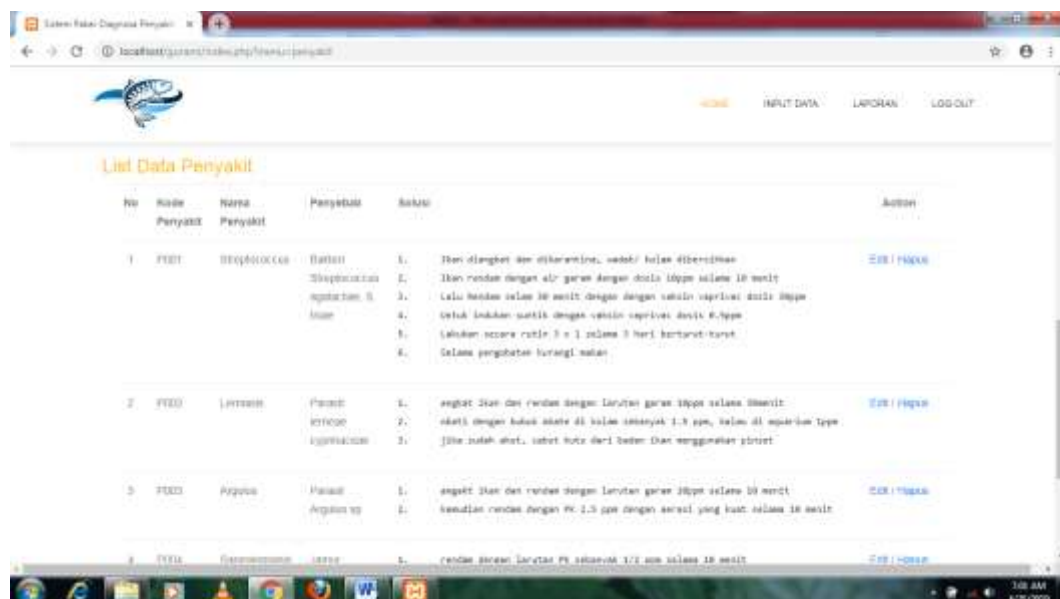
Gambar 4.7 Halaman data gejala

4.1.7 Halaman *Input Data Penyakit*

Pada halaman *input* data gejala, administrator dapat menambah dan menghapus data penyakit pada sistem, pada halaman ini juga terdapat *list* data penyakit yang sudah di masukkan



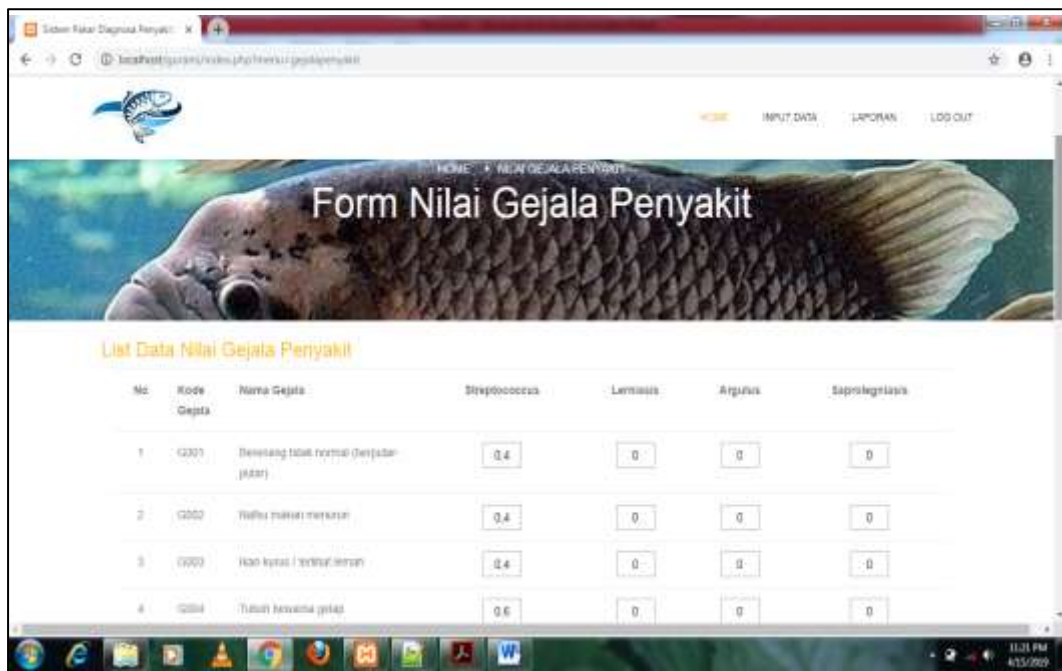
Gambar 4.8 Halaman *Input Data Penyakit*



Gambar 4.9 Halaman *Data Penyakit*

4.1.8 Halaman *Input* Data Nilai Gejala Penyakit

Pada halaman *input* data nilai gejala penyakit, administrator dapat menambah dan menghapus data nilai CF pakar terhadap gejala penyakit, pada halaman ini juga terdapat *list* data nilai gejala penyakit yang sudah di masukkan

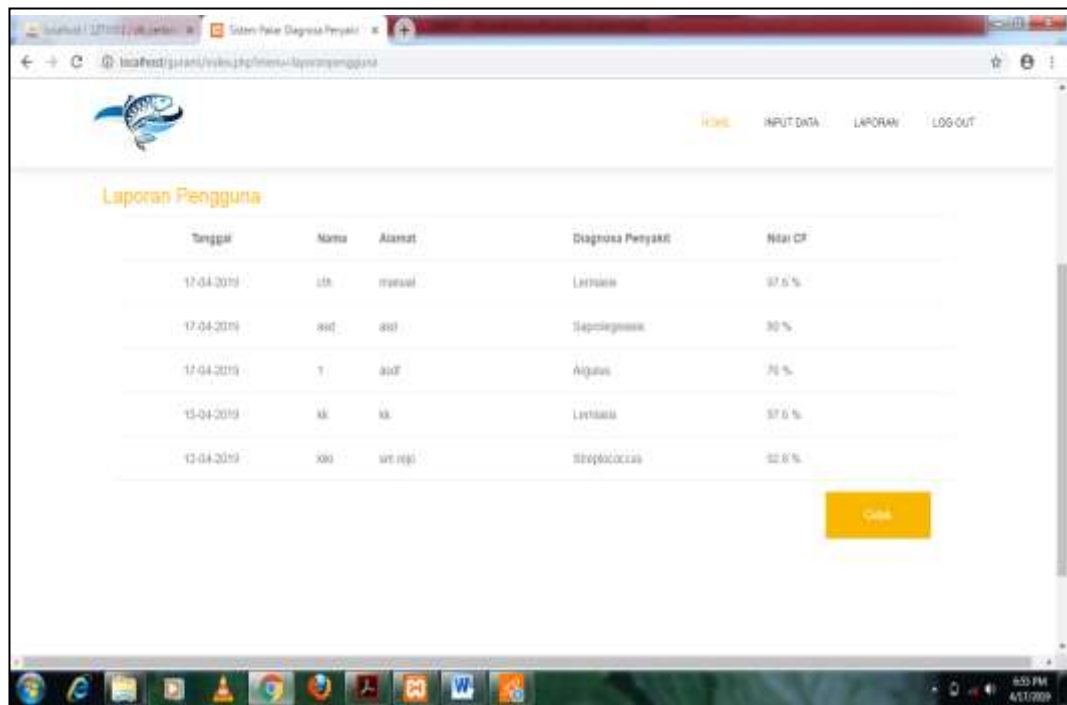


No.	Kode Gejala	Nama Gejala	Streptococcus	Lernasis	Argulus	Saprolegniasis
1	G201	Benjolan tidak normal (benjolan-jalan)	0.4	0	0	0
2	G202	Waktu makan menurun	0.4	0	0	0
3	G203	Kaki kapal / terburuk leman	0.4	0	0	0
4	G204	Tubuh berwarna gelap	0.6	0	0	0

Gambar 4.10 Halaman *input* Data Nilai Gejala Penyakit

4.1.9 Halaman Laporan

Pada halaman laporan sistem pakar diagnosa penyakit ikan gurami dengan menggunakan metode *Certainty Factor*, admin dapat melihat riwayat hasil diagnosa penyakit ikan yang telah dilakukan oleh pengguna dan admin dapat mencetak data tersebut.



Laporan Pengguna

Tanggal	Nama	Alamat	Diagnosa Penyakit	Nilai CF
17-04-2019	Lili	Marsel	Leishish	97.5 %
17-04-2019	Rid	Rid	Septemosa	90 %
17-04-2019	T	Rid	Alpasi	79 %
15-04-2019	M	M	Leishish	97.5 %
12-04-2019	XBO	Mrt rpi	Strepococcus	92.5 %

Cetak

Gambar 4.11 Halaman Laporan

BAB V

PENUTUP

5.1 Kesimpulan

Penerapan Metode *Certainty Factor* Untuk Diagnosa Penyakit Ikan Gurami Berbasis Website dapat dijadikan sebagai salah satu solusi dalam menyelesaikan permasalahan diagnosa penyakit ikan gurami yang berkualitas baik dan menjaga kesehatan ikan, dengan menerapkan beberapa pertanyaan dalam menentukan penyakit ikan gurami. Proses penentuan diagnosa bisa dilakukan lebih akurat dan tepat dibanding dengan hanya pengecekan dan perkiraan sendiri. Kesimpulan yang didapat pada penelitian ini adalah sebagai berikut :

- a. Dalam presentasi gejala penyakit pada ikan gurami memiliki 4 *rule* (Aturan) diagnosa yang selanjutnya dilakukan basis pengetahuan untuk digunakan menentukan penyakit ikan gurami
- b. Penerapan Metode *Certainty Factor* (CF) dapat mempermudah dan memberikan perhitungan penyelesaian seberapa pasti para *user* mengetahui penyakit ikan gurami.

5.2 Saran

Berdasarkan kesimpulan diatas, maka penulis dapat memberikan saran-saran sebagai berikut :

- a. Sistem pakar penentuan penyakit yang dihasilkan setelah proses perhitungan hanya berupa hasil perhitungan berdasarkan beberapa

penyakit saja. Untuk selanjutnya mungkin dapat dikembangkan lebih baik lagi.

- b. Penentuan penyakit ikan gurami harus mempertimbangkan faktor-faktor lain yang mempengaruhi.
- c. Untuk penggunaan metode diharapkan ada perbandingan dengan metode yang lain.
- d. Untuk pengembangan maka program sistem pakar ini dapat dikembangkan ke dalam aplikasi berbasis internet, agar bisa diakses di mana pun dan kapan pun.

DAFTAR PUSTAKA

- Akbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. *Jurnal Teknik dan Informatika*, 5(1), 1-5.
- Annahl Riadi. (2017). Penerapan Metode Certainty Factor Untuk Sistem Pakar Diagnosa Penyakit Diabetes Melitus Pada RSUD Bumi Panua Kabupaten Pohuwato. *ILKOM Jurnal Ilmiah*. 9 (3) 309-316. Diakses dari <https://media.neliti.com/media/publications/258753-penerapan-metode-certainty-factor-untuk>
- Arhami Muhammad. (2013). Konsep Dasar Sistem Pakar. Yogyakarta. Andi
- Desiani Anita, Arhami Muhammad. (2006). Konsep Dasar Kecerdasan Buatan. Elcom. (2013). Belajar Kilat Adobe Dreamweaver CS6. Yogyakarta. Andi
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Hartono, Bambang. (2013). Sistem Informasi Manajemen Berbasis Komputer. Jakarta. Rineka Cipta
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In Seminar Nasional Teknologi Informasi dan Multimedia (pp. 6-7).
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).
Informatika
- Jangkaru Zulkifli. (2007). Memacu Pertumbuhan Gurami (Edisi Revisi). Jakarta. Penebar Swadaya

- Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan*, 11(1), 1-6.
- Kurnia, D. (2017). Analisis QoS Pada Pembagian Bandwidth Dengan Metode Layer 7 Protocol, PCQ, HTB Dan Hotspot Di SMK Swasta Al-Washliyah Pasar Senen. *CESS (Journal of Computer Engineering, System and Science)*, 2(2), 102-111.
- Kurnia, D., Dafitri, H., & Siahaan, A. P. U. (2017). RSA 32-bit Implementation Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 279-284.
- Kusrini. (2008). Aplikasi Sistem Pakar Menentukan Faktor Kepastian Pengguna dengan Metode Kuantitatif Pertanyaan. Yogyakarta C.V Andi Offset.
- M.Shalahuddin, Rosa A.S. (2013). *Rekayasa Perangkat Lunak*. Bandung.
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Marlina, L., Putera, A., Siahaan, U., Kurniawan, H., & Sulistianingsih, I. (2017). Data Compression Using Elias Delta Code. *Int. J. Recent Trends Eng. Res*, 3(8), 210-217.
- Nuri Nia Yanti, slamet budi prayitno, sarjito. (2015). PATOGENISITAS DAN SENSITIVITAS AGENSIA PENYEBAB PENYAKIT BAKTERIAL PADA IKAN GURAMI (*Osphronemus gouramy*) TERHADAP BERBAGAI MACAM OBAT BEREDAR. *Journal of Aquaculture Management and Technology*. 4 (3) 75-83. Diakses dari <http://ejournal-s1.undip.ac.id/index.php/jamt> 75.
- Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research*, 3(8), 196-201.
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol.*, 7(2.13), 345-347.
- Ramadhani, S., Suherman, S., Melvasari, M., & Herdianto, H. (2018). Perancangan Teks Berjalan Online Sebagai Media Informasi Nelayan. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).

- Ruwaida, D., & Kurnia, D. (2018). Rancang Bangun File Transfer Protocol (FTP) dengan Pengamanan Open SSL pada Jaringan VPN Mikrotik di SMK Dwiwarna. *CESS (Journal of Computer Engineering, System and Science)*, 3(1), 45-49.
- Sihotang Tamando Hengki. (2014). Sistem Pakar Mendiagnosa Penyakit Kolesterol Pada Remaja Dengan Metode Certainty Factor (CF) Berbasis Web 15 (1) 16-23. Diakses dari <https://www.researchgate.net/publication/321213380>.
- Winarno Edy, Zaki Ali, SmitDev Community. (2013). *Buku Sakti Pemrograman PHP*. Jakarta. Elex Media Komputindo . Yogyakarta. CV. Andi Offset.
- Yuni Sugiarti. (2013). *Analisis & Perancangan UML Generated (Unified Modelling Language) VB.6*. Yogyakarta. Graha Ilmu