



**IMPLEMENTASI METODE CERTAINTY FACTOR PADA
PENYAKIT MATA BERBASIS ANDROID**

**Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Mengikuti Ujian Akhir
Memperoleh Gelar Sarjana Komputer Pada Fakultas Sains Dan Teknologi
Universitas Pembangunan Panca Budi
Medan**

SKRIPSI

OLEH

**NAMA : NURLAILA SARI
N.P.M : 1514370063
PROGRAM STUDI : SISTEM KOMPUTER**

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

LEMBAR PENGESAHAN

IMPLEMENTASI METODE CERTAINTY FACTOR PADA PENYAKIT MATA DENGAN ANDROID

Disusun Oleh :

NAMA : NURLAILA SARI
N.P.M : 1514370063
PROGRAM STUDI : SISTEM KOMPUTER

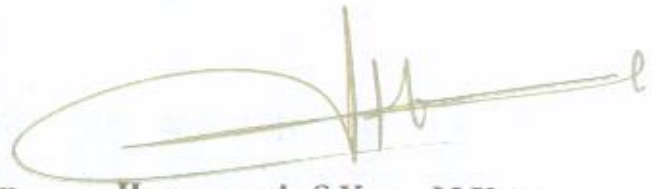
Skripsi telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 28 Agustus 2019 :

Dosen Pembimbing I

Dosen Pembimbing II



Eko Hariyanto, S.Kom., M.Kom



Hermansyah, S.Kom., M.Kom

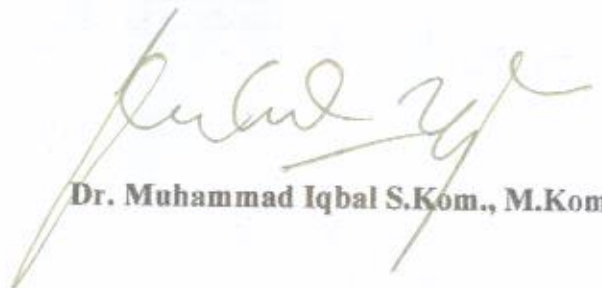
Mengetahui,

Dekan Fakultas Sains Dan Teknologi

Ketua Program Studi Sistem Komputer



Sri Murni Laila, S.T., M.Sc



Dr. Muhammad Iqbal S.Kom., M.Kom

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Nurlaila Sari
NPM : 1514370063
Prodi : Sistem Komputer
Konsentrasi : Keamanan Jaringan Komputer
Judul Skripsi : Implementasi Metode Certainty Factor Pada Penyakit Mata Dengan Android

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil Plagiat
2. Saya tidak akan menuntut perbaikan nilai indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau
3. Skripsi saya dapat dipublikasikan oleh pihak lembaga, dan saya tidak akan menuntut akibat publikasi tersebut

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya, terima kasih

Medan, 28 Agustus 2019



Yang membuat pernyataan

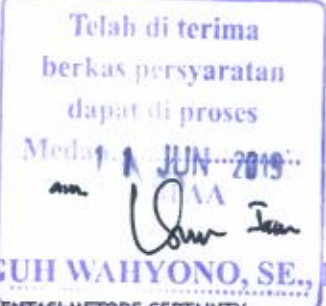
Nurlaila Sari
1514370063



FM-BPAA-2012-041

al : Permohonan Meja Hijau

Medan, 27 Mei 2019
 Kepada Yth : Bapak/Ibu Dekan
 Fakultas SAINS & TEKNOLOGI
 UNPAB Medan
 Di -
 Tempat



Dengan hormat, saya yang bertanda tangan di bawah ini :
 Nama : NURLAILA SARI
 Tempat/Tgl. Lahir : Dusun Ulu / 03-10-1997
 Nama Orang Tua : NURDIN
 N. P. M : 1514370063
 Fakultas : SAINS & TEKNOLOGI
 Program Studi : Sistem Komputer
 No. HP : 082276486338
 Alamat : Jl. Garuda Gg. Palapa No. 11 F

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul IMPLEMENTASI METODE CERTAINTY FACTOR PADA PENYAKIT MATA DENGAN ANDROID., Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 examplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 examplar untuk penguji (bentuk dan warna penjiilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan Ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	250.000
2. [170] Administrasi Wisuda	: Rp.	1,500,000
3. [202] Bebas Pustaka	: Rp.	100.000
4. [221] Bebas LAB	: Rp.	5,000
Total Biaya	: Rp.	1.505.000 1855.000

UKT 7-12 Rp 2.650.000
 Rp 4.505.000

Ukuran Toga : S



Hormat saya
 NURLAILA SARI
 1514370063

Catatan :

- 1. Surat permohonan ini sah dan berlaku bila ;
 - a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
 - b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2. Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.





UNIVERSITAS PEMBANGUNAN PANCA BUDI

FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI TEKNIK ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)

PERMOHONAN MENGAJUKAN JUDUL SKRIPSI

yang bertanda tangan di bawah ini :

Lengkap : NURLAILA SARI
 t/Tgl. Lahir : Dusun Ulu / 03 Oktober 1997
 r Pokok Mahasiswa : 1514370063
 m Studi : Sistem Komputer
 rtrasi : Keamanan Jaringan Komputer
 h Kredit yang telah dicapai : 136 SKS, IPK 3.62
 in ini mengajukan judul skripsi sesuai dengan bidang ilmu, dengan judul:

Judul SKRIPSI	Persetujuan
IMPLEMENTASI METODE CERTAINTY FACTOR PADA PENYAKIT MATA DENGAN ANDROID.	<input checked="" type="checkbox"/> <i>[Signature]</i>
MENDETEKSI MASALAH GANGGUAN PENCERNAAN BERBASIS ANDROID DENGAN MEMANFAATKAN METODE CERTAINTY FACTOR.	<input type="checkbox"/>
SISTEM MENDIAGNOSA PENYAKIT SARAF MENGGUNAKAN METODE FORWARD CHAINING BERBASIS MOBILE.	<input type="checkbox"/>

ini yang disetujui oleh Kepala Program Studi diberikan tanda

[Signature]
 (Ir. Bhakti Alamasyah, M.T., Ph.D.)

Medan, 12 Oktober 2018
 Pemohon,
[Signature]
 (Nurlaila Sari)

Nomor :
 Tanggal : Kamis, 16 Mei 2019
 Disetujui oleh :
 Dekan
[Signature]
 (Sri Shindi Indriyanti, S.T., M.Sc.)
 Tanggal : Kamis, 16 Mei 2019
 Disetujui oleh :
 Ka. Prodi Sistem Komputer
[Signature]
 (MUHAMMAD IQBAL, S.Kom., M.Kom.)

Tanggal : Rabu, 17 Oktober 2018
 Disetujui oleh :
 Dosen Pembimbing I
[Signature]
 Tanggal : Rabu, 17 Oktober 2018
 Disetujui oleh :
 Dosen Pembimbing II
[Signature]

No. Dokumen: FM-LPPM-08-01 Revisi: 02 Tgl. Eff: 20 Des 2015



**UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI**

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

itas : Universitas Pembangunan Panca Budi
 s : SAINS & TEKNOLOGI
 Pembimbing I : Eko Heriyanto S.kom. M.kom
 Pembimbing II : Hermansyah S.kom. M.kom
 Mahasiswa : NURLAILA SARI
 n/Program Studi : Sistem Komputer
 Pokok Mahasiswa : 1514370063
 g Pendidikan : SI
 Tugas Akhir/Skripsi : Implementasi Metode Certainty Factor Pada Penyakit
 Mata dengan Android

ANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
2- 19	* Aca BAB 1	[Signature]	
1/2- 19	* Bimbingan program 1	[Signature]	
1/3- 19	* Bimbingan program 2	[Signature]	
1/4- 19	* Aca program	[Signature]	
1/4- 19	* Aca seminar hasil	[Signature]	
1/4- 19	* Aca sidang meja hijau	[Signature]	
1/7- 19	* Aca jilid Skripsi	[Signature]	

Medan, 08 April 2019
Diketahui/Disetujui oleh :
Dekan



Sri Shindi Indira, S.T., M.Sc.



UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
 Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
 Medan - Indonesia

itas : Universitas Pembangunan Panca Budi
 s : SAINS & TEKNOLOGI
 Pembimbing I : Kana Saputra S.Pd Mkom
 Pembimbing II : Hermansyah S.kom M.kom
 Mahasiswa : NURLAILA SARI
 n/Program Studi : Sistem Komputer
 Pokok Mahasiswa : 1514370063
 g Pendidikan : SI
 tugas Akhir/Skripsi : Implementasi Metode Certainty Factor Pada Penyakit
 Mata dengan Android

NO	ANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
2	18	- Gynpt Bab II	[Signature]	- Ke bab I.
1	19	- Revisi Bab II, tujuh refer litera > 2013	[Signature]	.
1	19	- Revisi Bab II + jensi penyakit mata	[Signature]	
1	19	- Gynpt Bab III a. Demo Program	[Signature]	Acc Bab II
2	19	- Revisi Bab III a Program	[Signature]	
2	19	- Revisi Revisi + Program	[Signature]	
3	19	- Revisi Bab IV.	[Signature]	
3	19	- Laykugi Jener	[Signature]	
3	19	- Acc Jener	[Signature]	
4	19	- Acc Siday	[Signature]	
5	19	- Acc Jilid	[Signature]	

Medan, 09 Januari 2019
 Diketahui/Disetujui oleh
 Dekan,

[Signature]
 Sri Shindi Indira, S.T., M.Sc.

Plagiarism Detector v. 1092 - Originality Report:

Analyzed document: 10/04/2019 08:55:43

"NURLAILA SARI_1514370063_SISTEM KOMPUTER.docx"

Licensed to: Universitas Pembangunan Panca Budi_License4



Relation chart:



Distribution graph:

Comparison Preset: Rewrite. Detected language: Indonesian

Top sources of plagiarism:

% 8	wrds: 808	https://docplayer.info/74132-Sistem-pakar-untuk-mendiagnosa-baby-blues-pada-wanita-dalam-m...
% 8	wrds: 791	https://www.slideshare.net/roniracuda/cf-sistem-pakar-penyakit-tanaman
% 7	wrds: 688	https://docplayer.info/72419250-Sistem-pakar-mendiagnosa-penyakit-tanaman-kelapa-sawit-den...

Show other Sources:]

Processed resources details:

215 - Ok / 31 - Failed

Show other Sources:]

Important notes:

Wikipedia:

Google Books:

Ghostwriting services:

Anti-cheating:



YAYASAN PROF. DR. H. KADIRUN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambang Telp. 061-8455571
Medan - 20122

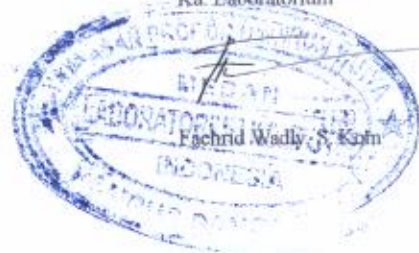
KARTU BEBAS PRAKTIKUM

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama : NURLAILA SARI
N.P.M. : 1514370063
Tingkat/Semester : Akhir
Fakultas : SAINS & TEKNOLOGI
Jurusan/Prodi : Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 10 Juni 2019
Ka. Laboratorium



ABSTRAK

NURLAILA SARI IMPLEMENTASI METODE CERTAINTY PADA PENYAKIT MATA DENGAN ANDROID 2019

Mata adalah salah satu indera yang penting bagi manusia, melalui mata manusia menyerap informasi visual yang digunakan untuk melaksanakan berbagai kegiatan (InfoDATIN, 2014). Mata merupakan salah satu dari panca indra yang memiliki peran yang sangat penting dalam kehidupan manusia yaitu sebagai organ penglihatan. Jika terjadi gangguan pada mata atau penyakit mata, maka akan sangat mengganggu dan jika tidak ditindaklanjuti dapat berakibat sangat fatal bagi kehidupan manusia (Ongko, 2013). Seiring dengan menurunnya kualitas dan gaya hidup seperti pola makan, olahraga, istirahat, bekerja, tingkat stres dan usia, jumlah individu dengan keluhan penyakit mata semakin bertambah. Penyebab utama dari kasus kebutaan ini adalah katarak, kelainan kornea, glaukoma, kelainan refraksi, kelainan retina dan kelainan nutrisi. Situasi tersebut dapat dihindari jika masyarakat memiliki sedikit pengetahuan tentang kesehatan mata. Pada Penelitian ini penulis ingin membuat sistem yang dapat membantu masyarakat dalam mendeteksi jenis penyakit mata yang diderita oleh seseorang. Sistem pakar adalah merupakan salah satu cabang kecerdasan buatan yang mempelajari bagaimana meniru cara berpikir seorang pakar dalam menyelesaikan suatu permasalahan, membuat keputusan maupun mengambil kesimpulan sejumlah fakta. Dimana sistem pakar (expert system) merupakan suatu perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran inferensi menyerupai seorang pakar dalam menyelesaikan masalah. Diharapkan dengan adanya perancangan aplikasi ini, mendeteksi seseorang terserang penyakit mata dapat di diagnosa dan ditangani sedini mungkin. sehingga pengguna diharapkan dapat menjalani kehidupan yang sehat dan lebih produktif.

Kata kunci : Android, Certainty Factor, Penyakit Mata, Sistem Pakar.

KATA PENGANTAR

Assalamua'laikum Warahmatullahi Wabarakatuh.

Alhamdulillah rabbi 'allamin, dengan mengucapkan puji syukur ke Hadirat Allah SWT, karena berkat rahmat dan Hidayah-Nya penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat beriringkan salam mari kita junjung tinggi kan kepada Nabi Muhammad SAW yang telah mengantarkan umatnya dari alam kegelapan dan kebodohan menuju alam yang terang benderang dan penuh dengan ilmu pengetahuan.

Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan Pendidikan Strata Satu (S1) dan memperoleh gelar Sarjana Komputer pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.

Dengan kerendahan hati penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada pihak-pihak yang telah banyak membantu penulis dalam menyelesaikan skripsi ini, diantaranya adalah :

1. Kedua orang tua dan keluarga penulis tercinta yang telah banyak memberikan dorongan, semangat terutama kepada orang tua wali saya berkat bantuan do'a maupun materil sehingga penulis dapat menyelesaikan skripsi ini.

2. Bapak Dr. H. Muhammad Isa Indrawan, SE., MM, selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. Ibu Sri Shindi Indira, ST.,MSc, selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Muhammad Iqbal, S.Kom., M.Kom, selaku Ketua Program Studi Sistem Komputer.
5. Bapak Eko Hariyanto, S.Kom., M.Kom, selaku Dosen Pembimbing I.
6. Bapak Hermansyah, S.Kom., M.Kom, selaku Dosen Pembimbing II yang telah memberikan curahan pengetahuan serta bimbingan dalam penyelesaian skripsi ini.
7. Seluruh Dosen-Dosen Universitas Pembangunan Panca Budi yang telah mengajarkan banyak hal kepada penulis.
8. Seluruh Teman Penulis dari Program Studi Sistem Komputer Fakultas Sains dan Teknologi terutama buat kak Isnani Ulfa Indun yang telah banyak membantu dalam menyelesaikan skripsi ini.

Dengan sepenuh hati penulis memohon maaf atas segala kekurangan dan kelemahan dalam penyusunan skripsi ini baik sistematika penulisan maupun penggunaan bahasa yang tidak sesuai dengan aturan yang telah ditetapkan. Semua itu disebabkan ketidak sengajaan dan kesilapan penulis dalam mengerjakan skripsi ini. Penulis menyadari skripsi ini masih jauh dari kesempurnaan, baik dalam penyajian skripsi, penyajian materi, pembahasan masalah, maupun penyusunan

kata-kata. Untuk itu penulis mengharapkan kritik dan saran dari berbagai yang bersifat membangun dalam penyempurnaan skripsi ini, agar lebih bermanfaat bagi penulis dan bagi kita semua. Semoga skripsi ini dapat bermanfaat bagi semua pihak yang membacanya. Akhir kata penulis ucapkan banyak terima kasih.

Billahi Fi Sabilil Haq.

Wassalamua'laikum Warahmatullahi Wabarakatuh.

Medan, 19 Agustus 2019
Penulis,

NURLAILA SARI
1 5 1 4 3 7 0 0 6 3

DAFTAR ISI

HALAMAN

LEMBAR JUDUL

LEMBAR PENGESAHAN

ABSTRAK

KATA PENGANTAR i

DAFTAR ISI iv

DAFTAR GAMBAR viii

DAFTAR TABEL x

DAFTAR LAMPIRAN xi

BAB I PENDAHULUAN

1.1. Latar Belakang Masalah 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

2.1. Penyakit Mata 5

2.1.1. Pengertian Penyakit Mata 5

2.1.2. Bagian Mata 6

2.1.3. Jenis Penyakit Mata 6

2.1.4. Gejala Penyakit Mata 7

2.2. Sistem Pakar 9

2.3. Metode Certainty Factor	14
2.4. Aplikasi Mobile	17
2.5. Android	18
2.6. Arsitektur Android	19
2.7. Defenisi Visual Studio 2015	20
2.8. Pengertian Basis Data	22
2.9. Database SQ Lite	25
2.10. Unifed Modeling Language (UML)	26
2.10.1. Pengenalan UML	26
2.10.2. Use Case Diagram	26
2.10.3. Activity Diagram	28
2.10.4. Sequence Diagram	29
2.10.5. Class Diagram	30

BAB III METODE PENELITIAN

3.1. Metodologi Penelitian	32
3.1.1. Metode Pengumpulan Data	32
3.1.2. Metode Pengembangan Perangkat Lunak	33
3.2. Analisis Sistem	36
3.3. Fungsional Aplikasi	37
3.4. Analisis Kebutuhan	37
3.5. Rule Aplikasi Metode Certainty Factor Pada Penyakit Mata	39
3.5.1. Gejala Penyakit	39
3.5.2. Penyakit	43

3.5.3. Aturan	53
3.6. Perancangan Sistem	55
3.6.1. Use Case Diagram	55
3.6.2. Use Case	56
3.6.3. Activity Diagram	59
3.6.4. Diagram Sequence.....	61
3.7. Perancangan Aplikasi	63
3.8. Perancangan Class Diagram	67
3.9. Perancangan Database	68
3.9.1. Perancangan Tabel	68

BAB IV HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem Yang Digunakan	69
4.1.1. Spesifikasi Perangkat Keras	69
4.1.2. Spesifikassi Perangkat Lunak.....	70
4.2. Tampilan Aplikasi Metode Certainty Faktor Pada Penyakit Mata.....	70
4.2.1. Tampilan Menu Utama.....	70
4.2.2. Tampilan Penyakit Mata	72
4.2.3. Tampilan Diagnosa Penyakit	72
4.2.4. Tampilan Hasil	73
4.2.5. Tampilan Detail Penyakit.....	74
4.2.6. Tampilan Login	75
4.2.7. Tampilan Menu Admin	76

4.2.8. Tampilan Data Gejala.....	77
4.2.9. Tampilan Data Penyakit	78
4.2.10. Tampilan Data Admin	79
4.2.11. Tampilan Tentang	80
4.3. Pengujian Aplikasi Metode Certainty Factor Pada Penyakit Mata	81

BAB V PENUTUP

5.1. Kesimpulan.....	86
5.2. Saran.....	86

DAFTAR PUSTAKA

BIOGRAFI PENULIS

DAFTAR TABEL

	Halaman
2.1 Nilai CF.....	16
2.2 Simbol Use Case Diagram.....	27
2.3 Simbol Activity Diagram.....	29
2.4 Simbol Sequence Diagram.....	30
2.5 Simbol Class Diagram.....	31
3.1 Gejala.....	39
3.2 Penyakit.....	43
3.3 Aturan.....	53
3.4 Definisi Aktor.....	56
3.5 Definisi Use Case.....	57
3.6 Skenario Use Case Penyakit Mata.....	58
3.7 Skenario Use Case Diagnosa.....	58
3.8 Skenario Use Case Profil Pembuat.....	59
3.9 Gejala.....	68
3.10 Penyakit.....	68

DAFTAR GAMBAR

	Halaman
2.1 Struktur Skematis Sistem Pakar	12
2.2 Diagram Forward Chaining.....	13
2.3 Diagram Backward Chaining.....	14
2.4 Framework Android	20
2.5 Tampilan Visual Studio 2015	21
3.1 Flowchart Penulisan	32
3.2 Paradigma Waterfall (Classic Life Cycle)	34
3.3 Use Case Diagram Aplikasi Certainty Factor Pada Penyakit Mata	56
3.4 Activity Diagram Penyakit Mata.....	59
3.5 Activity Diagram Diagnosa.....	60
3.6 Activity Diagram Profil Pembuat.....	60
3.7 Diagram Sequence Penyakit Mata	61
3.8 Diagram Sequence Diagnosa	62
3.9 Diagram Sequence Profil Pembuat	63
3.10 Rancangan Tampilan Menu Utama	64
3.11 Rancangan Tampilan Penyakit Mata	65
3.12 Rancangan Tampilan Diagnosa	66
3.13 Rancangan Tampilan Menu Tentang Diagram Sequence.....	67

3.14 Class Diagram Aplikasi Pada Penyakit Mata	67
4.1 Tampilan Menu Utama	71
4.2 Tampilan Penyakit Mata	72
4.3 Tampilan Diagnosa Gejala	73
4.4 Tampilan Hasil Diagnosa	74
4.5 Tampilan Detail Penyakit.....	75
4.6 Tampilan Login.....	76
4.7 Tampilan Menu Admin	77
4.8 Tampilan Data Gejala	78
4.9 Tampilan Data Penyakit.....	79
4.10 Tampilan Data Admin.....	80
4.11 Tampilan Tentang	81
4.12 Menu Utama.....	82
4.13 Pilih Gejala.....	83
4.14 Hasil Diagnosa	84
4.15 Detail Penyakit.....	85

DAFTAR LAMPIRAN

	Halaman
1. Listing Program	L-1
2. Pengajuan Judul	L-2
3. Berita Acara.....	L-3
4. Plagiat Cheker	L-4
5. Pengajuan Sidang	L-5
6. Bebas Pratikum.....	L-6

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Mata adalah salah satu indera yang penting bagi manusia, melalui mata manusia menyerap informasi visual yang digunakan untuk melaksanakan berbagai kegiatan (InfoDATIN, 2014). Mata merupakan salah satu dari panca indra yang memiliki peran yang sangat penting dalam kehidupan manusia yaitu sebagai organ penglihatan. Jika terjadi gangguan pada mata atau penyakit mata, maka akan sangat mengganggu dan jika tidak ditindak lanjuti dapat berakibat sangat fatal bagi kehidupan manusia (Ongko, 2013).

Seiring dengan menurunnya kualitas dan gaya hidup seperti pola makan, olahraga, istirahat, bekerja, tingkat stres dan usia, jumlah individu dengan keluhan penyakit mata semakin bertambah. Penyebab utama dari kasus kebutaan ini adalah katarak, kelainan kornea, glaukoma, kelainan refraksi, kelainan retina dan kelainan nutrisi. Situasi tersebut dapat dihindari jika masyarakat memiliki sedikit pengetahuan tentang kesehatan mata. Pada Penelitian ini penulis ingin membuat sistem yang dapat membantu masyarakat dalam mendeteksi jenis penyakit mata yang diderita oleh seseorang.

Sistem pakar adalah merupakan salah satu cabang kecerdasan buatan yang mempelajari bagaimana meniru cara berpikir seorang pakar dalam menyelesaikan suatu permasalahan, membuat keputusan maupun mengambil kesimpulan sejumlah

fakta. Dimana sistem pakar (*expert system*) merupakan suatu perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran inferensi menyerupai seorang pakar dalam menyelesaikan masalah. (Azhar, 2014)

Faktor kepastian (*certainty factor*) menyatakan kepercayaan dalam sebuah kejadian (fakta atau hipotesa) berdasar bukti atau penilaian pakar. *Certainty factor* menggunakan suatu nilai untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. (Sari, 2013) Untuk itu penulis membuat suatu aplikasi yang dapat mendeteksi gejala yang diderita pada mata seseorang dengan menggunakan metode *certainty factor* agar dapat memberikan hasil dari jenis penyakit mata yang diderita. Sistem akan memberikan pertanyaan gejala yang dialami sampai bisa mengidentifikasi suatu objek berdasarkan jawaban yang diterimanya. Perancangan dengan media aplikasi *mobile* ini diperuntukkan bagi semua orang, karena saat ini semua orang cenderung dekat dengan *gadget* dan perkembangan teknologi. Diharapkan dengan adanya perancangan aplikasi ini, mendeteksi seseorang yang terserang penyakit pada mata dapat didiagnosa dan ditangani sedini mungkin. Berdasarkan latar belakang diatas maka penulis tertarik untuk memilih judul **“IMPLEMENTASI METODE *CERTAINTY FACTOR* PADA PENYAKIT MATA DENGAN ANDROID”**.

1.2. Rumusan Masalah

Adapun permasalahan yang dihadapi dalam perancangan sistem ini adalah sebagai berikut :

1. Bagaimana mengetahui jenis penyakit pada mata seseorang dengan menggunakan bantuan aplikasi *mobile*?
2. Bagaimana memanfaatkan metode *certainty factor* dalam mendeteksi jenis penyakit mata pada seseorang berdasarkan gejala yang dialami?

1.3. Batasan Masalah

Dalam perancangan implementasi metode *certainty factor* pada penyakit mata dengan android ini, penulis membatasi masalah sebagai berikut :

1. Aplikasi sistem deteksi penyakit hanya pada jenis penyakit pada mata.
2. Perancangan aplikasi dibuat dengan berbasis *mobile android* dengan menggunakan bahasa pemrograman C# dan *database SQLite*.
3. Metode yang digunakan pada sistem adalah metode *certainty factor* untuk mendeteksi jenis penyakitnya.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai penulis dalam merancang implementasi metode *certainty factor* pada penyakit mata dengan android ini adalah :

1. Menerapkan metode *certainty factor* dalam pembuatan aplikasi pada *smartphone android*.
2. Dengan menggunakan sistem ini dapat mengetahui manfaat sistem pakar dengan cara mengetahui jenis penyakit mata yang dialami.

1.5. Manfaat Penelitian

Perancangan implementasi metode *certainty factor* pada penyakit mata dengan android ini bermanfaat bagi penulis dan masyarakat luas antara lain :

1. Dapat mengetahui jenis penyakit mata yang diderita oleh seseorang dengan mudah menggunakan bantuan aplikasi yang berbasis android sebelum mengkonsultasikan ke yang lebih ahli.
2. Agar dapat segera menganggunangi penyakit mata yang diderita sedini mungkin sehingga terhindarnya dari penyakit yang lebih parah.

BAB II

LANDASAN TEORI

2.1. Penyakit Mata

2.1.1. Pengertian Penyakit Mata

Mata merupakan salah satu dari panca indra yang memiliki peran yang sangat penting dalam kehidupan manusia yaitu sebagai organ penglihatan. Jika terjadi gangguan pada mata atau penyakit mata, maka akan sangat mengganggu dan jika tidak ditindaklanjuti dapat berakibat sangat fatal bagi kehidupan manusia. (Ongko, 2013).

Mata adalah organ penglihatan yang mendeteksi cahaya. meneruskan sinyal tersebut ke retina, dan membuat efek visual yang dikirim ke otak. Secara sederhana mata hanya mengetahui apakah lingkungan sekitarnya adalah terang atau gelap. (Kurniawan, 2011).

2.1.2. Bagian Mata

Terdapat beberapa bagian organ mata yang bekerjasama mengantarkan cahaya dari sumbernya menuju ke otak untuk dapat dicerna oleh sistem saraf manusia. Bagianbagian tersebut adalah: (Kurniawan, 2011).

1. Kornea

Merupakan bagian terluar dari bola mata yang menerima cahaya dari sumber cahaya.

2. Pupil dan Iris atau Selaput Pelangi

Dari kornea, cahaya akan diteruskan ke pupil. Pupil menentukan kuantitas cahaya yang masuk ke bagian mata yang lebih dalam. Pupil mata akan melebar jika kondisi ruangan yang gelap, dan akan menyempit jika kondisi ruangan terang. Lebar pupil dipengaruhi oleh iris di sekelilingnya. Iris berfungsi sebagai diafragma. Iris terlihat sebagai bagian yang berwarna pada mata.

3. Lensa mata

Lensa mata menerima cahaya dari pupil dan meneruskannya pada retina. Fungsi lensa mata adalah mengatur fokus cahaya, sehingga cahaya jatuh tepat pada bintik kuning retina. Untuk melihat objek yang jauh (cahaya datang dari jauh), lensa mata akan menipis. Sedangkan untuk melihat objek yang dekat (cahaya datang dari dekat), lensa mata akan menebal.

4. Retina atau Selaput Jala

Retina adalah bagian mata yang paling peka terhadap cahaya, khususnya bagian retina yang disebut bintik kuning. Setelah memasuki retina, cahaya diteruskan ke saraf optik.

5. Saraf

Saraf optik yang memasuki sel tali dan kerucut dalam retina, untuk menuju ke otak.

2.1.3. Jenis Penyakit Mata

Penyakit mata merupakan penyakit dengan jumlah penderita yang terus meningkat setiap tahunnya di Indonesia. Prevalensi angka kebutaan di Indonesia

berkisar 1,2% dari jumlah penduduk. Penyebab utama dari kasus kebutaan ini adalah katarak, kelainan kornea, glaukoma, kelainan refraksi, kelainan retina dan kelainan nutrisi. (Setiawan, 2014) .

Seperti pada organ tubuh yang lain, mata juga selalu mengalami gangguan atau disebut juga penyakit pada mata. Banyak jenis penyakit mata yang sering muncul disaat melakukan aktifitas sehari-hari. Hal ini terkait karena organ mata yang lebih kompleks dari organ tubuh lainnya dan juga karena beberapa faktor pemicu penyakit mata. Terdapat beberapa penyakit mata yang umum diderita manusia, di antaranya adalah: *Konjungtivitis, Episkleritis, Keratitis, Skleritis, Hordeolum, Uveitis*. (Kurniawan, 2011).

2.1.4. Gejala Penyakit Mata

Adapun gejala penyakit pada mata adalah sebagai berikut :

Gejala
Mata nyeri hebat
Mata menonjol
Penglihatan kabur
Peka terhadap cahaya
Mata merah
Mata berair
Mata perih
Mata gatal
Kelopak Mata membengkak

Mata ungu
Mata sakit
Air mata berlebihan
Mata tegang
Mata meradang
Mata kering
Mata iritasi
Mata nyeri bila ditekan
Demam
Menekan kedipan berlebihan
Sel batang retina sulit beradaptasi diruang yang remang-remang
Pada siang hari penglihatan menurun
Tidak dapat melihat pada lingkungan yang kurang bercahaya
Pergerakan mata terbatas
Mata tampak mengkilat
Bola mata membengkak dan tampak berkabut
Sumber cahaya akan berwarna pelangi bila memandang lampu neon
Penglihatan yang tadinya kabur lama-kelamaan menjadi normal
Malam hari kesulitan melihat
Siang hari ketajaman mata menurun
Mata silau akan cahaya
Sering ganti kacamata
Penglihatan ganda pada salah satu sisi mata
Lensa mata membengkak

Berbentuk keropeng pada kelopak mata ketika bangun pada siang hari
Penglihatan menurun pada ruang gelap
Penglihatan menurun pada malam hari
Keluarnya cairan kotoran dari mata
Berbaliknya bulu mata
Pembengkakan kelenjar getah bening didepan telinga
Munculnya garis parutan pada kornea
Komplikasi pada,telinga,hidung dan tenggorokan
Mata mempersempit, perubahan bentuk
Benjolan pada mata bagian atas atau bawah
Gangguan penglihatan pada salah satu mata
Garis mata lurus terlihat bergelombang
Mata tidak nyeri
Mata melihat melayang-layang
Mata melihat kilatan cahaya
Seperti ada benda asing di mata
Sakit kepala
Riwayat penyakit menular seksual pada ibu
Mata membengkak

2.2. Sistem Pakar

Sistem pakar (*expert system*) berasal dari istilah sistem pakar berbasis pengetahuan. Sistem pakar adalah suatu sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang

biasanya memerlukan keahlian manusia. Sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Sistem pakar merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem ini bekerja untuk mengadopsi pengetahuan manusia ke komputer yang menggabungkan dasar pengetahuan untuk menggantikan seorang pakar dalam menyelesaikan suatu masalah. (Harto, 2013).

Sistem pakar berasal dari istilah *knowledge base expert system*. Sistem pakar adalah suatu sistem yang dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli dalam menjawab pertanyaan dan memecahkan suatu masalah. Dengan sistem pakar ini orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli sistem pakar ini juga membantu aktivitasnya sebagai asisten yang sangat berpengalaman.

Sistem pakar merupakan cabang dari AI (*Artificial Inteligent*) yang membuat ekstensi khusus untuk spesialisasi pengetahuan guna memecahkan suatu permasalahan pada *Human Expert* (ahli manusia). *Human Expert* (ahli manusia) merupakan seseorang yang ahli dalam suatu bidang ilmu pengetahuan tertentu, ini berarti bahwa expert memiliki suatu pengetahuan atau skill khusus yang dimiliki oleh orang lain. *Expert* dapat memecahkan suatu permasalahan yang tidak dapat dipecahkan oleh orang lain dengan cara efisien (Nirmala, 2014).

Pengetahuan di dalam *expert system* berasal dari orang atau *knowledge* yang berasal dari buku-buku referensi, surat kabar atau karya ilmiah orang lain, pengetahuan manusia ke dalam komputer, agar komputer dapat menyelesaikan

masalah seperti yang biasa dilakukan oleh para ahli. Atau dengan kata lain sistem pakar adalah sistem yang didesain dan diimplementasikan dengan bantuan bahasa pemrograman tertentu untuk dapat menyelesaikan masalah seperti yang dilakukan oleh para pakar dalam hal ini adalah dokter (Nirmala, 2014).

Proses inferensi dilakukan dalam suatu modul yang disebut *inference engine* (mesin inferensi). Ketika representasi pengetahuan pada bagian *knowledge base* (dasar pengetahuan) telah lengkap, atau paling tidak telah berada pada level cukup akurat, maka referensi pengetahuan tersebut telah siap digunakan. Sedangkan *inferensi engine* (dasar pengetahuan) merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning* (pemikiran).

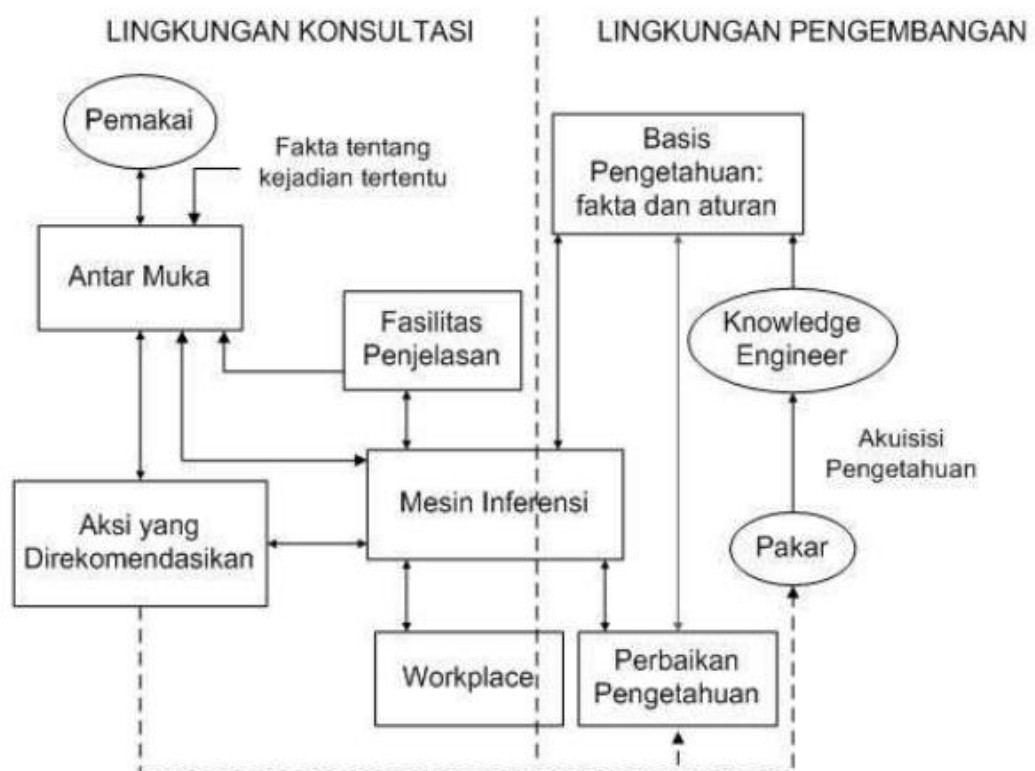
Terdapat dua metode umum penalaran yang dapat digunakan apabila pengetahuan dipresentasikan untuk mengikuti aturan-aturan sistem pakar yaitu metode *forward chaining* dan metode *backward chaining*.

Sistem pakar yang baik harus memenuhi ciri-ciri sebagai berikut :
(Wamiliana, 2013)

- 1) Terbatas pada bidang yang spesifik.
- 2) Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
- 3) Dapat mengemukakan rangkaian alasan yang diberikan dengan cara yang dapat dipahami.
- 4) Berdasarkan rule atau kaidah tertentu.
- 5) Dirancang untuk dapat dikembangkan secara bertahap.
- 6) Outputnya bersifat nasihat atau anjuran.

- 7) Output tergantung dari dialog dengan user.
- 8) Knowledge base dan inference engine terpisah.
- 9) Dapat digunakan dalam berbagai jenis komputer.

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar.



Gambar 2.1 Struktur Skematis Sistem Pakar

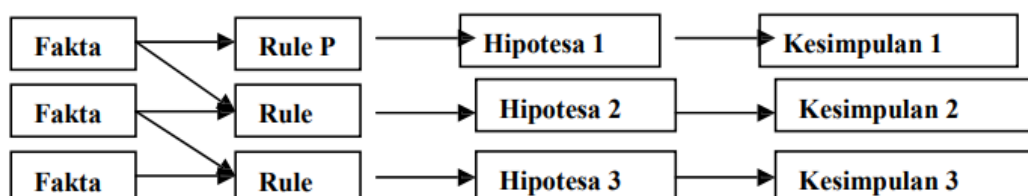
Sumber : (Wamiliana, 2013)

Mesin inferensi merupakan teknik penelusuran yang mengandung mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisis suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Dalam teknik inferensi, pelacakan dimulai dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada. Ada dua tipe teknik inferensi yaitu forward chaining dan backward chaining. (Wamiliana, 2013)

1) Forward Chaining

Forward Chaining adalah suatu metode pelacakan ke depan, dimana penelusuran diawali dari fakta-fakta yang diberikan user kemudian dicari rule yang sesuai dengan fakta-fakta yang ada. Setelah itu diadakan hipotesis untuk memperoleh kesimpulan. Forward Chaining melakukan pencarian dari suatu masalah kepada solusinya, jika klausa premis sesuai dengan situasi maka proses akan memberikan kesimpulan.

Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, mungkin proses menambahkan data ke memori kerja. Kemudian setelah data didapatkan aturan tersebut dijalankan. Proses diulang sampai ditemukan suatu hasil.

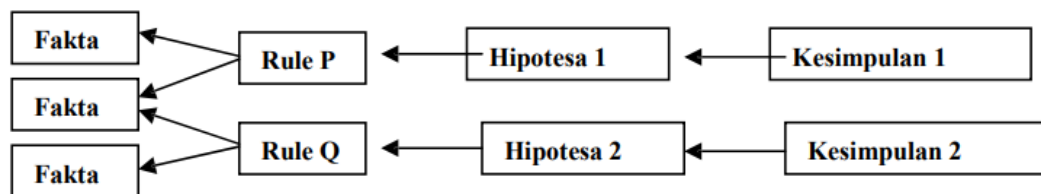


Gambar 2.2 Diagram *Forward Chaining*

Sumber : (Wamiliana, 2013)

2) Backward Chaining

Backward Chaining adalah suatu metode pengambilan keputusan yang juga umum digunakan dalam sistem pakar. Metode backward chaining adalah kebalikan dari forward chaining. Percobaan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN dulu). Dengan kata lain, penalaran dimulai dari hipotesis terlebih dahulu dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. Proses pencarian dengan metode backward chaining berangkat dari kanan ke kiri, yaitu dari kesimpulan sementara menuju kepada premis, metode ini sering disebut good driven pencarian dikendalikan oleh tujuan yang diberikan.



Gambar 2.3 Diagram *Backward Chaining*

Sumber : (Wamiliana, 2013)

2.3. Metode *Certainty Factor*

Certainty Factor (CF) adalah untuk mengakomodasi ketidakpastian pemikiran (*inexact reasoning*) seorang pakar yang di usulkan oleh *Shortliffe dan Buchanan* pada tahun 1975. Seorang pakar (misalnya dokter) sering menganalisis informasi yang ada dengan ungkapan dengan ketidak pastian, untuk mengakomodasi hal ini kita menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

Dalam mengekspresikan derajat kepastian, certainty factor untuk mengasumsikan derajat kepastian seorang pakar terhadap suatu data. Konsep ini kemudian diformulasikan dalam rumusan dasar sebagai berikut : (Harto, 2013)

$$CF [h,e] = MB [h,e] - MD [h,e] \dots\dots\dots(1)$$

Keterangan :

CF = Certainty factor (factor kepastian) dalam hipotesa H yang dipengaruhi oleh fakta E

MB(H,E) = measure of belief (ukuran kepercayaan) terhadap hipotesa H, jika diberikan evidence E (antara 0 dan 1)

MD(H,E) = measure of disbelief (ukuran kepercayaan) terhadap evidence H, jika diberikan evidence E (antara 0 dan 1)

Hipotesa = Hipotesa

E = Evidence (peristiwa atau fakta)

$$CF[H,E]_1 = CF[H] * CF[E] \dots\dots\dots$$

Dimana :

CF(E) = certainty factor evidence E yang dipengaruhi oleh evidence E

CF(H) = certainty factor hipotesa dengan asumsi evidence diketahui dengan pasti, yaitu ketika $CF(E,e) = 1$

CF(H,E) = certainty factor hipotesa yang dipengaruhi oleh evidence e diketahui dengan pasti

Certainty Factor untuk kaidah dengan kesimpulan yang serupa (similarly concluded rules) :

$$CF_{combine} CF[H,E]_{1,2} = CF[H,E]_1 + CF[H,E]_2 * [1 - CF[H,E]_1] \dots\dots\dots$$

$$CF_{combine} CF[H,E]_{old,3} = CF[H,E]_{old} + CF[H,E]_3 * (1 - CF[H,E]_{old}) ..$$

Tabel 2.1 Nilai CF

<i>Uncertainty Term</i>	CF
<i>Definitely not</i> (pasti tidak)	-1.0
<i>Almost certainly not</i> (hampir pasti tidak)	-0.8
<i>Probably not</i> (kemungkinan tidak)	-0.6
<i>Maybe not</i> (mungkin tidak)	-0.4
<i>Unknown</i> (tidak tahu)	-0.2 to 0.2
<i>Maybe</i> (mungkin)	0.4
<i>Probably</i> (kemungkinan benar)	0.6
<i>Almost certainly</i> (hampir pasti)	0.8
<i>Definitely</i> (pasti)	1.0

Sumber : (Harto, 2013)

Penggabungan kepercayaan dan ketidakpercayaan dalam bilangan yang tunggal memiliki dua kegunaan, yaitu pertama faktor kepastian digunakan untuk tingkat hipotesa di dalam urutan kepentingan. Sebagai contoh seorang pasien memiliki gejala tertentu yang menyarankan beberapa kemungkinan penyakit kemudian penyakit dengan CF tertinggi menjadi urutan pertama dalam pengurutan pengujian.

Metode certainty factor mempunyai kelebihan dan kekurangan, adapun kelebihan dari metode certainty factor antara lain :

- 1) Metode ini cocok dipakai dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendiagnosis dan mengidentifikasi hama atau penyakit sebagai salah satu contohnya.

- 2) Perhitungan dengan metode ini dalam sekali hitung hanya dapat mengolah dua data saja sehingga keakuratan data dapat terjaga.

Adapun kekurangan dari metode certainty factor yaitu :

- 1) Ide umum dari pemodelan kepastian manusia dengan menggunakan numeric certainty factor biasanya diperdebatkan sebagian orang akan membantah pendapat bahwa formula untuk metode certainty factor diatas memiliki sedikit kebenaran.
- 2) Metode ini dapat mengolah ketidakpastian / kepastian hanya dua data saja perlu dilakukan beberapa kali pengolahan data untuk data yang lebih dari dua buah.

2.4. Aplikasi *Mobile*

Aplikasi *Mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *Handphone*. Dengan menggunakan aplikasi *Mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya. Pemanfaatan aplikasi *Mobile* untuk hiburan paling banyak digemari oleh hampir 70% pengguna telepon seluler, karena dengan memanfaatkan adanya fitur game, music player, sampai video player membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun. (Kosidin, 2016) Dengan menggunakan aplikasi mobile, maka dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan,

berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya. (Surahman, 2017)

Beberapa penelitian juga sudah banyak yang menggunakan aplikasi mobile, baik itu untuk hiburan, mempermudah dalam layanan komunikasi data, maupun sebagai pengendali alat kamera DSLR. Aplikasi mobile dibangun dengan beberapa bahasa pemrograman mobile. Adapun contoh dari mobile programming untuk ponsel di antaranya adalah Javafx mobile, J2ME, C++, C#.NET dan Flash Lite. (Surahman, 2017)

2.5. *Android*

Android merupakan salah satu *Mobile Operating System* atau sistem operasi *handphone* yang berupa software platform open source untuk *Mobile device*, yang mana *Mobile Operating System* yaitu sistem operasi yang dapat mengontrol sistem dan kinerja barang elektronik berbasis *Mobile*, yang fungsinya sama seperti *Windows*, *Linux* dan *Mac OS X* pada *desktop PC* atau Notebook atau Laptop tetapi lebih sederhana. (Muharom, 2013)

Android merupakan sistem operasi yang berisi middleware serta aplikasi-aplikasi dasar. Basis sistem operasi *Android* yaitu kernel linux 2.6 yang telah diperbaharui untuk *Mobile device*. Pengembangan aplikasi *Android* menggunakan bahasa pemrograman java. Yang mana konsep-konsep pemrograman java berhubungan dengan Pemrograman Berbasis Objek (OOP)). Selain itu pula dalam pengembangan aplikasi *Android* membutuhkan software development kit (SDK)

yang disediakan *Android*, SDK ini memberi jalan bagi programmer untuk mengakses *application programming interface* (API) pada *Android*.

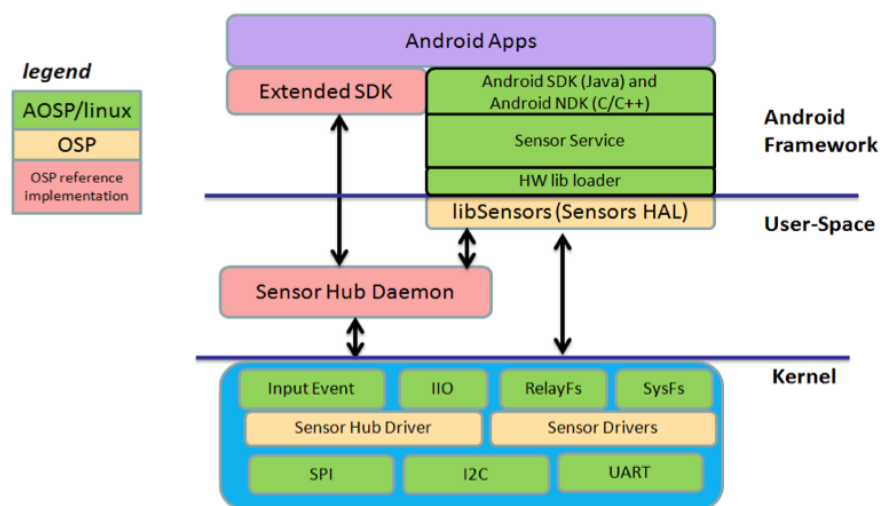
Android memiliki beberapa fitur yang menarik bagi yang ingin mengembangkan aplikasi, diantaranya sebagai berikut : (Fahnun, 2013)

- 1) *Application Framework* yang memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- 2) *Dalvik Virtual Machine*, yaitu mesin virtual yang dioptimalkan untuk perangkat *Mobile*.
- 3) *Graphic Library*, yang mendukung grafik 2D dan 3D berdasarkan OpenGL Library.
- 4) *Media Supported*, yang mendukung beberapa media seperti: audio, video, dan berbagai format gambar(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- 5) *Hardware Independent*, mendukung GSM, *Bluetooth*, EDGE, 3G, Wifi, kamera, GPS, kompas, dan *accelerometer*

2.6. Arsitektur *Android*

Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Pada lapisan *application framework* yang mencakup program untuk mengatur fungsi dasar *smartphone*. *Application Framework* merupakan serangkaian tool dasar seperti alokasi *resource smartphone*, aplikasi telepon, pergantian antar proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool

dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen - komponen, setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama memungkinkan pengguna mengganti komponen - komponen yang dikehendaki. (Fahnun, 2013)



Gambar 2.4 Framework Android

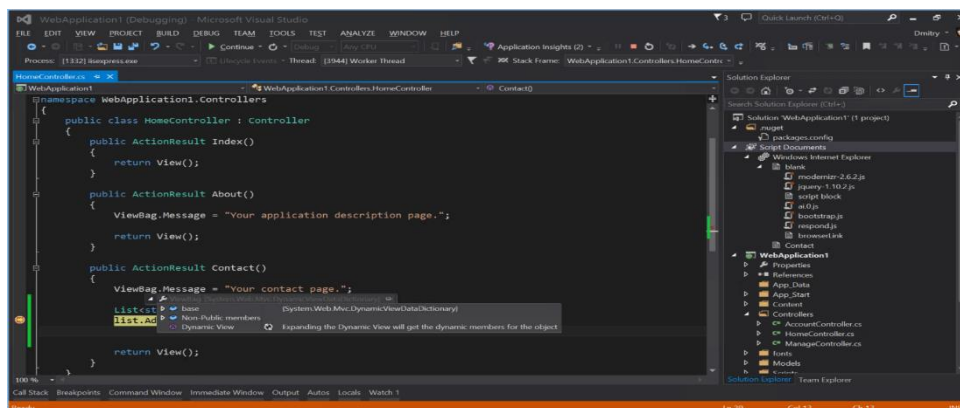
Sumber : (Tahir, 2011)

2.7. Definisi Visual Studio 2015

Visual Studio 2015 adalah rilis pertama *Microsoft* yang besar sejak beralih ke pendekatan *open source* untuk .NET dan teknologi terkait. Termasuk *compiler Roslyn* baru untuk C# dan *Visual Basic*, *.NET Core Framework*, *ASP.NET* itu sendiri, dan banyak lagi. Hasilnya memungkinkan jangkauan yang lebih luas untuk aplikasi .NET, termasuk membangun dan penggelaran di *Mac*, *Linux*, dan

Windows. Microsoft juga telah bekerja untuk mengintegrasikan Visual Studio dengan *framework* JavaScript sumber berbasis komunitas, manajer paket, dan perangkat UI. Model ASP.NET 5 menyederhanakan pengembangan web modern menggunakan kerangka kerja seperti *Bootstrap*, *AngularJS*, *Knockout*, *Gulp*, dan banyak lagi. (Lars, 2015)

Visual Studio 2015 mendukung model *Universal App* yang baru untuk dibangun di Windows. Aplikasi ini bisa ditulis sekali dan disesuaikan dengan desktop, tablet, dan telepon. Ini termasuk dukungan yang akan datang untuk pengembangan Windows 10. Pengembangan *Mobile cross-platform* juga didukung. Microsoft telah menyediakan template proyek untuk *Apache Cordova open-source*. Hal ini memungkinkan pengembang untuk membangun aplikasi *Mobile* yang berjalan di iOS, *Android*, dan *Windows Phone* menggunakan teknologi web *Hypertext Markup Language (HTML)*, *Cascading Style Sheets (CSS)*, dan *JavaScript*.



Gambar 2.5 Tampilan Visual Studio 2015

Sumber : (*microsoft.com*)

2.8. Pengertian Basis Data (Sistem Basis Data)

DBMS adalah suatu koleksi dari data yang saling berhubungan dan serangkaian program untuk mengakses data tersebut. Secara umum Database Manajement Sistem (DBMS) merupakan software yang akan menentukan data diorganisasikan, disimpan, diubah, diambil kembali, dan membaca data. DBMS merupakan antarmuka bagi pemakai dalam mengorganisasikan database yang disusunnya.(Yulansari, 2013)

Tujuan Basis data sendiri adalah sebagai berikut ini:

- 1) Kecepatan serta kemudahan dalam menyimpan, memanipulasi atau juga menampilkan kembali data tersebut.
- 2) Efisiensinya ruang penyimpanan, karena dengan basis data, redundansi data akan bisa dihindari.
- 3) Keakuratan (*Accuracy*) data.
- 4) Ketersediaan (*Availability*) data.
- 5) Kelengkapan (*Completeness*) data, Bisa melakukan perubahan struktur dalam basis data, baik dalam penambahan objek baru (tabel) atau dengan penambahan *field-field* baru pada table.
- 6) Keamanan (*Security*) data, dapat menentukan pemakai yang boleh menggunakan basis data beserta objek-objek yang ada didalamnya serta menentukan jenis -jenis operasi apa saja yang boleh dilakukannya.
- 7) Kebersamaan Pemakai (*Sharability*), Pemakai basis data bisa lebih dari satu orang, tetapi tetap menjaga atau menghindari masalah baru seperti: inkonsistensi data (karana data yang sama diubah oleh banyak pemakai

pada saat yang bersamaan) dan juga kondisi deadlock (karena ada banyak pemakai yang saling menunggu untuk menggunakan data tersebut.

Sedangkan pengertian sistem basis data adalah sistem yang terdiri dari koleksi data atau kumpulan data yang saling berhubungan dan program-program untuk mengakses data tersebut. Komponen Utama Sistem Basis Data :

- 1) Perangkat Keras (*Hardware*)
- 2) Sistem Operasi (*Operating Sistem*)
- 3) Basis data (*Database*)
- 4) Sistem Pengelola Basis Data (*Database Management Sistem* atau disingkat DBMS) Yaitu pengelola basis data secara fisik tidak dilakukan oleh pemakai secara langsung, akan tetapi ditangani oleh sebuah perangkat lunak yang khusus. Perangkat Lunak inilah yang disebut DBMS (*Database Management Sistem*) yang akan menentukan bagaimana data diorganisasi, disimpan, diubah serta diambil kembali. Perangkat Lunak ini juga yang menerapkan mekanisme pengamanan data, pemakaian data secara bersama-sama, konsistensi data dan sebagainya.
- 5) Pemakai (*User*).
- 6) Aplikasi atau Perangkat Lunak yang lainnya.
- 7) Tujuan Utama Sistem Basis Data sendiri adalah :
- 8) Menunjukkan suatu lingkungan yang tepat dan efisien didalam melakukan pengambilan (*retrieving*) dan penyimpanan (*storing*) informasi basis data,

serta menyediakan antarmuka yang lebih ramah kepada user dalam melihat data.

Kegunaan atau Fungsi Sistem Basis Data, mengatasi masalah-masalah pemrosesan data yang sering ditemui dengan menggunakan metode konvensional, permasalahan yang diatasi diantaranya:

- 1) Redudansi data dan juga inkonsistensi data.
- 2) Kesulitan dalam pengaksesan data.
- 3) *Data Isolation*.
- 4) Konkurensi pengaksesan.
- 5) Masalah keamanan.
- 6) Masalah Integritas.

Pemakai sistem basis data diantaranya:

- 1) Programmer Aplikasi yaitu orang atau pemakai yang berinteraksi dengan basis data melalui *Data Manipulation Language* (DML),
- 2) User Mahir (*Casual User*) yaitu pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Mereka menyatakan query untuk mengakses data dengan bahasa query yang telah disediakan oleh suatu DBMS.
- 3) User Umum (*End User/ Naïve User*) yaitu Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen (*executable program*) yang telah disediakan sebelumnya.
- 4) User Khusus (*Specialized User*) yaitu Pemakai yang menulis aplikasi basis data non konvensional, tetapi untuk keperluan-keperluan khusus.

2.9. Database SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. (Maulana, 2017)

Tidak seperti pada paradigma *client-server* umumnya, inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi overhead, latency times, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi desain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

SQLite merupakan paket perangkat lunak yang bersifat *public domain* yang menyediakan sistem manajemen basis data relasional atau RDBMS. Sistem basis data relasional digunakan untuk menyimpan *record* yang didefinisikan oleh pengguna pada ukuran tabel yang besar dan memproses perintah *query* yang kompleks dan menggabungkan data dari berbagai tabel untuk menghasilkan laporan dan rangkuman data. Kata '*Lite*' pada SQLite tidak menunjuk pada kemampuannya, melainkan menunjuk pada sifat dari SQLite, yaitu ringan ketika

dihubungkan dengan kompleksitas pengaturan, *administrative overhead*, dan pemakaian sumber. (Noer, 2017)

2.10. Unified Modeling Language (UML)

2.10.1. Pengenalan UML

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang berisi sintak dalam memodelkan sistem secara visual (Haviluddin, 2011). Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktur, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan *real time* (waktu nyata).


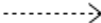




Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML).





2.10.2. Use Case Diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case*

digambarkan sebagai elips horizontal dalam suatu diagram UML *use case* (Haviluddin, 2011).

Tabel 2.2 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.






7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : (Gellysa Urva, 2015)

2.10.3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.3. Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity</i> <i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran




Sumber : (Gellysa Urva, 2015)

2.10.4. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode

yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 2.4. Simbol *Sequence Diagram*

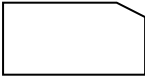
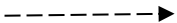
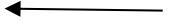
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : (Gellysa Urva, 2015)

2.10.5. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Tabel 2.5. Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
2		<i>dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya
3		<i>extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.

Sumber : (Gellysa Urva, 2015)

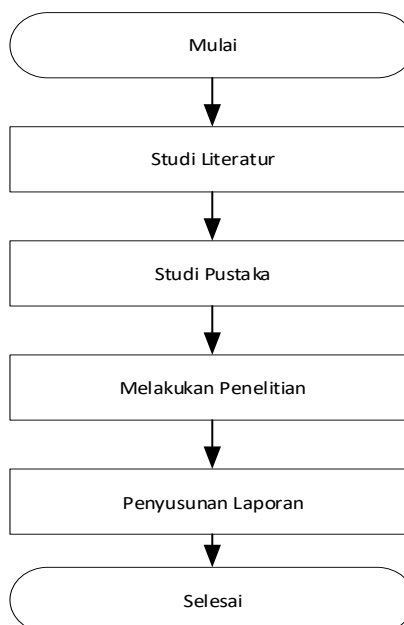
BAB III

METODE PENELITIAN

3.1. Metodologi Penelitian

Untuk memperoleh data-data yang diperlukan dalam penulisan skripsi ini, maka penulis menggunakan beberapa metode, sebagai berikut

3.1.1. Metode Pengumpulan Data



Gambar 3.1. Flowchart Penulisan

Metode pengumpulan data yang dilakukan dalam penelitian ini adalah :

1) Studi Literatur

Pengumpulan data dengan cara mengumpulkan *literature*, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

2) Studi Pustaka

Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis.

3) Melakukan Penelitian

Pengumpulan data dari penelitian pada Rumah Sakit Mata SMEC untuk mengetahui gejala dan penyakit mata yang diderita pasien.

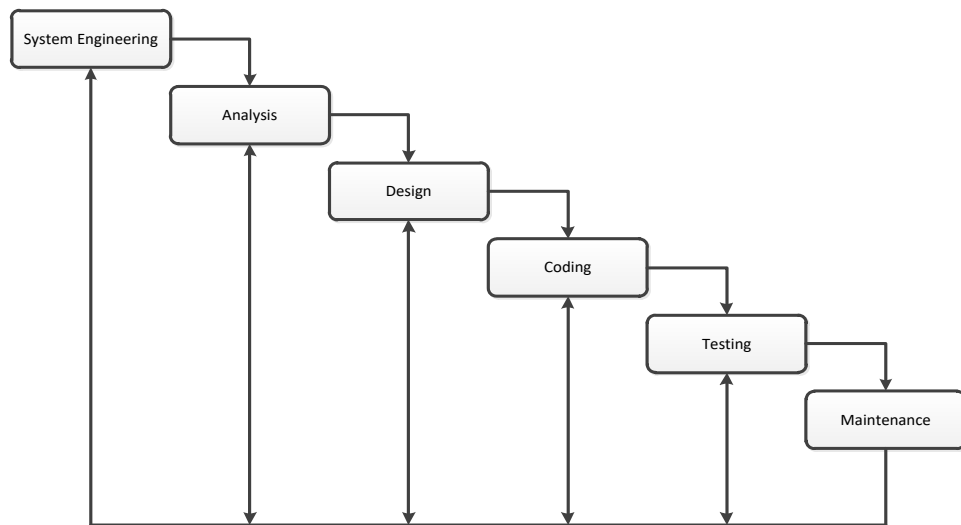
4) Penyusunan Laporan

Melakukan penyusunan laporan skripsi sesuai dengan hasil penelitian dan pengumpulan data yang telah didapat. Penyusunan laporan skripsi ini juga merupakan syarat penting untuk memperoleh gelar Sarjana Komputer pada Universitas Pembangunan Panca Budi.

3.1.2. Metode Pengembangan Perangkat Lunak

Metode yang digunakan untuk membangun sistem ini adalah Model *Waterfall*. Model ini merupakan sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematis, dengan beberapa tahapan, yaitu: *System Engineering, Analysis, Design, Coding, Testing* dan *Maintenance*.

Untuk lebih jelasnya tahapan-tahapan dari Paradigma *Waterfall* dapat dilihat pada gambar berikut:



Gambar 3.2. Paradigma Waterfall (Classic Life Cycle)

Sumber : Iqbal, 2017

Penjelasan Metodologi *Waterfall*:

- 1) *System Engineering* adalah Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan sebagainya.
- 2) *Analisis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan *software*. Dalam hal ini dilakukan dengan menganalisa dokumen-dokumen yang digunakan dalam informasi data.
- 3) *Design* adalah tahap penterjemah dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
- 4) *Coding* adalah tahap penterjemah data/pemecahan masalah *software* yang telah dirancang dalam bahasa pemrograman yang telah ditentukan

dan digunakan dalam pembuatan sistem menggunakan *software Visual Studio 2017* dengan bahasa pemrograman *Android C#* dan *SQLite Database*.

- 5) *Testing* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dimulai dengan membuat suatu uji kasus untuk setiap fungsi pada perangkat lunak untuk sistem informasi pengolahan data-data kemudian dilanjutkan dengan pengujian terhadap modul-modul dan terakhir pada tampilan antar muka untuk memastikan tidak ada kesalahan dan semua berjalan dengan baik dan input yang diberikan hasilnya sesuai dengan yang diinginkan.
- 6) *Implementation* adalah menguji *software* aplikasi yang telah dibuat dan dirancang dengan membuat aplikasi *software* menjadi *exe*, dan diimplementasikan sesuai pembahasan tersebut.
- 7) *Maintenance* adalah perangkat lunak yang telah dibuat dapat mengalami perubahan sesuai permintaan pemakai. Pemeliharaan dapat dilakukan jika ada permintaan tambahan fungsi sesuai dengan keinginan pemakai ataupun adanya pertumbuhan dan perkembangan baik perangkat lunak maupun perangkat keras.

3.2. Analisis Sistem

Analisis masalah adalah penguraian dari suatu masalah yang utuh ke dalam bagian-bagian komponennya dengan maksud mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

Mata adalah salah satu indera yang penting bagi manusia, melalui mata manusia menyerap informasi visual yang digunakan untuk melaksanakan berbagai kegiatan. Mata merupakan salah satu dari panca indra yang memiliki peran yang sangat penting dalam kehidupan manusia yaitu sebagai organ penglihatan. Jika terjadi gangguan pada mata atau penyakit mata, maka akan sangat mengganggu dan jika tidak ditindak lanjuti dapat berakibat sangat fatal bagi kehidupan manusia.

Seiring dengan menurunnya kualitas dan gaya hidup seperti pola makan, olahraga, istirahat, bekerja, tingkat stres dan usia, jumlah individu dengan keluhan penyakit mata semakin bertambah. Penyebab utama dari kasus kebutaan ini adalah katarak, kelainan kornea, glaukoma, kelainan refraksi, kelainan retina dan kelainan nutrisi. Situasi tersebut dapat dihindari jika masyarakat memiliki sedikit pengetahuan tentang kesehatan mata. Pada Penelitian ini penulis ingin membuat sistem yang dapat membantu masyarakat dalam mendeteksi jenis penyakit mata yang diderita oleh seseorang.

Melihat dari jenis penyakit mata yang diderita tersebut, penulis ingin membuat suatu aplikasi yang dapat mendeteksi serangan penyakit mata ini dengan cara membuat suatu aplikasi sistem pakar yang dapat mendeteksi penyakit mata yang diderita. Cara mendiagnosa penyakit mata adalah dengan cara menjawab pertanyaan seputar gejala yang sedang dialami oleh penderita yang sedang

mengalami gejala penyakit mata. Sehingga penderita dapat lebih cepat dalam melakukan penanganan penyakit mata yang diderita karena sudah terdeteksi jenis penyakit mata yang menyerangnya.

System yang diusulkan nantinya diharapkan bisa lebih membantu untuk proses mendeteksi penyakit mata yang diderita apakah terserang penyakit mata atau tidaknya. Proses yang dilakukan nantinya dapat dilakukan tanpa memakan waktu dan uang yang dikeluarkan oleh pengguna.

3.3. Fungsional Aplikasi

Perancangan aplikasi metode *certainty factor* pada penyakit mata memiliki fungsi sebagai berikut :

- 1) Memberikan hasil jenis penyakit mata yang diderita.
- 2) Memberikan solusi penanganan dari jenis penyakit mata agar lebih cepat dalam memberikan pengobatan secara cepat.

3.4. Analisis Kebutuhan

1. Analisis Kebutuhan Fungsional

Terdapat tiga alat penelitian dalam penyelesaian aplikasi metode *certainty factor* pada penyakit mata ini, yaitu :

1) *Smartphone Android*

Smartphone Android yang digunakan untuk menguji coba aplikasi ini, memiliki spesifikasi sebagai berikut :

- a) *CPU* : *Qualcomm MSM8909 1,6 GHz*

- b) *Memory Internal* : 1 GB RAM, 8 GB ROM
- c) *Memory External* : 8 GB
- d) *Operating System* : Android OS, V5.0 (Lollipop)
- e) *Tipe Layar* : Corning Gorilla Glass 3 Multi Touch
Screen
- f) *Ukuran Layar* : 720 x 1280 pixel

2) *Hardware*

Hardware yang digunakan memiliki spesifikasi sebagai berikut :

- a) *Processor* : Intel® Core™ i3
- b) *Memory* : 2 GB DDR 3
- c) *Harddisk* : 500 GB
- d) *Display* : 14 Inch WXGA (1366 x 768)
- e) *Sound Card* : Integrated
- f) *Video Type* : Intel®HD Graphics dan Nvidia G-Force GT
520M
- g) *Keyboard, Mouse, Speaker, Headset.*

3) *Software*

Software yang digunakan dalam pembuatan aplikasi metode certainty factor pada penyakit mata adalah :

- a) *Visual Studio .Net 2017*

2. Analisis Kebutuhan Non Fungsional (Untuk Pengguna)

Spesifikasi minimum *hardware* yang dapat menjalankan aplikasi metode certainty factor pada penyakit mata ini adalah sebagai berikut :

1) *Smartphone Android*

Smartphone Android yang digunakan untuk menguji coba aplikasi ini, memiliki spesifikasi sebagai berikut :

- a) *CPU* : *Dual-core 1.2 GHz Cortex-A9*
- b) *Memory Internal* : *512 GB RAM, 4 GB ROM*
- c) *Memory External* : *2 GB*
- d) *Operating System* : *Android OS, V5.0 (Lollipop)*
- e) Tipe Layar : *Corning Gorilla Glass 3 Multi Touch Screen*
- f) Ukuran Layar : *840 x 480 pixel*

3.5. Rule Aplikasi Metode Certainty Factor Pada Penyakit Mata

Berikut adalah deskripsi aturan pada aplikasi metode certainty factor pada penyakit mata:

3.5.1. Gejala Penyakit

Tabel 3.1. Gejala

Kode	Gejala	Nilai
G01	Mata nyeri hebat	0.6
G02	Mata menonjol	0.3
G03	Penglihatan kabur	0.8
G04	Peka terhadap cahaya	0.8

G05	Mata merah	0.8
G06	Mata berair	0.5
G07	Mata perih	0.5
G08	Mata gatal	0.7
G09	Kelopak Mata membengkak	0.8
G10	Mata ungu	0.6
G11	Mata sakit	0.5
G12	Air mata berlebihan	0.4
G13	Mata tegang	0.8
G14	Mata meradang	0.9
G15	Mata kering	0.8
G16	Mata iritasi	0.5
G17	Mata nyeri bila ditekan	0.6
G18	Demam	0.2
G19	Menekan kedipan berlebihan	0.7
G20	Sel batang retina sulit beradaptasi diruang yang remang-remang	0.6
G21	Pada siang hari penglihatan menurun	0.3
G22	Tidak dapat melihat pada lingkungan yang kurang bercahaya	0.4
G23	Pergerakan mata terbatas	0.8
G24	Mata tampak mengkilat	0.4
G25	Bola mata membengkak dan tampak berkabut	0.6
G26	Sumber cahaya akan berwarna pelangi bila memandang lampu neon	0.5

G27	Penglihatan yang tadinya kabur lama-kelamaan menjadi normal	0.5
G28	Malam hari kesulitan melihat	0.3
G29	Siang hari ketajaman mata menurun	0.7
G30	Mata silau akan cahaya	0.4
G31	Sering ganti kaca mata	0.9
G32	Penglihatan ganda pada salah satu sisi mata	0.8
G33	Lensa mata membengkak	0.5
G34	Berbentuk keropeng pada kelopak mata ketika bangun pada siang hari	0.6
G35	Penglihatan menurun pada ruang gelap	0.5
G36	Penglihatan menurun pada malam hari	0.6
G37	Keluarnya cairan kotor dari mata	0.2
G38	Berbaliknya bulu mata	0.3
G39	Pembengkakan kelenjar getah bening didepan telinga	0.3
G40	Munculnya garis parutan pada kornea	0.3
G41	Komplikasi pada,telinga, hidung dan tenggorokan	0.5
G42	Mata mempersempit, perubahan bentuk	0.8
G43	Benjolan pada mata bagian atas atau bawah	0.7
G44	Gangguan penglihatan pada salah satu mata	0.5
G45	Garis mata lurus terlihat bergelombang	0.6
G46	Mata tidak nyeri	0.5
G47	Mata melihat melayang-layang	0.3
G48	Mata melihat kilatan cahaya	0.4
G49	Seperti ada benda asing di mata	0.5

G50	Sakit kepala	0.6
G51	Riwayat penyakit menular seksual pada ibu	0.8
G52	Mata membenkakan	0.5
G53	Peka terhadap cahaya (fotofobia)	0.8
G54	Terasa nyeri	0.4
G55	Tampak bintik nanah berwarna kuning keputihan pada kornea	0.6
G56	Terdapat kotoran mata	0.5
G60	Gangguan Penglihatan	0.4
G61	Penglihatan kabur	0.9
G62	Kesulitan melihat di malam hari	0.8
G64	Mengalami iritasi	0.8
G65	Terjadi pembengkakan bundar pada kelopak mata dan tumbuh secara perlahan	0.4
G66	Terbentuk daerah kemerahan/abu-abu di bawah kelopak mata	0.6
G67	Bulu mata rontok	0.5
G68	Mata sukar dibuka ketika bangun dipagi Hari	0.5
G69	Alergi	0.3
G70	Mata terasa panas	0.7

3.5.2. Penyakit

Tabel 3.2. Penyakit

Kode	Penyakit	Solusi
P01	<i>Xerophthalmia</i>	<p>Menghindari iklim atau kondisi ruangan yang kering.</p> <p>Menggunakan alat pembersih atau pelembab udara di dalam ruangan.</p> <p>Memakai kaca mata pelindung yang dapat memperlambat penguapan air dari permukaan mata.</p> <p>Menggunakan salep, gel, atau air mata buatan. Kendati demikian, hindari air mata dengan bahan pengawet jika harus menggunakannya lebih dari empat kali dalam sehari.</p> <p>Mengistirahatkan mata setelah melakukan kegiatan yang memerlukan ketajaman penglihatan dalam waktu lama.</p>
P02	<i>Selulitis Orbitalitas</i>	<p>Tindakan pengobatan yang biasa dilakukan terhadap penderita penyakit ini adalah dengan memberikan obat antibiotika yang dikonsumsi secara oral oleh penderita. Namun jika penyakit ini sudah parah, maka pengobatan terbaik adalah melalui</p>

		pembedahan untuk mengeluarkan nanah.
P03	<i>Glaukoma</i>	<p>Golongan analog prostaglandin. Contohnya latanaprost, travoprost, tafluprost, dan bimatoprost. Cara pakainya adalah satu kali sehari di malam hari. Kemanjuran obat ini baru bisa dirasakan dalam 4 minggu setelah memulai pengobatan glaukoma. Salah satu efek samping yang paling sering terjadi adalah warna iris (lingkaran hitam mata) berubah menjadi lebih gelap.</p> <p>Golongan antagonis β-adrenergik. Contohnya timolol dan betaxolol. Obat tetes mata golongan ini biasa digunakan di pagi hari. Betaxolol akan menjadi pilihan dokter untuk Anda jika Anda memiliki gangguan paru-paru.</p> <p>Golongan inhibitor karbonik anhidrase. Contohnya dorzolamide dan brinzolamide. Obat golongan ini digunakan tiga kali sehari dan dapat terus digunakan sebagai terapi jangka panjang. Efek samping yang paling</p>

		<p>sering adalah rasa pahit di mulut yang timbul setelah obat diteteskan.</p> <p>Golongan parasimpatomimetik. Contoh pilokarpin. Obat ini biasa digunakan sebagai tambahan pada kasus tekanan bola mata tinggi dalam jangka waktu panjang yang sudah menjalani prosedur laser namun target tekanan yang diinginkan belum tercapai.</p>
P04	<i>Dakriosistitis</i>	<ul style="list-style-type: none"> - Fase kemerahan. Pada fase ini, Anda mungkin diresepkan antibiotik minum dan obat pereda nyeri. Anda juga bisa kompres hangat mata untuk meredakan gejala-gejalanya. - Fase bengkak (sudah mengandung nanah). Selain pemberian obat antibiotik, akan dilakukan operasi kecil untuk mengeluarkan nanah. - Fase penyembuhan. Saat kemerahan dan bengkak sudah hilang, akan dilakukan tindakan operasi untuk menutup daerah berlubang yang tadinya terisi oleh nanah. Tindakan ini perlu dilakukan untuk mencegah infeksi terjadi lagi.

P05	<i>Katarak</i>	<p>Operasi katarak. Ada 2 jenis operasi untuk meringankan gejala katarak, yaitu:</p> <ul style="list-style-type: none"> - Small incision cataract surgery (phacoemulsification). Operasi ini dilakukan dengan melakukan insisi kecil pada tepi kornea. Dokter menyinarkan gelombang ultrasound untuk menghancurkan lensa lalu diambil menggunakan penghisap. - Extracapsular surgery yang membutuhkan insisi yang lebih lebar untuk mengeluarkan inti lensa yang berkabut. Sisa lensa dikeluarkan dengan menggunakan penghisap
P06	<i>Konjungtivitis</i>	<ul style="list-style-type: none"> - Hindari kontak langsung dengan siapapun jika Anda menderita mata merah. Gunakan sapu tangan atau tisu untuk membersihkan kotoran mata - Sering cuci tangan - Gunakan handuk, lap, dan bantal yang terpisah dari keluarga Anda di rumah. - Buang kosmetik mata yang sudah lama dan jangan berbagi kosmetik mata dengan orang lain - Hindari penyebab alergi, jika bisa - Gunakan pengobatan sesuai yang diinstruksikan

		<ul style="list-style-type: none"> - Jangan sentuh area yang terinfeksi atau mengucek mata Anda - Jangan gunakan lensa kontak sampai pengobatan selesai. Anda mungkin juga harus mengganti lensa kontak Anda atau kotak penyimpanannya
P07	<i>Retinitis Pigmentosa</i>	<p>Pengobatan dengan antioksidan (seperti vitamin A palmitat dosis tinggi) dapat memperlambat penyakit. Namun, mengonsumsi vitamin A dosis tinggi dapat menyebabkan masalah liver yang serius. Manfaat dari pengobatan harus ditimbang dengan risiko pada liver.</p>
P08	<i>Trakoma</i>	<ul style="list-style-type: none"> - Surgery (pembedahan) - Antibiotics (pemberian antibiotik) - Face cleanliness (menjaga kebersihan daerah wajah) - Environmental improvement (menjaga kebersihan lingkungan).
P09	<i>Oveitis</i>	<p>mengonsumsi dosis vitamin E yang cukup dapat membantu meningkatkan penglihatan untuk orang dengan uveitis. Mengonsumsi vitamin E dengan vitamin C dengan diminum dapat memperbaiki penglihatan,</p>

		namun tidak mengurangi pembengkakan pada orang dengan uveitis.
P10	<i>Hordeolum</i>	<ul style="list-style-type: none"> - Selalu menjaga muka, kulit kepala, alis, dan tangan Anda dalam kondisi bersih - Gunakan sedikit make up mata atau tidak sama sekali, kemudian bersihkan dengan sampai benar-benar bersih - Jangan memeras atau mencoba memencet timbilan - Jangan menggunakan lensa kontak selama timbilan masih ada
P11	<i>Degenerasi Makula</i>	<ul style="list-style-type: none"> - Makan lebih banyak buah-buahan, sayuran, ikan, dan menghindari makanan tinggi lemak - Dilarang merokok - Lakukan pemeriksaan mata berkala dan ikuti petunjuk dokter - Membaca atau melakukan hal-hal yang membutuhkan ketajaman mata hanya jika ada cahaya yang cukup
P12	<i>Ablasio Retina</i>	<ul style="list-style-type: none"> - Mengenakan kacamata pelindung saat berolahraga atau menggunakan peralatan pelindung wajah. - Apabila Anda memiliki diabetes, kendalikan gula

		<p>darah dan kunjungi dokter secara rutin.</p> <ul style="list-style-type: none"> - Lakukan pemeriksaan mata tahunan, terutama apabila Anda memiliki risiko ablasio retina.
P13	<i>Pterygium</i>	<ul style="list-style-type: none"> - Gunakan kaca mata hitam saat Anda di luar ruangan - Biarkan mata beristirahat di bawah tempat yang teduh setelah berada di luar ruangan - Gunakan tetes mata atau air mata buatan untuk melembabkan mata - Hindari debu, angin, asap, dan pollen
P14	<i>Miopi</i>	<ul style="list-style-type: none"> - Memeriksa mata secara teratur. - Mengendalikan kondisi kesehatan kronis, misalnya diabetes dan tekanan darah tinggi. - Melindungi mata dari sinar matahari dengan memakai kacamata yang menghambat radiasi ultraviolet (UV). - Mencegah cedera mata. Anda disarankan memakai pelindung mata saat melakukan aktivitas tertentu macam berolahraga, memangkas rumput, mengecat, atau menggunakan produk lain yang mengandung asap beracun. - Berhenti merokok. Hal ini tidak hanya membantu mengistirahatkan tubuh

		<p>melainkan juga kesehatan mata.</p> <ul style="list-style-type: none"> - Menggunakan lensa korektif yang tepat. - Mengurangi kelelahan mata. Setelah beberapa menit menggunakan komputer atau membaca, Anda harus berpaling sehingga mata dapat rileks.
P15	<i>Oftalmia neonatorum</i>	<p>Pengobatan konjungtivitis diberikan sesuai dengan penyebabnya. Jika konjungtivitis disebabkan oleh bakteri, maka perlu diberikan obat mata yang mengandung antibiotik. Sekitar 50% bayi yang menderita konjungtivitis karena infeksi klamidia juga mengalami infeksi klamidia di bagian tubuh lainnya. Oleh karena itu juga perlu diberikan antibiotik melalui mulut.</p> <p>Untuk konjungtivitis yang disebabkan oleh infeksi virus herpes, penderita juga perlu diobati dengan obat anti virus, dengan pertimbangan bahwa virus telah menyebar atau akan menyebar ke organ lainnya. Untuk memastikan diagnosa, serta mendapatkan pilihan dan dosis pengobatan yang tepat, sebaiknya anak segera dibawa ke dokter untuk diperiksa terlebih dahulu.</p>

P16	<i>Ulkus Kornea</i>	<ul style="list-style-type: none"> - Jika Anda menggunakan lensa kontak, segera lepas. - Gunakan kompres dingin pada mata yang terpengaruh. - Jangan menyentuh atau mengucek mata dengan jari Anda. - Batasi penyebaran infeksi dengan sering mencuci tangan dan mengeringkannya dengan handuk yang bersih. - Gunakan obat penawar rasa sakit yang dijual bebas, seperti acetaminophen (Typlenol) atau ibuprofen (Motrin). - Hindari tidur saat masih mengenakan lensa kontak. - Bersihkan dan sterilkan lensa kontak sebelum dan setelah pemakaian. - Bilas mata untuk mengeluarkan objek asing. <p>Cuci tangan sebelum menyentuh mata Anda.</p>
P17	<i>Keratokonus</i>	<ul style="list-style-type: none"> - Kacamata atau lensa kontak yang lembut bisa memperbaiki gangguan penglihatan pada tahap awal keratoconus. Namun, Anda mungkin harus sering ganti resep untuk

		<p>lensanya akibat perubahan kornea.</p> <ul style="list-style-type: none"> - Lensa kontak yang kaku biasanya akan diresepkan sebagai pengobatan berikutnya. Awalnya mungkin terasa kurang nyaman, tapi banyak orang akhirnya terbiasa juga karena lensa seperti ini bisa dipaskan dengan bentuk kornea Anda. - Lensa dobel bisa direkomendasikan kalau Anda merasa tidak nyaman pakai lensa yang kaku. Anda bisa menggunakan lensa kontak yang lembut dulu, baru yang kaku. - Lensa kontak hibrid, bagian tengahnya kaku sedangkan bagian tepinya lebih lembut agar lebih nyaman digunakan. - Lensa kontak sklera digunakan pada orang yang kondisinya sudah cukup parah. Lensa kontak ini cukup nyaman karena diletakkan pada sklera (bagian putih mata), bukan pada kornea seperti lensa kontak pada umumnya.
P18	<i>Kalazion</i>	<ul style="list-style-type: none"> - usaplah secara perlahan kelopak mata dengan kain bersih atau kapas - jangan memencet atau mencoba untuk menekan mata - jagalah agar wajah, kulit kepala, alis dan tangan agar tetap bersih - batasi penggunaan makeup pada mata

		<p>- beritahu dokter jika Anda merasakan kelainan atau gejala tidak pulih ketika telah menggunakan obat tetes mata atau obat minum yang diresepkan oleh dokter</p> <p>- beritahu dokter jika Kalazion tidak hilang setelah 2 minggu pengobatan</p>
P19	<i>Blefaritis</i>	<p>- Basahi kain bersih dengan air hangat (jangan air panas).</p> <p>- Peras dan letakkan pada mata yang tertutup selama 5 menit.</p> <p>- Basahi kembali seperlunya untuk menjaga tetap hangat. Hal ini dapat melembutkan kerak dan meluruhkan serpihan berminyak.</p>

3.5.3. Aturan

Tabel 3.3. Aturan

Kode	Penyakit	Kode Gejala
P01	<i>Xerophthalmania</i>	G20, G21, G22
P02	<i>Selulitis Orbitalitas</i>	G01, G02, G23, G09, G24, G05, G10, G18, G25
P03	<i>Glaukoma</i>	G26, G11, G52, G27, G19
P04	<i>Dakriosistitis</i>	G12, G01, G52
P05	<i>Katarak</i>	G28, G21, G30, G31, G32, G01, G33

P06	<i>Konjungtivitis</i>	G06, G01, G08, G03, G04, G034
P07	<i>Retinitis Pigmentosa</i>	G35, G36, G09
P08	<i>Trakoma</i>	G37, G09, G38, G39, G40, G41
P09	<i>Oveitis</i>	G14, G06, G04, G42, G43
P10	<i>Hordeolum</i>	G05, G01, G06, G04
P11	<i>Degenerasi Makula</i>	G44, G45, G46
P12	<i>Ablasio Retina</i>	G47, G48, G03, G13
P13	<i>Pterygium</i>	G43, G05, G16, G15, G06, G49, G03
P14	<i>Miopi</i>	G01, G07, G12, G50
P15	<i>Oftalmia eonatorium</i>	G05, G09, G17, G51, G03
P16	<i>Ulkus Kornea</i>	G53, G54, G55, G56, G60
P17	<i>Keratokonus</i>	G53, G57, G61, G62
P18	<i>Kalazion</i>	G64, G65, G66
P19	<i>Blefaritis</i>	G53, G63, G67, G68, G69, G70, G58, G59

Berikut adalah hasil diagnosa *Xerophthalmia* dengan menggunakan perhitungan secara manual :

1. *Xerophthalmia*

$$\begin{aligned} \text{CFgejala G20} &= \text{CFuser}(0.8) * \text{CF}(0.6) \\ &= 0.48 \end{aligned}$$

$$\begin{aligned} \text{CFgejala G21} &= \text{CFuser}(0.8) * \text{CF}(0.3) \\ &= 0.24 \end{aligned}$$

$$\begin{aligned} \text{CFgejala G22} &= \text{CFuser}(0.8) * \text{CF}(0.4) \\ &= 0.32 \end{aligned}$$

$$\text{CFcombine1}(\text{CF G20}, \text{CF G21}) = 0.48 + 0.24 * (1 - 0.48)$$

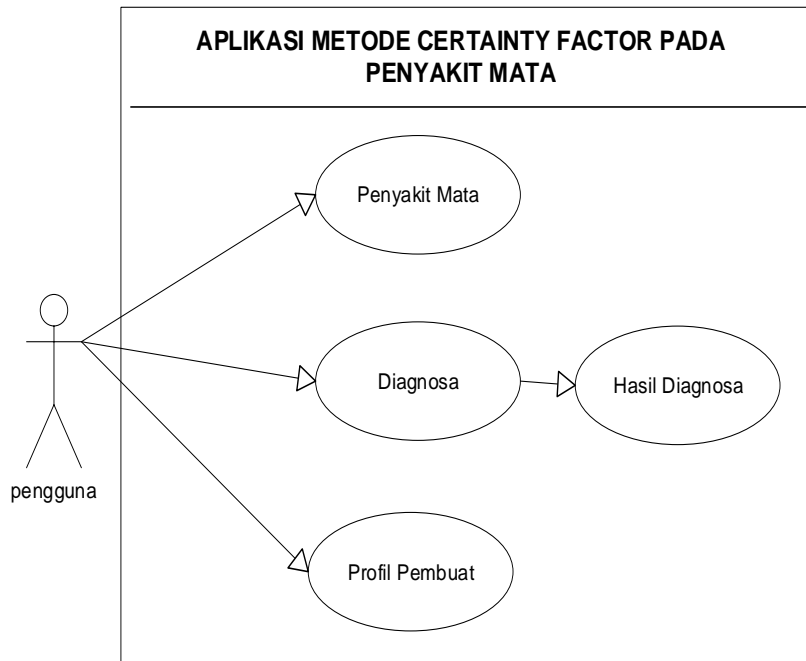
$$\begin{aligned}
\text{CFold1} &= 0.6048 \\
\text{CFcombine2}(\text{CFold1}, \text{CF G22}) &= 0.6048 + 0.32 * (1 - 0.6048) \\
\text{CFold2} &= 0.7312 \\
\text{Persentase} &= \text{Cfold2} * 100 \\
&= 0.7312 * 100 \\
&= 73.1264\%
\end{aligned}$$

Dari perhitungan secara manual di atas, didapatkan nilai faktor kepastian dari masukan gejala yang mengarah ke *Xerophthalmia* adalah 73.1264%.

3.6. Perancangan Sistem

3.6.1. Use Case Diagram

Untuk mendapatkan informasi dari sebuah sistem yang dibuat, maka penulis menggunakan *use case diagram*. Dengan diagram ini, proses yang terjadi pada sebuah aplikasi akan dapat diketahui. *Use case diagram* dari aplikasi metode certainty factor pada penyakit mata dapat dilihat dibawah ini :



Gambar 3.3 Use Case Diagram Aplikasi Certainty Factor Pada Penyakit Mata

3.6.2. Use Case

1) Definisi Aktor

Berikut adalah deskripsi pendefinisian aktor pada aplikasi metode certainty factor pada penyakit mata :

Tabel 3.4 Definisi Aktor

Aktor	Deskripsi
Pengguna	Orang yang menggunakan aplikasi metode certainty factor pada penyakit mata.

2) Definisi *Use Case*

Berikut adalah deskripsi pendefinisian *Use case* pada aplikasi metode certainty factor pada penyakit mata :

Tabel 3.5 Definisi *Use Case*

No	<i>Use case</i>	Deskripsi
1.	Penyakit Mata	Merupakan menu yang berisi pengertian tentang penyakit mata yang sering diderita.
2.	Diagnosa	Merupakan menu yang berisi proses diagnosa mata apakah terserang penyakit dengan memberikan pertanyaan seputar gejala yang dialami.
3.	Profil Pembuat	Merupakan menu yang berisi tentang pembuat aplikasi metode certainty factor pada penyakit mata

1) Skenario *Use case*

Berikut adalah skenario jalannya masing-masing *use case* yang telah didefinisikan sebelumnya :

1. Skenario *Use Case* Penyakit Mata

Nama *Use case* : Penyakit Mata

Skenario :

Tabel 3.6 Skenario Use Case Penyakit Mata

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu penyakit Mata	
	2. Menampilkan pengertian tentang penyakit mata yang sering diderita

2. Skenario *Use case* Diagnosa

Nama *Use case* : Diagnosa

Skenario :

Tabel 3.7 Skenario Use Case Diagnosa

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu diagnosa	
	2. Menampilkan menu diagnosa apakah sedang terserang penyakit mata atau tidak.

3. Skenario *Use case* Profil Pembuat

Nama *Use case* : Profil Pembuat

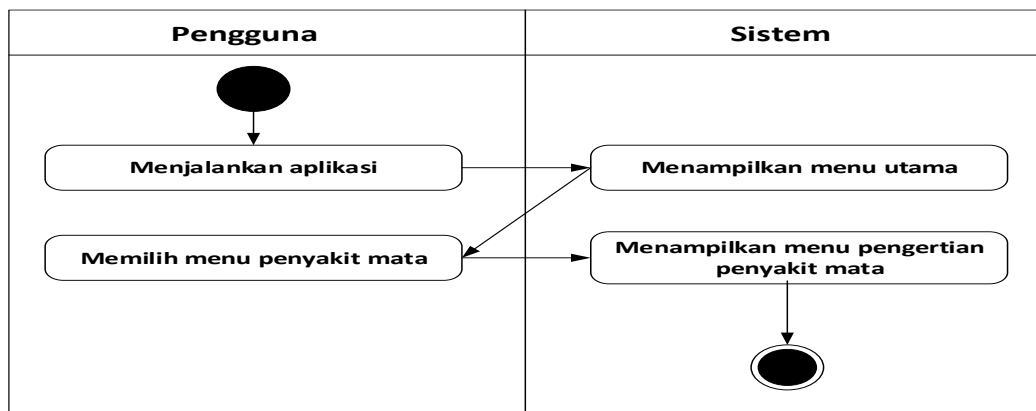
Skenario :

Tabel 3.8 Skenario Use Case Profil Pembuat

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu profil pembuat	
	2. Menampilkan menu yang berisi profil pembuat aplikasi metode certainty factor pada penyakit mata.

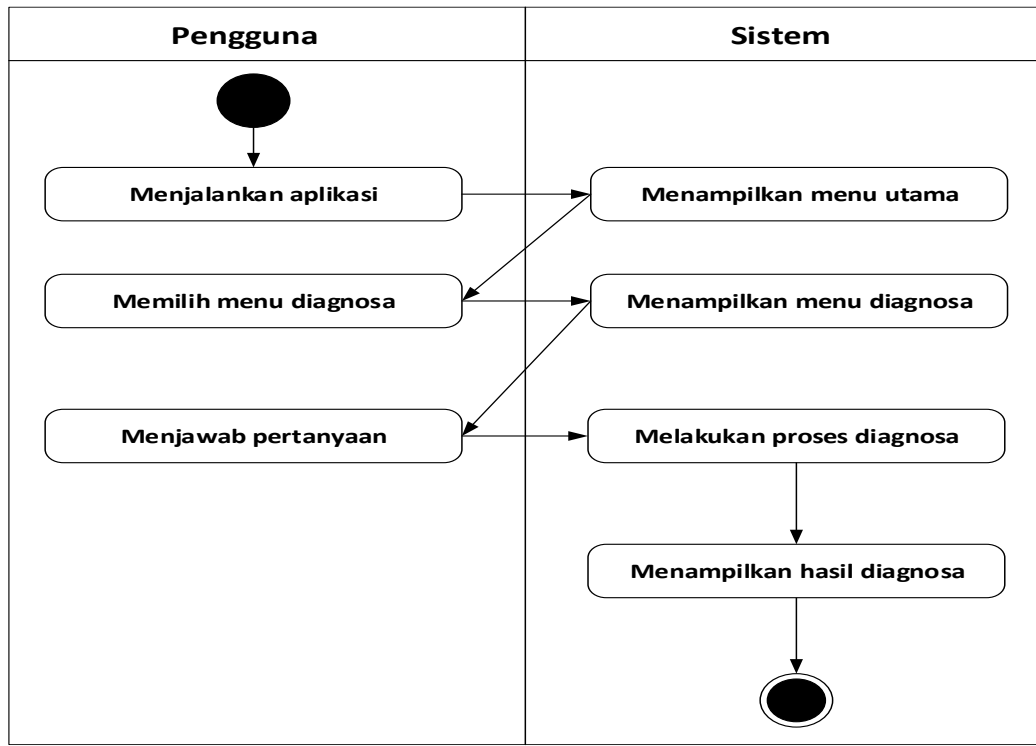
3.6.3. Activity Diagram

1. Activity Diagram Penyakit Mata



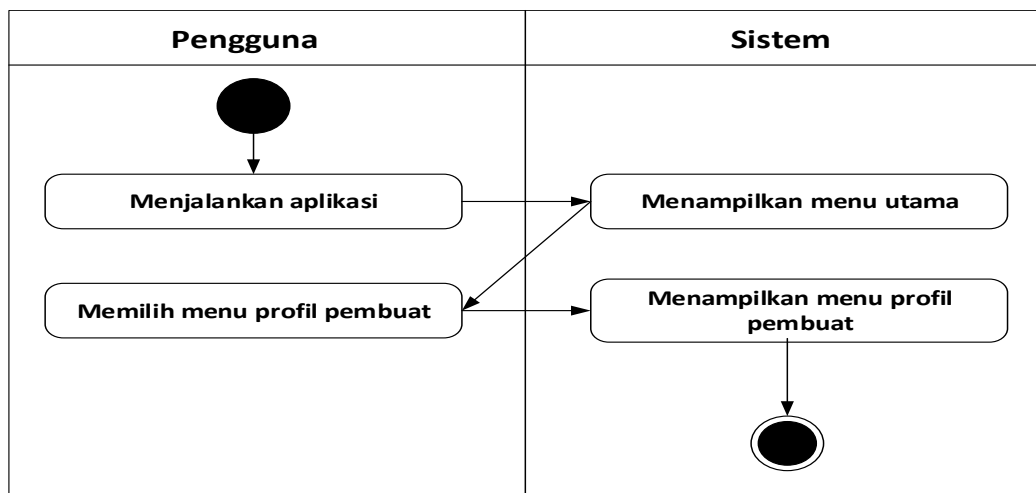
Gambar 3.4 Activity Diagram Penyakit Mata

2. Activity Diagram Diagnosa



Gambar 3.5 Activity Diagram Diagnosa

3. Activity Diagram Profil Pembuat

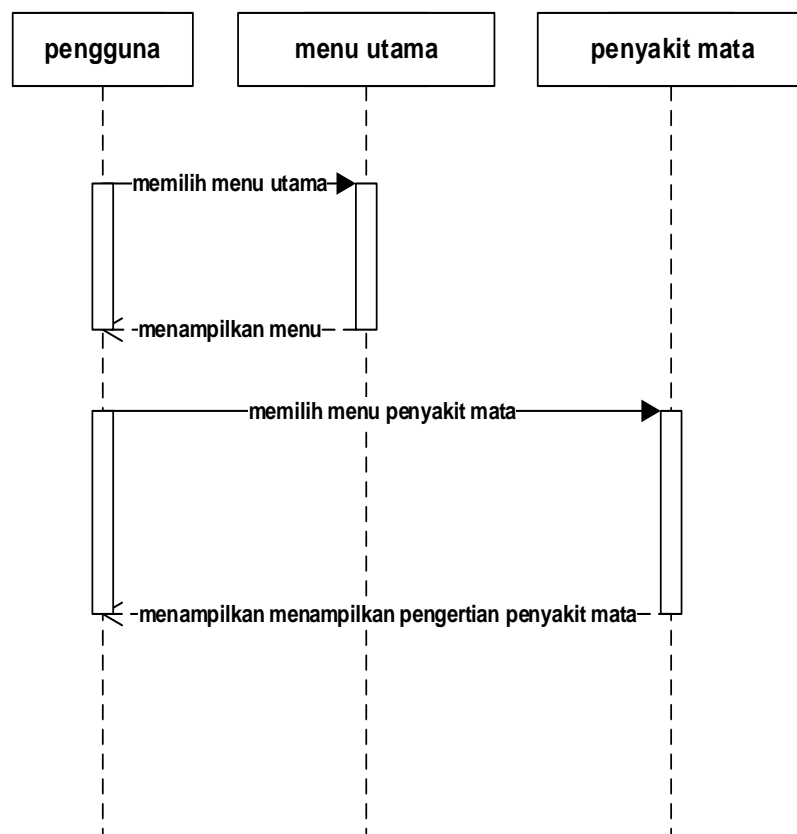


Gambar 3.6 Activity Diagram Profil Pembuat

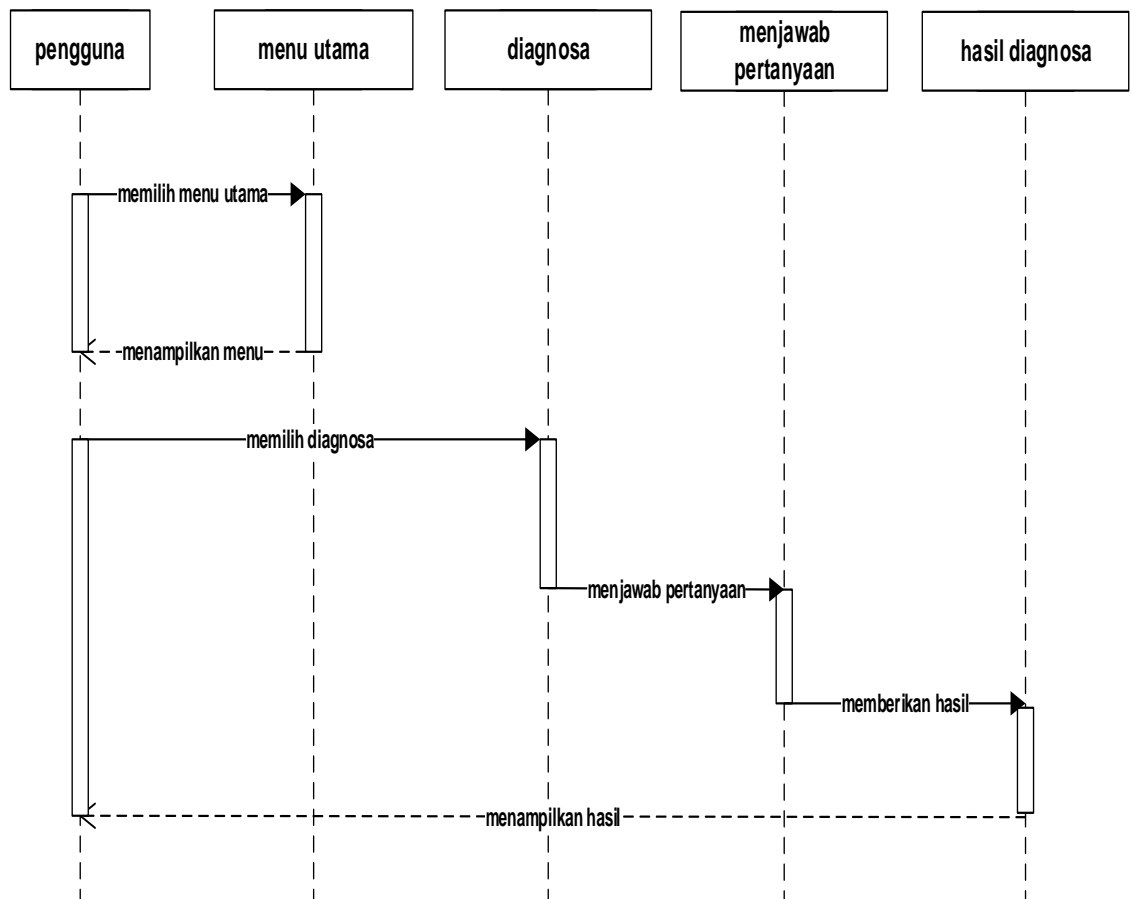
3.6.4. Diagram Sequence

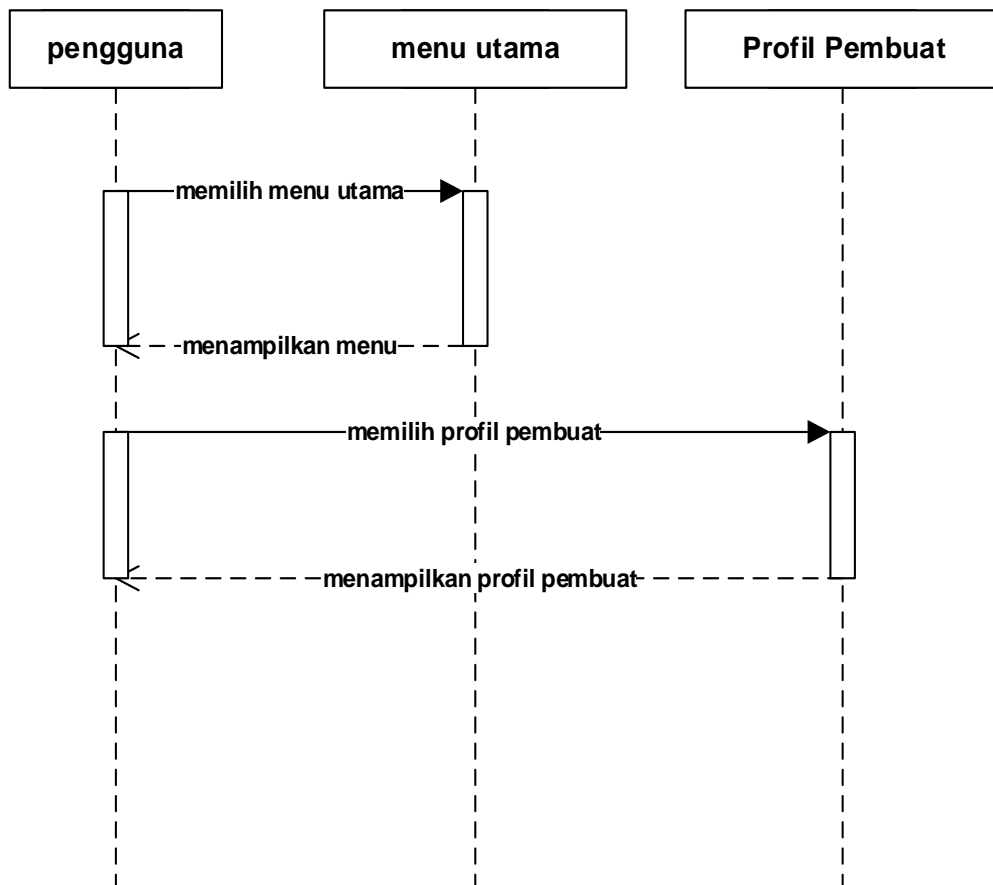
Berikut adalah *diagram sequence* aplikasi metode certainty factor pada penyakit mata:

1) *Diagram Sequence* Penyakit Mata



Gambar 3.7 *Diagram Sequence* Penyakit Mata

2) *Diagram Sequence Diagnosa***Gambar 3.8** *Diagram Sequence Diagnosa*3) *Diagram Sequence Profil Pembuat*



Gambar 3.9 Diagram Sequence Profil Pembuat

3.7. Perancangan Aplikasi

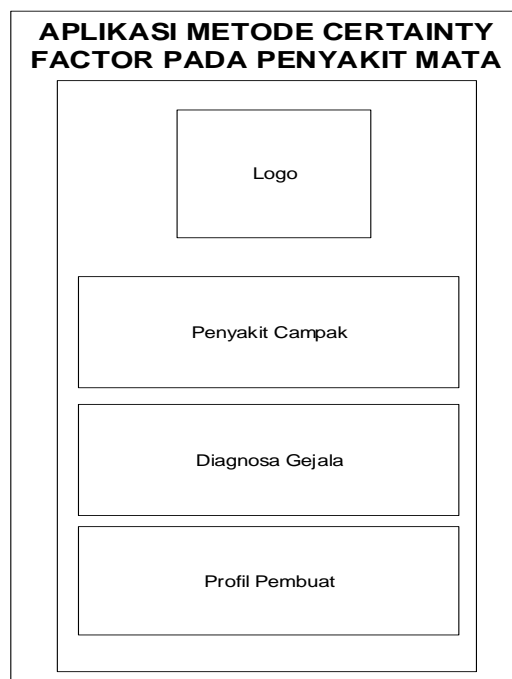
1. Perancangan Antarmuka (*User Interface*)

Perancangan antar muka adalah rancangan yang dilakukan untuk memberikan gambaran aplikasi yang akan ditampilkan secara sederhana kepada pengguna. Diharapkan pengguna yang menggunakan aplikasi ini dapat dengan mudah mengerti fungsi dari tombol yang ada pada aplikasi. Dalam aplikasi metode certainty factor pada penyakit mata ini, terdapat beberapa bagian tampilan yang

memiliki fungsi berbeda pada setiap tombolnya. Fungsi – fungsi dari tombol yang ada pada setiap bagian tampilan akan dijelaskan dan dapat dilihat pada gambar berikut :

1) Rancangan Tampilan Menu Utama

Rancangan tampilan menu utama adalah tampilan yang pertama kali ditampilkan dan memiliki beberapa fungsi untuk menghubungkan ke tampilan lainnya. Tampilan ini disebut dengan tampilan utama, yang dapat digunakan oleh pengguna untuk menuju ke tampilan yang diinginkannya dengan memilih menu yang ada pada tampilan. Menu utama memiliki teks judul dan 3 tombol.



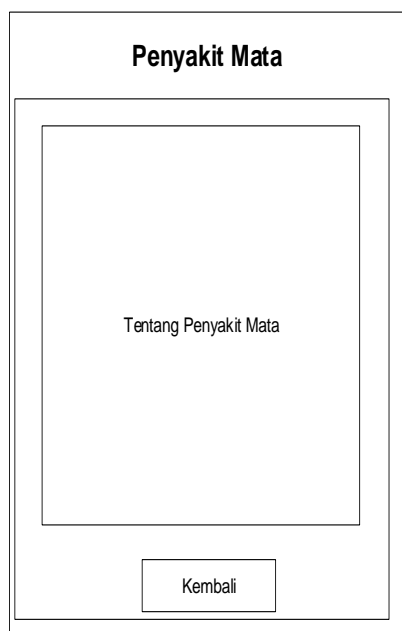
Gambar 3.10 Rancangan Tampilan Menu Utama

Berikut fungsi dari 3 tombol yang ada pada menu utama :

- a) Penyakit mata berfungsi untuk menuju ke tampilan menu pengertian dasar tentang penyakit mata.
- b) Diagnosa berfungsi untuk menuju ke tampilan proses diagnosa apakah sedang terserang penyakit mata atau tidaknya.
- c) Tentang pembuat berfungsi untuk menuju ke tampilan tentang.

2) Rancangan Tampilan *Form* Penyakit Mata

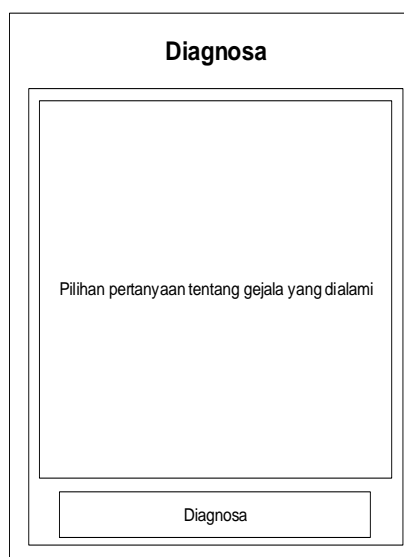
Rancangan tampilan penyakit mata adalah tampilan yang berisikan pengertian dasar tentang penyakit mata yang diderita. Tampilan penyakit mata memiliki teks dan 1 tombol.



Gambar 3.11 Rancangan Tampilan Penyakit Mata

3) Rancangan Tampilan Diagnosa

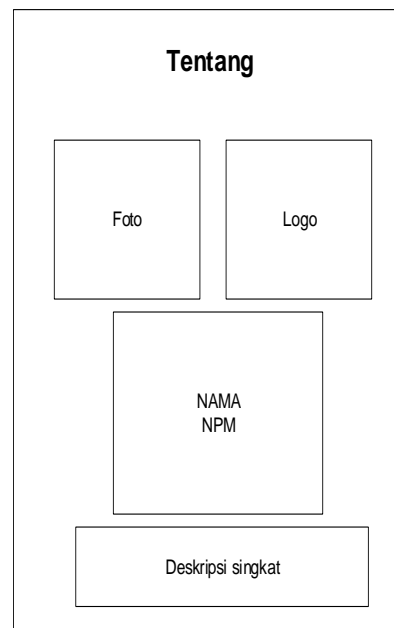
Rancangan tampilan diagnosa adalah tampilan yang berisikan pertanyaan yang harus dijawab oleh pengguna agar dapat mendiagnosa apakah sedang menderita penyakit mata atau tidak. diagnosa memiliki teks pertanyaan dan 1 tombol.



Gambar 3.12 Rancangan Tampilan Diagnosa

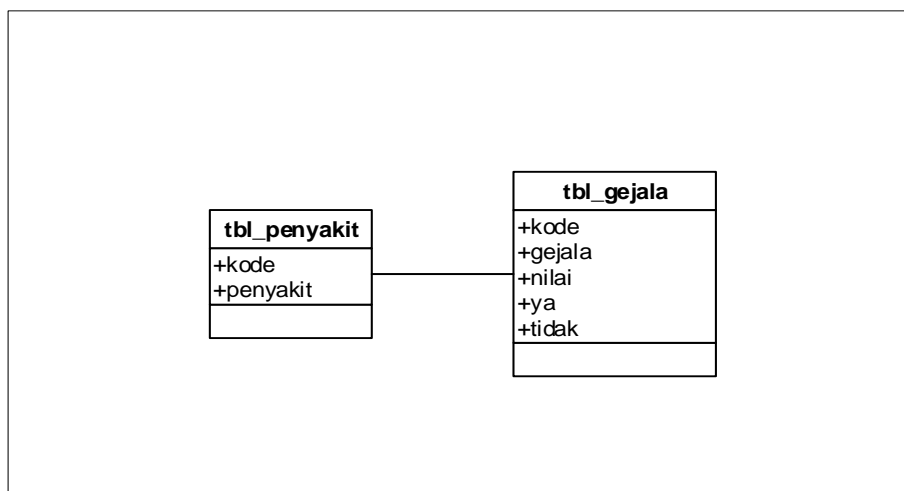
4) Rancangan Tampilan Menu Tentang

Rancangan ini adalah rancangan yang menampilkan informasi dari si pembuat aplikasi metode certainty factor pada penyakit mata.



Gambar 3.13 Rancangan Tampilan Menu Tentang

3.8. Perancangan *Class Diagram*



Gambar 3.14 Class Diagram Aplikasi Pada Penyakit Mata

3.9. Perancangan *Database*

3.9.1. Perancangan Tabel

Struktur file digunakan dalam perancangan sistem untuk menentukan nilai atau tipe data suatu atribut pada file yang terdapat pada database. Pada tahapan perancangan struktur file untuk mempermudah dalam mengetahui suatu nilai atau tipe data yang ada pada file penyimpanan ini akan dijelaskan mengenai perancangan basis data yang akan digunakan. Penyusunan table ini pada dasarnya digunakan untuk memudahkan dalam pemasukan dengan penyimpanan data yang sesuai dengan kelompok dari data atau informasi tersebut.

1) Tabel Gejala

Tabel 3.8 Gejala

Nama field	Type data	Size	Keterangan
Kode	Varchar	10	Primary Key
Gejala	Varchar	255	-
Nilai	Varchar	10	-
Ya	Varchar	100	-
Tidak	Varchar	20	-

2) Tabel Penyakit

Tabel 3.9 Penyakit

Nama_field	Type data	Size	Keterangan
Kode	Varchar	10	Primary key
Penyakit	Varchar	100	Foreign Key

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem Yang Digunakan

Dalam perancangan aplikasi metode certainty factor pada penyakit mata ini, penulis menggunakan program aplikasi yang berbasis android dengan C# sebagai Bahasa pemrogramannya. Aplikasi yang dibuat cukup mudah dalam pengoperasiannya untuk pengguna. Pengoperasiannya hanya perlu melakukan klik pada tombol yang telah disediakan untuk mengakses menu lainnya. Pengguna yang ingin melakukan diagnosa pada jenis penyakit mata yang diderita itu hanya tinggal memilih gejala yang sedang dialaminya dan akan tampil hasil penyakit mata yang sedang diderita dengan mudah.

Tahapan implementasi yang dilakukan untuk menyelesaikan perancangan aplikasi metode certainty factor pada penyakit mata ini diperlukan informasi mengenai penyediaan perangkat keras (*Hardware*) dan perangkat lunak (*Software*).

Berikut disediakan perangkat keras dan perangkat lunak yang dibutuhkan.

4.1.1. Spesifikasi Perangkat Keras

Aplikasi metode certainty factor pada penyakit mata ini, telah diuji pada smartphone dengan spesifikasi perangkat keras sebagai berikut:

- 1) *CPU* : *Qualcomm MSM8909 1,2 GHz, GPU*
- 2) *Memory Internal* : 1 GB RAM, 8 GB ROM
- 3) *Memory External* : 8 GB
- 4) *Operating System* : Android OS, 5.0 (Lollipop)

5) Tipe Layar : *Corning Gorilla Glass 3*

6) Ukuran Layar : *720 x 1280 pixel*

4.1.2. Spesifikasi Perangkat Lunak

Aplikasi ini dijalankan pada perangkat lunak dengan spesifikasi sebagai berikut:

1) Sistem Operasi : *Android OS, 5.0 (Lollipop)*

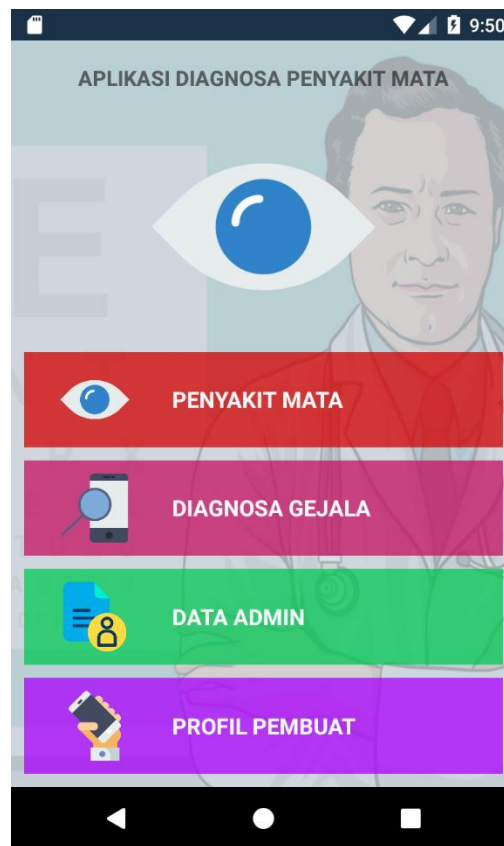
4.2. Tampilan Aplikasi Metode Certainty Factor Pada Penyakit Mata

Tampilan aplikasi metode certainty factor pada penyakit mata ini terdiri dari tampilan pengertian penyakit mata, diagnosa, hasil diagnosa, data gejala, data penyakit, data admin dan tentang. Menu utama berisi menu - menu aplikasi yaitu pengertian penyakit mata, diagnosa, tambah data dan tentang.

Adapun tampilan menu-menu aplikasi metode certainty factor pada penyakit mata adalah sebagai berikut :

4.2.1. Tampilan Menu Utama

Tampilan menu utama terdiri dari pengertian penyakit mata, diagnosa, tambah data dan tentang.



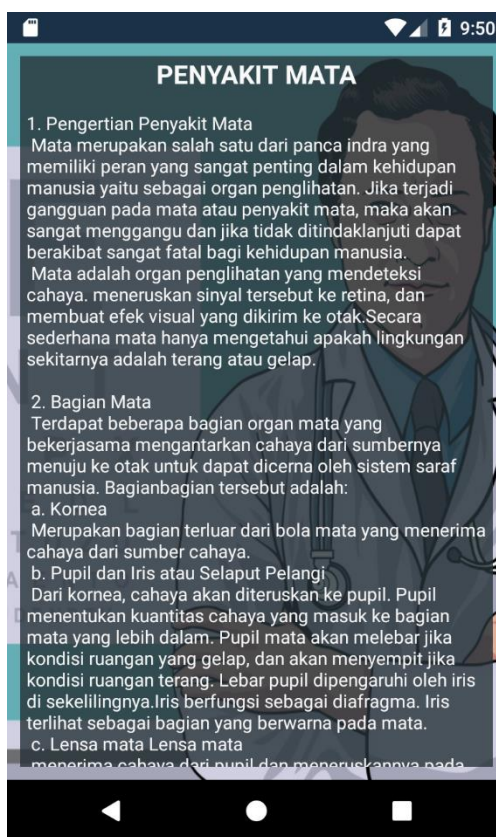
Gambar 4.1 Tampilan Menu Utama

Berikut fungsi dari 4 tombol yang ada pada menu utama :

- 1) Tombol penyakit mata berfungsi untuk melihat pengertian tentang penyakit mata.
- 2) Tombol diagnosa gejala berfungsi untuk melakukan diagnosa gejala pada penderita yang mengalami penyakit pada mata.
- 3) Tombol data admin berfungsi untuk menambahkan data gejala, penyakit dan admin pada aplikasi.
- 4) Tombol profil pembuat berfungsi untuk melihat profil dari pembuat aplikasi diagnosa penyakit mata.

4.2.2. Tampilan Penyakit Mata

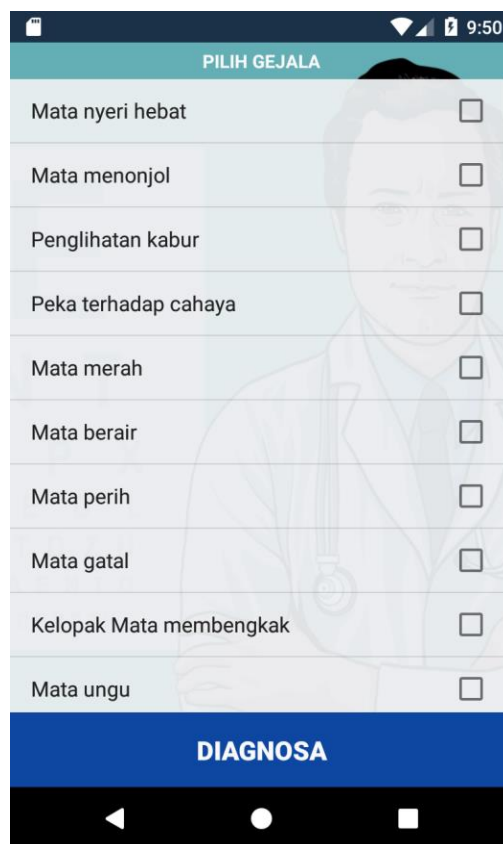
Tampilan penyakit mata terdiri dari sekilas informasi dari penyakit yang terjadi pada mata.



Gambar 4.2 Tampilan Penyakit Mata

4.2.3. Tampilan Diagnosa Gejala

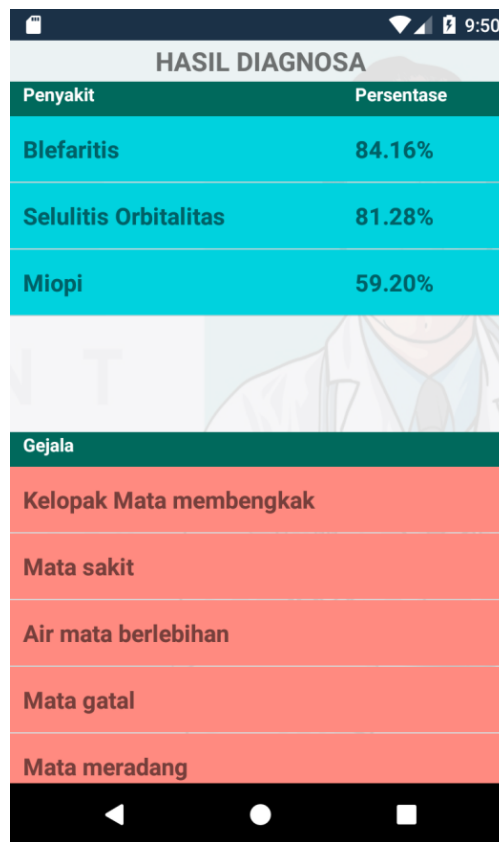
Tampilan diagnosa gejala memiliki fungsi untuk melakukan diagnosa gejala pada penderita yang mungkin terserang penyakit pada matanya.



Gambar 4.3 Tampilan Diagnosa Gejala

4.2.4. Tampilan Hasil Diagnosa

Tampilan hasil diagnosa memiliki fungsi untuk melihat hasil dari diagnosa gejala pengguna sesuai dengan gejala yang dialami pengguna saat ini.



Penyakit	Persentase
Blefaritis	84.16%
Selulitis Orbitalitas	81.28%
Miopi	59.20%

Gejala
Kelopak Mata membengkak
Mata sakit
Air mata berlebihan
Mata gatal
Mata meradang

Gambar 4.4 Tampilan Hasil Diagnosa

4.2.5. Tampilan Detail Penyakit

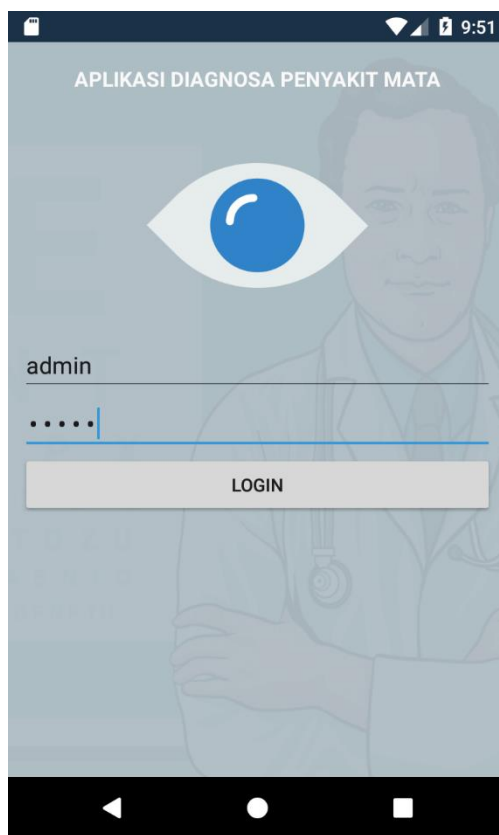
Tampilan detail penyakit memiliki fungsi untuk melihat nilai dan jenis penyakit beserta gambar penderita yang pernah mengalami penyakit tersebut.



Gambar 4.5 Tampilan Detail Penyakit

4.2.6. Tampilan Login

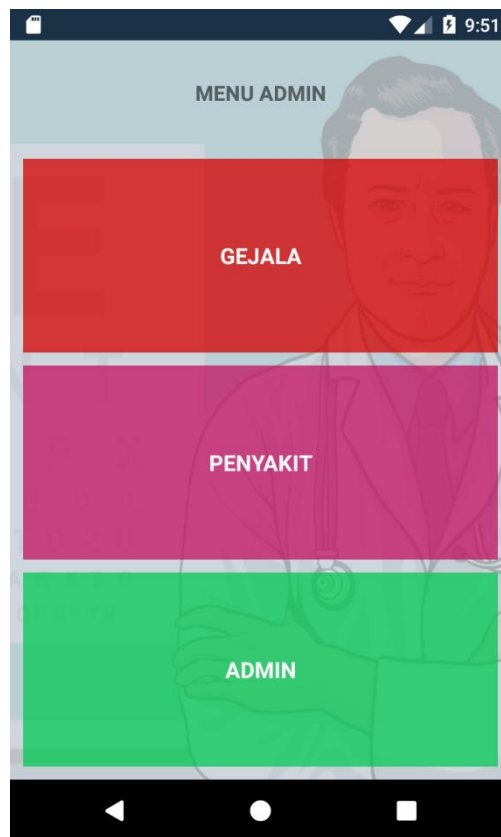
Tampilan login memiliki fungsi untuk masuk kedalam data admin untuk dapat melakukan perubahan pada data yang terdapat pada aplikasi seperti data gejala, data penyakit dan data admin.



Gambar 4.6 Tampilan Login

4.2.7. Tampilan Menu Admin

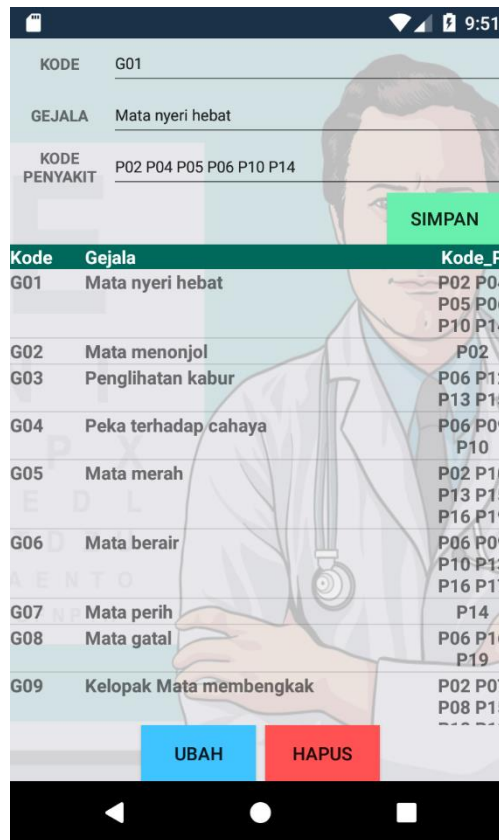
Tampilan menu admin memiliki fungsi untuk menghubungkan ketampilan yang lainnya yaitu tampilan data gejala, data penyakit dan data admin.



Gambar 4.7 Tampilan Menu Admin

4.2.8. Tampilan Data Gejala

Tampilan data gejala memiliki fungsi untuk menambahkan maupun mengurangi data gejala pada aplikasi.



Gambar 4.8 Tampilan Data Gejala

4.2.9. Tampilan Data Penyakit

Tampilan data penyakit memiliki fungsi untuk menambahkan dan mengurangi data penyakit pada aplikasi.

KODE P01

PENYAKIT Xerophthalmia

SOLUSI - Menghindari iklim atau kondisi ruangan yang

SIMPAN

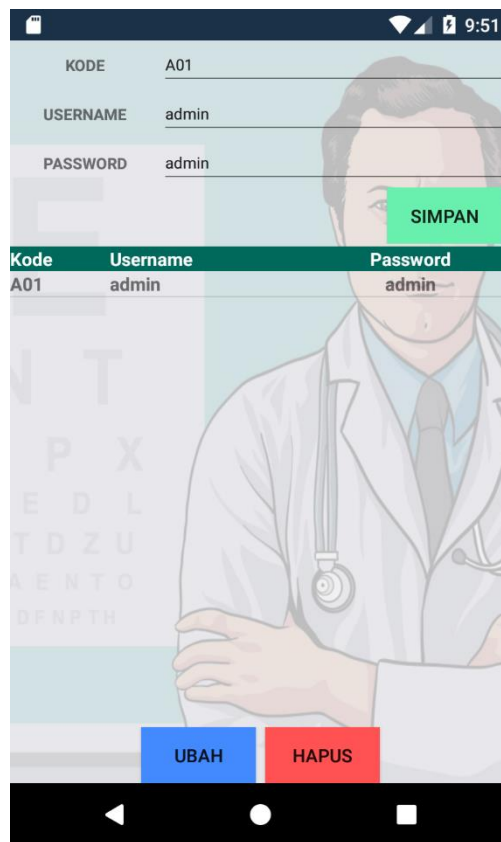
Kode	Penyakit	Solusi
P01	Xerophthalmia	<ul style="list-style-type: none"> - Menghindari iklim atau kondisi ruangan yang kering, Menggunakan alat pembersih atau pelembab udara di dalam ruangan. - Memakai kaca mata pelindung yang dapat memperlambat penguapan air dari permukaan mata. - Menggunakan salep, gel, atau air mata buatan. Kendati demikian, hindari air mata dengan bahan pengawet jika harus menggunakannya lebih

UBAH HAPUS

Gambar 4.9 Tampilan Data Penyakit

4.2.10. Tampilan Data Admin

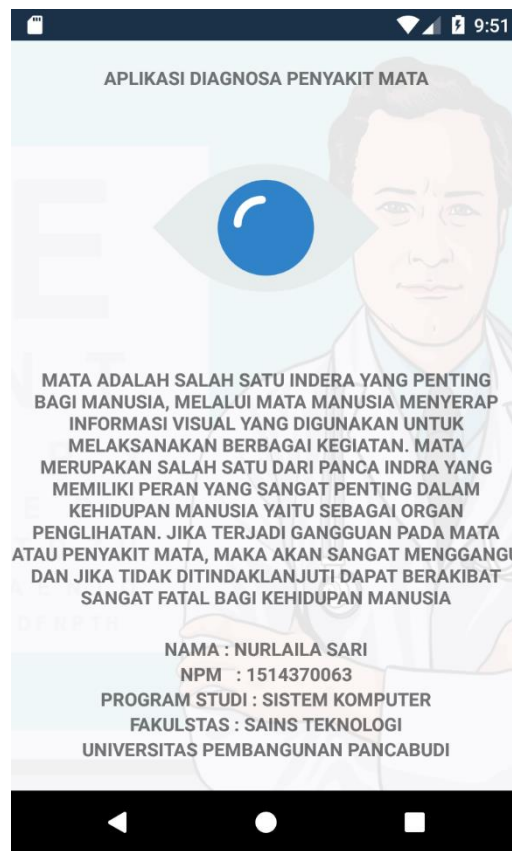
Tampilan data admin memiliki fungsi untuk menambahkan data pengguna yang dapat mengelola aplikasi diagnosa penyakit mata.



Gambar 4.10 Tampilan Data Admin

4.2.11. Tampilan Tentang

Tampilan tentang memiliki fungsi untuk melihat profil pembuat aplikasi diagnosa penyakit mata.

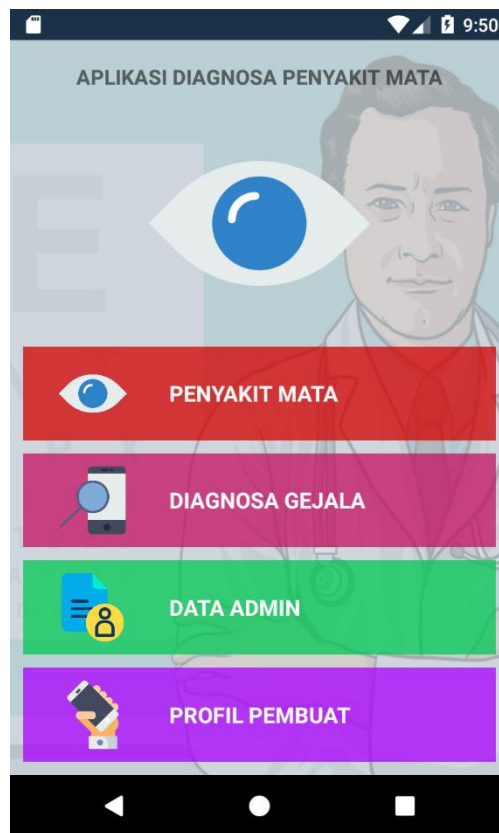


Gambar 4.11 Tampilan Tentang

4.3. Pengujian Aplikasi Metode Certainty Factor Pada Penyakit Mata

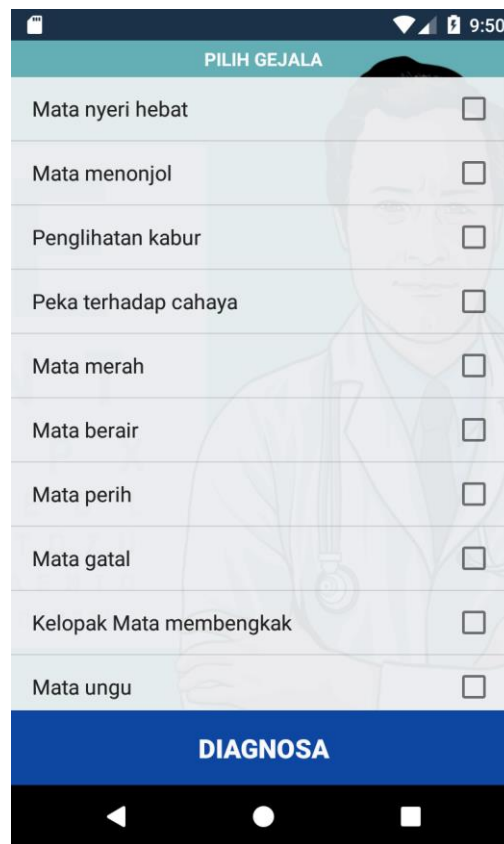
Pengujian aplikasi metode certainty factor pada penyakit mata ini digunakan untuk menguji sistem pada salah satu menu dimana data yang digunakan adalah proses melakukan diagnosa jenis penyakit yang diderita oleh pengguna. Cara menggunakan aplikasi metode certainty factor pada penyakit mata adalah sebagai berikut :

- 1) Langkah awalnya pengguna menjalankan aplikasi dan akan ditampilkan menu utama aplikasi.



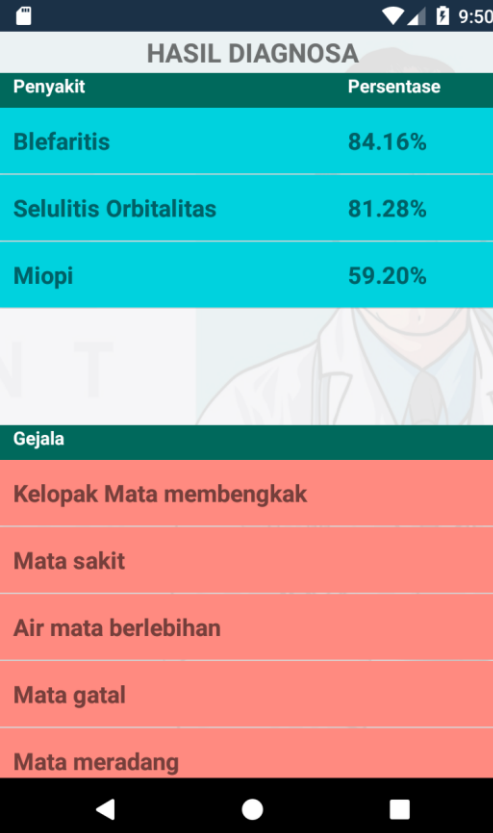
Gambar 4.12 Menu Utama

- 2) Kemudian pengguna melakukan klik pada menu diagnosa gejala untuk melakukan diagnosa jenis penyakit yang diderita pengguna.
- 3) Setelah melakukan klik pada diagnosa gejala, akan tampil tampilan baru yaitu tampilan pertanyaan seputar gejala.



Gambar 4.13 Pilih Gejala

- 4) Setelah pengguna memilih gejala yang dialami, pengguna dapat langsung melakukan klik pada tombol diagnosa untuk dapat melihat hasil jenis penyakit sesuai dengan gejala yang dialami.



The screenshot shows a mobile application interface with a dark blue header bar containing the title "HASIL DIAGNOSA". Below the header is a table with two columns: "Penyakit" (Disease) and "Persentase" (Percentage). The table lists three diseases: Blefaritis (84.16%), Selulitis Orbitalitas (81.28%), and Miopi (59.20%). Below the table is a section titled "Gejala" (Symptoms) with a list of symptoms: Kelopak Mata membengkak, Mata sakit, Air mata berlebihan, Mata gatal, and Mata meradang. The background of the app features a faint illustration of a doctor in a white coat.

Penyakit	Persentase
Blefaritis	84.16%
Selulitis Orbitalitas	81.28%
Miopi	59.20%

Gejala

- Kelopak Mata membengkak
- Mata sakit
- Air mata berlebihan
- Mata gatal
- Mata meradang

Gambar 4.14 Hasil Diagnosa

- 5) Dalam hasil diagnosa terdapat jenis penyakit yang diderita pengguna beserta nilai persentase dari jenis penyakit yang diduga diderita oleh pengguna.
- 6) Untuk melihat lebih lanjut tentang jenis penyakit yang diderita pengguna dapat melakukan klik pada nama jenis penyakit untuk dapat melihat detailnya.



Gambar 4.15 Detail Penyakit

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pembuatan aplikasi metode certainty factor pada penyakit mata, maka didapat beberapa kesimpulan seperti berikut:

- 1) Dalam mengetahui jenis penyakit mata yang diderita oleh pengguna, pengguna hanya tinggal menjalankan aplikasi yang berbasis android dan hanya melakukan klik pada tombol diagnosa gejala dan memilih gejala yang dialami untuk dapat memberikan hasil dari jenis penyakit yang dideritanya.
- 2) Untuk menghasilkan nilai persentase dari gejala yang dialami pengguna, penulis membuat sistem dengan menggunakan bantuan metode *certainty factor* dalam proses pengolahan datanya dengan memberikan nilai pada setiap gejala yang dialami pengguna untuk memberikan hasil dari jenis penyakit dan nilai persentasenya.

5.2. Saran

Berikut adalah saran dari penulis agar aplikasi metode certainty factor pada penyakit mata ini dapat bermanfaat dan dikembangkan menjadi lebih baik lagi :

- 1) Sistem deteksi penyakit yang telah dibangun ini hanya dapat melakukan diagnosa pada penyakit mata saja dan tidak dapat melakukan diagnosa pada penyakit lainnya.
- 2) Aplikasi yang telah dirancang penulis dibuat dengan menggunakan bahasa pemrograman C# dalam menghasilkan aplikasi yang berbasis android, aplikasi yang dihasilkan ini tidak dapat dijalankan pada sistem operasi lainnya selain sistem operasi android dengan *minimum requirement android 5.0*.
- 3) Dalam proses pengolahan data pada aplikasi ini, penulis hanya menggunakan metode *certainty factor* dalam melakukan proses diagnosa gejala yang dialami beserta nilai persentase pada jenis penyakitnya.

DAFTAR PUSTAKA

- Azhar, S., Sari, H. L., & Zulita, L. N. (2014). Sistem Pakar Penyakit Ginjal Pada Manusia. *Jurnal Media Infotama*, 10(1), 16–26.
- Fahnun, B. U., Noviana, R., Prananingrum, L., & Tjioe, E. (2013). Informasi kampus berbasis web pada android. *Seminar Nasional Teknologi Informasi Dan Multimedia 2013*, 25–32.
- Harto, D. (2010). Perancangan Sistem Pakar Untuk Mengidentifikasi Penyakit Dengan Gejala Demam Menggunakan Metode Certainty Factor. *Informatika*, 6, 22–27.
- HAVILUDDIN. (2013). Memahami Penggunaan UML (Unified Modelling Language). *Jurnal Media Infotama*, 9(2), 1–6.
- InfoDatin. (2013). Gangguan Penglihatan dan Kebutaan Republik Indonesia. *Pusat Data Dan Informasi Kementerian Kesehatan RI*.
- Iqbal, Zarlis, & Harumy. (2017). Inovasi Aplikasi Check In Spot nelayan binaan Kecamatan Bagan Deli Medan. *Seminar Nasional APTIKOM (SEMNASTIKOM)*, (November), 1–6.
- Kosidin, & Farizah, R. N. (2016). Pemodelan Aplikasi Mobile Reminder Berbasis Android. *Seminar Nasional Teknologi Informasi Dan Komunikas 2016, 2016(Sentika)*, 271–280.
- Kurniawan, R., & Wardhani, L. K. (2011). Sistem Pakar Untuk Mendiagnosa Penyakit Mata Dengan Metode Bayesian Network. *Sntiki Iii 2011*, 309–315.
- Lars. (2014). *Microsoft® Visual Studio® 2015 Unleashed*. Pearson Education, Inc.
- Maulana, M. R. W. (2017). Pengembangan Aplikasi Android Untuk Studi Bahasa Carakan Madura. *Journal Information Engineering and Educational Technology*, 01, 2549–869.
- Muharom, Cahyana, & Bunyamin. (2013). PENGEMBANGAN APLIKASI SUNDA BERBASIS ANDROID MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT (RAD). *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, 1–11.
- Nirmala, S., & Supartha. (2014). Sistem Pakar Diagnosa Awal Penyakit Kulit Pada Sapi Bali dengan Menggunakan Metode Forward chaining dan Certainty Factor. *Jurnal Ilmiah Teknik Informasi*, 3(Sistem Pakar), 110–117.

- Ongko, E., & Dude. (2013). Perancangan Sistem Pakar Diagnosa Penyakit pada Mata. *STMIK IBBI, II(2)*, 10–17.
- Sari, nur anjas. (2014). Sistem Pakar Mendiagnosa Jenis Penyakit Stroke Menggunakan Metode Certainty Factor. *Jurnal Ilmiah Teknik Informasi*, 2
- Surahman, S., & Setiawan, E. B. (2018). Aplikasi Mobile Driver Online Berbasis Android Untuk Perusahaan Rental Kendaraan. *Jurnal ULTIMA InfoSys*, 8(1)
- Tahir, M., & Rasheed, M. (2011). iPhone & Android Client For Mobile Webshop Framework. *Linnaeus University*.
- Urva, G., & Siregar, H. F. (2015). Pemodelan UML E-Marketing Minyak Goreng. *Jtsi*, 1(9), 92–101.
- Wamiliana, Deprianto, & Aristoteles. (2015). Pengembangan Sistem Pakar Berbasis Web Mobile Untuk Mengidentifikasi Penyebab Kerusakan Telepon Seluler Dengan Menggunakan Metode Forward Dan Backward Chaining. *Jurnal Komutasi, I(Sistem Pakar)*, 1–9.
- Yulansari, K., & Sukandi. (2013). Sistem Informasi Pengelolaan Data Iuran Badan Pembantu Penyelenggaraan Pendidikan Sekolah Menengah Kejuruan Negeri 2 Donorojo. *Seminar Riset Unggulan Nasional Informatika Dan Komputer FTI UNSA 2013*, 2(1), 5–13.
- Badawi, A. (2018). Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." *IT Journal Research and Development* 2.1 (2017): 1-11.
- Bahri, S. (2018). Metodologi Penelitian Bisnis Lengkap Dengan Teknik Pengolahan Data SPSS. Penerbit Andi (Anggota Ikapi). Percetakan Andi Offset. Yogyakarta.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Fitriani, W., Rahim, R., Oktaviana, B., & Siahaan, A. P. U. (2017). Vernam Encrypted Text in End of File Hiding Steganography Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 214-219.
- Hardinata, R. S. (2019). Audit Tata Kelola Teknologi Informasi menggunakan Cobit 5 (Studi Kasus: Universitas Pembangunan Panca Budi Medan). *Jurnal Teknik dan Informatika*, 6(1), 42-45.
- E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 1(1).

- 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.
- 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).
- Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. *Int. J. Sci. Res. Sci. Technol*, 3(6), 504-509.
- 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.
- Muttaqin, Muhammad. "ANALISA PEMANFAATAN SISTEM INFORMASI E-OFFICE PADA UNIVERSITAS PEMBANGUNAN PANCA BUDI MEDAN DENGAN MENGGUNAKAN METODE UTAUT." *Jurnal Teknik dan Informatika* 5.1 (2018): 40-43.
- Ramadhan, Z., Zarlis, M., Efendi, S., & Siahaan, A. P. U. (2018). Perbandingan Algoritma Prim dengan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek (Shortest Path Problem). *JURIKOM (Jurnal Riset Komputer)*, 5(2), 135-139.
- 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.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu* 10.2 (2018): 1899-1902.

LISTING PROGRAM

1. Admin

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace App_Mata.mata
{
    [Activity(Label = "_Admin")]
    public class _Admin : Activity
    {
        dbClass db;
        List<tbl_admin> dataAdmin = new
List<tbl_admin>();
        List<tbl_admin> lstSource = new
List<tbl_admin>();
        ListView lstData;
        string Ubah;
        protected override void
OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            SetContentView(Resource.Layout.Admin)
            ;
            db = new dbClass(this);
            dataAdmin = db.selectAdmin();
            var edit1 =
            FindViewById<EditText>(Resource.Id.edi
tText1);
            var edit2 =
            FindViewById<EditText>(Resource.Id.edi
tText2);
            var edit3 =
            FindViewById<EditText>(Resource.Id.edi
tText3);
            lstData =
            FindViewById<ListView>(Resource.Id.lis
tView1);
            var btn1 =
            FindViewById<Button>(Resource.Id.butto
n1);
            var btn2 =
            FindViewById<Button>(Resource.Id.butto
n2);
            var btn3 =
            FindViewById<Button>(Resource.Id.butto
n3);
```



```

loadData();
lstData.ItemClick += (s, e) =>
{
    var txt1lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView1);
    var txt2lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView2);
    var txt3lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView3);
    edit1.Text = txt1lst.Text;
    edit2.Text = txt2lst.Text;
    edit3.Text = txt3lst.Text;
    Ubah = txt1lst.Text;
};

btn1.Click += delegate
{
    tbl_admin Admin = new
tbl_admin()
{
    Kode = edit1.Text,
    Username = edit2.Text,
    Password = edit3.Text
};
    db.insertAdmin(Admin);
    loadData();
};

btn2.Click += delegate
{
    tbl_admin Admin = new
tbl_admin()
{
    Kode = edit1.Text,
    Username = edit2.Text,
    Password = edit3.Text
};
    db.updateAdmin(Admin, Ubah);
    loadData();
};

btn3.Click += delegate
{
    db.deleteAdmin(Ubah);
    loadData();
};
}

public void loadData()
{
    lstSource = db.selectAdmin();
    var adapter = new
lstClassAdmin(this, lstSource);
    lstData.Adapter = adapter;
}
}
}

4. Detail

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;

```

```

using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace App_Mata.mata
{
    [Activity(Label = "_detail")]
    public class _detail : Activity
    {
        dbClass db;
        List<tbl_penyakit> lstSource = new
List<tbl_penyakit>();
        string kode, Nilai;
        TextView txt1, txt2, txt3;
        ImageView img1;
        protected override void
OnCreate(Bundle savedInstanceState)
        {
            kode =
Intent.GetStringExtra("Kode") ?? "Data
not available";
            Nilai =
Intent.GetStringExtra("Nilai") ?? "Data
not available";

base.OnCreate(savedInstanceState);

SetContentView(Resource.Layout.detail);
            txt1 =
findViewById<TextView>(Resource.Id.te
xtView1);
            txt2 =
findViewById<TextView>(Resource.Id.te
xtView2);

```

```

txt3 =
findViewById<TextView>(Resource.Id.te
xtView3);
            img1 =
findViewById<ImageView>(Resource.Id.
imageView1);
            txt2.Text = "Dengan Persentase = "
+ Nilai;
            db = new dbClass(this);
            loadData();

            var myDrawableName =
txt1.Text.Replace(" ", "");
            var myPackageName =
Application.PackageName;
            var resourceType = "drawable";

            var resourceId =
Application.Resources.GetIdentifier(myDr
awableName.ToLowerInvariant(),
resourceType, myPackageName);

            img1.SetImageResource(resourceId);
            img1.LayoutParameters.Height =
400;
        }

        public void loadData()
        {
            lstSource =
db.selectPenyakit_ID(kode);
            txt1.Text = lstSource[0].Penyakit;
            txt3.Text = lstSource[0].Solusi;

```

```

        //Toast.MakeText(this,
lstSource[0].Solusi,
ToastLength.Long).Show();
    }
}
}

```

2. Gejala

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace App_Mata.mata
{
    [Activity(Label = "_Gejala")]
    public class _Gejala : Activity
    {
        dbClass db;
        List<tbl_gejala> dataGejala = new
List<tbl_gejala>();
        List<tbl_gejala> lstSource = new
List<tbl_gejala>();
        ListView lstData;
        string Ubah;

```

```

        protected override void
        OnCreate(Bundle savedInstanceState)
        {
            db = new dbClass(this);
            dataGejala = db.selectGejala();

            base.OnCreate(savedInstanceState);

            SetContentView(Resource.Layout.Gejala);
            var edit1 =
            FindViewById<EditText>(Resource.Id.edi
tText1);
            var edit2 =
            FindViewById<EditText>(Resource.Id.edi
tText2);
            var edit3 =
            FindViewById<EditText>(Resource.Id.edi
tText3);
            lstData =
            FindViewById<ListView>(Resource.Id.lis
tView1);
            var btn1 =
            FindViewById<Button>(Resource.Id.butto
n1);
            var btn2 =
            FindViewById<Button>(Resource.Id.butto
n2);
            var btn3 =
            FindViewById<Button>(Resource.Id.butto
n3);
            loadData();
            lstData.ItemClick += (s, e) =>
            {

```

```

        var txt1lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView1);
        var txt2lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView2);
        var txt3lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView3);
        edit1.Text = txt1lst.Text;
        edit2.Text = txt2lst.Text;
        edit3.Text = txt3lst.Text;
        Ubah = txt1lst.Text;
    };

    btn1.Click += delegate
    {
        tbl_gejala Gejala = new
tbl_gejala()
        {
            Kode = edit1.Text,
            Gejala = edit2.Text,
            Kode_P = edit3.Text
        };
        db.insertGejala(Gejala);
        loadData();
    };

    btn2.Click += delegate
    {
        tbl_gejala Gejala = new
tbl_gejala()
        {
            Kode = edit1.Text,
            Gejala = edit2.Text,
            Kode_P = edit3.Text
        };
        db.updateGejala(Gejala, Ubah);
        loadData();
    };

    btn3.Click += delegate
    {
        db.deleteGejala(Ubah);
        loadData();
    };
}

public void loadData()
{
    lstSource = db.selectGejala();
    var adapter = new
lstClassGejala(this, lstSource);
    lstData.Adapter = adapter;
}
}
}

5. Login

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.Database.Sqlite;
using Android.OS;

```

```

using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace App_Mata.mata
{
    [Activity(Label = "_login")]
    public class _login : Activity
    {
        dbClass db;
        //SQLiteDatabase sqliteDB;
        List<tbl_admin> Admin = new
List<tbl_admin>();
        protected override void
        OnCreate(Bundle savedInstanceState)
        {
            db = new dbClass(this);
            //sqliteDB = db.WritableDatabase;

base.OnCreate(savedInstanceState);

SetContentView(Resource.Layout.Login);
        var txt1 =
        FindViewById<EditText>(Resource.Id.edi
tText1);
        var txt2 =
        FindViewById<EditText>(Resource.Id.edi
tText2);
        var btn =
        FindViewById<Button>(Resource.Id.butto
n1);
        btn.Click += delegate
        {

```

```

Admin =
db.loginAdmin(txt1.Text, txt2.Text);
        if (Admin.Count == 1)
        {
            StartActivity(typeof(_menuadmin));
            Admin.Clear();
            Finish();
        }
        else
        {
            Toast.MakeText(this,
"Username atau Password salah..",
ToastLength.Short).Show();
        }
    };
}
}
}
}
}

```

6. Menu Admin

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;

```

```

namespace App_Mata.mata
{
    [Activity(Label = "_menuadmin")]
    public class _menuadmin : Activity
    {
        protected override void
        OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            SetContentView(Resource.Layout.menuad
            min);
            var menu1 =
            FindViewById<LinearLayout>(Resource.I
            d.menu1);
            var menu2 =
            FindViewById<LinearLayout>(Resource.I
            d.menu2);
            var menu3 =
            FindViewById<LinearLayout>(Resource.I
            d.menu4);

            menu1.Click += delegate
            {
                StartActivity(typeof(_Gejala));
            };
            menu2.Click += delegate
            {
                StartActivity(typeof(_Penyakit));
            };
            menu3.Click += delegate

```

```

        {
            StartActivity(typeof(_Admin));
        };
    }
}

```

7. Pakar

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using Android.App;

using Android.Content;

using Android.OS;

using Android.Runtime;

using Android.Views;

using Android.Widget;

```

```

namespace App_Mata.mata
{
    [Activity(Label = "_Pakar")]
    public class _Pakar : Activity
    {
        dbClass db;

        ListView lstData;

```

```

        Button btn1;

        //string RegKodeP;

        //string RegKodeP_Pilih;

        List<tbl_gejala> lstSource = new
List<tbl_gejala>();

        //List<tbl_penyakit> lstSource2 =
new List<tbl_penyakit>();

        Dictionary<string, double> Kode_ =
new Dictionary<string, double>();

        Dictionary<string, string> Kode__ =
new Dictionary<string, string>();

        //Dictionary<string, string> Kode___
= new Dictionary<string, string>();

        //List<string> Penyakit_ = new
List<string>();

        protected override void
OnCreate(Bundle savedInstanceState)

        {

            db = new dbClass(this);

base.OnCreate(savedInstanceState);

SetContentView(Resource.Layout.Pakar);

            lstData =
findViewById<ListView>(Resource.Id.lis
tView1);

            btn1 =
findViewById<Button>(Resource.Id.butto
n1);

            loaddata();

            db.deleteTempALL();

            db.deleteHasilALL();

            db.deleteTempALL_G();

            lstData.ItemClick +=
OnListItemClick;

            btn1.Click += Btn1_Click;

        }

        private void Btn1_Click(object
sender, EventArgs e)

        {

            //var pilih = Kode__;

            //var hitung = pilih.Count;

            //for (int i = 0; i < hitung; i++)

            //{

                // RegKodeP_Pilih =
RegKodeP_Pilih +
pilih.Values.ElementAt(i);

            //}

            //var pilih2 = db.selectGejala();

            //string pattern = RegKodeP;

```

```

//string input = RegKodeP_Pilih;

//foreach (Match match in
Regex.Matches(input, pattern))

//{

//
ListBox1.Items.Add(match.Value.ToStrin
g);

//  karakterspam +=
match.ToString;

//}

//Label3.Text =
karakterspam.Count;

//List<float> aaa = new
List<float>();

//var pilih = db.selectTemp_GG();

//var hitung = pilih.Count;

//for (int i = 0; i < hitung; i++)

//{

//  var i1 =
db.selectTemp_GG_2(pilih[i].Kode_P);

//  var i2 =
db.selectTemp_PP_2(pilih[i].Kode_P);

//  float ia =
float.Parse(i1[0].Total) /
float.Parse(i2[0].Total);

//  aaa.Add(ia);

//}

```

```

//var ib = aaa.Sum();

//var ic = aaa.Count();

//for (int ip = 0; ip < ic; ip++)

//{

//  float id =
float.Parse(aaa[ip].ToString()) /
float.Parse(ib.ToString()) * 100;

//  var ll =
db.selectPenyakit_ID(pilih[ip].Kode_P);

//  tbl_hasil hsl = new tbl_hasil()

//  {

//    Kode = ll[0].Kode,

//    Kode_P = ll[0].Penyakit,

//    Nilai_P =
id.ToString("00.0")

//  };

//  db.insertHasil(hsl);

//}

Hitung();

StartActivity(typeof(_Pakar_Hasil));

Finish();

}

void Hitung()

{

```



```

var hitung2 = Kode__.Count;

for (int i = 0; i < hitung2; i++)

{

    var nilai =
db.selectTemp_ID(Kode__.Values.Elemen
tAt(i));

    var penyakit =
db.selectPenyakit_ID(Kode__.Values.Ele
mentAt(i));

    double asd = 0;

    double asdKurang = 0;

    for (int ia = 0; ia < nilai.Count;
ia++)

    {

        var gejalanya =
db.selectGejala_ID(nilai[0].Kode_G);

        tbl_temp_G G_hasil = new
tbl_temp_G()

        {

            Kode =
gejalanya[0].Kode_P,

            Gejala =
gejalanya[0].Gejala

        };

        db.insertTemp_G(G_hasil);

    }

```

```

if (nilai.Count >= 2)

{

    asdKurang = 1 - nilai[0].Nilai;

    asd = nilai[0].Nilai +
nilai[1].Nilai * asdKurang;

    double a1 = 0;

    double b1 = 0;

    double a2 = 0;

    double b2 = 0;

    double c1 = 0;

    double c2 = 0;

    double d1 = 0;

    double d2 = 0;

    double e1 = 0;

    double e2 = 0;

    double f1 = 0;

    double f2 = 0;

    double g1 = 0;

    double g2 = 0;

    int cass = nilai.Count;

    switch (cass)

    {

        case 2:

```

```

tbl_hasil hsl0 = new
tbl_hasil()
{
    Kode =
penyakit[0].Kode,
    Kode_P =
penyakit[0].Penyakit,
    Nilai_P = (asd *
100).ToString("00.00")
};
db.insertHasil(hsl0);
break;
case 3:
    a2 = 1 - asd;
    a1 = asd + nilai[2].Nilai
* a2;
tbl_hasil hsl = new
tbl_hasil()
{
    Kode =
penyakit[0].Kode,
    Kode_P =
penyakit[0].Penyakit,
    Nilai_P = (a1 *
100).ToString("00.00")
};
db.insertHasil(hsl);
break;
case 4:
    a2 = (1 - asd);
    a1 = asd + nilai[2].Nilai
* a2;
    b2 = (1 - a1);
    b1 = a1 + nilai[3].Nilai *
b2;
tbl_hasil hsl2 = new
tbl_hasil()
{
    Kode =
penyakit[0].Kode,
    Kode_P =
penyakit[0].Penyakit,
    Nilai_P = (b1 *
100).ToString("00.00")
};
db.insertHasil(hsl2);
break;
case 5:
    a2 = (1 - asd);
    a1 = (asd +
(nilai[2].Nilai) * a2);
    b2 = (1 - a1);
    b1 = (a1 + (nilai[3].Nilai)
* b2);

```

```

                c2 = (1 - b1);
                c1 = (b1 + (nilai[4].Nilai)
* c2);
                tbl_hasil hsl3 = new
tbl_hasil()
                {
                    Kode =
                penyakit[0].Kode,
                    Kode_P =
                penyakit[0].Penyakit,
                    Nilai_P = (c1 *
100).ToString("00.00")
                };
                db.insertHasil(hsl3);
                break;
            case 6:
                a2 = (1 - asd);
                a1 = asd + (nilai[2].Nilai)
* a2;
                b2 = (1 - a1);
                b1 = a1 + (nilai[3].Nilai)
* b2;
                c2 = (1 - b1);
                c1 = b1 + (nilai[4].Nilai)
* c2;
                d2 = (1 - c1);
                d1 = c1 + (nilai[5].Nilai)
* d2;
                d1 = c1 + (nilai[5].Nilai)
* d2;
                tbl_hasil hsl4 = new
tbl_hasil()
                {
                    Kode =
                penyakit[0].Kode,
                    Kode_P =
                penyakit[0].Penyakit,
                    Nilai_P = (d1 *
100).ToString("00.00")
                };
                db.insertHasil(hsl4);
                break;
            case 7:
                a2 = (1 - asd);
                a1 = asd + (nilai[2].Nilai)
* a2;
                b2 = (1 - a1);
                b1 = a1 + (nilai[3].Nilai)
* b2;
                c2 = (1 - b1);
                c1 = b1 + (nilai[4].Nilai)
* c2;
                d2 = (1 - c1);
                d1 = c1 + (nilai[5].Nilai)
* d2;

```

```

                e2 = (1 - c1);
                e1 = c1 + (nilai[6].Nilai)
* e2;

                tbl_hasil hsl5 = new
tbl_hasil()
                {
                    Kode =
penyakit[0].Kode,
                    Kode_P =
penyakit[0].Penyakit,
                    Nilai_P = (e1 *
100).ToString("00.00")
                };
                db.insertHasil(hsl5);
                break;
            case 8:
                a2 = (1 - asd);
                a1 = asd + (nilai[2].Nilai)
* a2;
                b2 = (1 - a1);
                b1 = a1 + (nilai[3].Nilai)
* b2;
                c2 = (1 - b1);
                c1 = b1 + (nilai[4].Nilai)
* c2;
                d2 = (1 - c1);

                d1 = c1 + (nilai[5].Nilai)
* d2;
                e2 = (1 - c1);
                e1 = c1 + (nilai[6].Nilai)
* e2;
                f2 = (1 - c1);
                f1 = c1 + (nilai[7].Nilai)
* f2;
                tbl_hasil hsl6 = new
tbl_hasil()
                {
                    Kode =
penyakit[0].Kode,
                    Kode_P =
penyakit[0].Penyakit,
                    Nilai_P = (f1 *
100).ToString("00.00")
                };
                db.insertHasil(hsl6);
                break;
            case 9:
                a2 = (1 - asd);
                a1 = asd + (nilai[2].Nilai)
* a2;
                b2 = (1 - a1);
                b1 = a1 + (nilai[3].Nilai)
* b2;

```

```

                c2 = (1 - b1);
                }
                c1 = b1 + (nilai[4].Nilai)
                //Toast.MakeText(this,
* c2;                cass.ToString(),
                ToastLength.Short).Show();
                d2 = (1 - c1);
                }
                d1 = c1 + (nilai[5].Nilai)
                nilai.Clear();
* d2;                asd = 0;
                e2 = (1 - c1);
                asdKurang = 0;
                e1 = c1 + (nilai[6].Nilai)
                }
* e2;                }
                f2 = (1 - c1);
                }
                f1 = c1 + (nilai[7].Nilai)
* f2;
                void OnListItemClick(object sender,
                AdapterView.ItemClickEventArgs e)
                {
                var listView = sender as ListView;
                var t = lstSource[e.Position];
                if (Kode_.ContainsKey(t.Kode) ==
                true)
                {
                Kode =
                penyakit[0].Kode,
                Kode_P =
                penyakit[0].Penyakit,
                Nilai_P = (g1 *
                100).ToString("00.00")
                };
                db.insertHasil(hsl7);
                break;
                }
                else
                {

```

```

        Kode_.Add(t.Kode, 0.8 *
t.Nilai);

        //Kode__.Add(t.Kode,
t.Kode_P);

        db.insertTemp(t.Kode,
t.Kode_P, 0.8 * t.Nilai);
    }
}

public void loaddata()
{
    lstSource = db.selectGejala();

    List<string> spinnerArray = new
List<string>();

    var hitung = lstSource.Count;

    for (int i = 0; i < hitung; i++)
    {
spinnerArray.Add(lstSource[i].Gejala);

    }

    var adapter = new
ArrayAdapter(this,
Android.Resource.Layout.SimpleListItem
MultipleChoice, spinnerArray);

    lstData.ChoiceMode =
ChoiceMode.Multiple;

    lstData.Adapter = adapter;

```

```

var kodeP = db.selectPenyakit();

var hitung2 = kodeP.Count;

for (int i = 0; i < hitung2; i++)
{
    //RegKodeP = RegKodeP + "|" +
kodeP[i].Kode;

    Kode__.Add(i.ToString(),
kodeP[i].Kode);
}

//var kodeG = db.selectGejala();

//var hitung3 = kodeG.Count;

//for (int i = 0; i < hitung3; i++)

//{

//    //RegKodeP = RegKodeP + "|"
+ kodeP[i].Kode;

//    Kode__.Add(i.ToString(),
kodeG[i].Kode);

//}

//RegKodeP =
RegKodeP.Substring(0, RegKodeP.Length
- 1);
}
}
}

```

9. Pakar Hasil

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace App_Mata.mata
{
    [Activity(Label = "_Pakar_Hasil")]
    public class _Pakar_Hasil : Activity
    {
        dbClass db;
        List<tbl_hasil> lstSource = new
List<tbl_hasil>();
        List<tbl_temp_G> lstSource2 = new
List<tbl_temp_G>();
        ListView lstdata, lstdata2;
        protected override void
OnCreate(Bundle savedInstanceState)
        {

base.OnCreate(savedInstanceState);

SetContentView(Resource.Layout.Pakar_
Hasil);
```

```
        lstdata =
FindViewById<ListView>(Resource.Id.lis
tView1);
        lstdata2 =
FindViewById<ListView>(Resource.Id.lis
tView2);
        var txt =
FindViewById<TextView>(Resource.Id.te
xtView3);
        var lnr =
FindViewById<LinearLayout>(Resource.I
d.linearLayout2);
        db = new dbClass(this);
        loadData();
        if (lstdata.Count == 0)
        {
            txt.Visibility =
ViewStates.Visible;
            lnr.Visibility =
ViewStates.Gone;
        }
        else
        {
            txt.Visibility =
ViewStates.Gone;
            lnr.Visibility =
ViewStates.Visible;
        }

        lstdata.ItemClick += (s, e) =>
        {
            var txt1lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView1);
```

```

        var txt3lst =
e.View.FindViewById<TextView>(Resou
rce.Id.textView3);
        var i = new Intent(this,
typeof(_detail));
        i.PutExtra("Kode", txt3lst.Text);
        i.PutExtra("Nilai", txt1lst.Text);
        StartActivity(i);
        //Finish();
    };
}

public void loadData()
{
    lstSource = db.selectHasil_ID();
    var adapter = new
lstClassHasil(this, lstSource);
    lstdata.Adapter = adapter;

    lstSource2 = db.selectTemp_G();
    var adapter2 = new
lstClassHasil_2(this, lstSource2);
    lstdata2.Adapter = adapter2;
}
}
}

```

3. Pengertian

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace App_Mata.mata
{
    [Activity(Label = "_Pengertian")]
    public class _Pengertian : Activity
    {
        protected override void
        OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            SetContentView(Resource.Layout.Pengerti
an);
        }
    }
}

```

10. Penyakit

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;

```



```

using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace App_Mata.mata
{
    [Activity(Label = "_Penyakit")]
    public class _Penyakit : Activity
    {
        dbClass db;
        List<tbl_penyakit> dataPenyakit =
new List<tbl_penyakit>();
        List<tbl_penyakit> lstSource = new
List<tbl_penyakit>();
        ListView lstData;
        string Ubah;
        protected override void
OnCreate(Bundle savedInstanceState)
        {
            db = new dbClass(this);
            dataPenyakit = db.selectPenyakit();

base.OnCreate(savedInstanceState);

SetContentView(Resource.Layout.Penyaki
t);
            var edit1 =
findViewById<EditText>(Resource.Id.edi
tText1);
            var edit2 =
findViewById<EditText>(Resource.Id.edi
tText2);

```

```

            var edit3 =
findViewById<EditText>(Resource.Id.edi
tText3);
            lstData =
findViewById<ListView>(Resource.Id.lis
tView1);
            var btn1 =
findViewById<Button>(Resource.Id.butto
n1);
            var btn2 =
findViewById<Button>(Resource.Id.butto
n2);
            var btn3 =
findViewById<Button>(Resource.Id.butto
n3);
            loadData();
            lstData.ItemClick += (s, e) =>
            {
                var txt1lst =
e.View.findViewById<TextView>(Resou
rce.Id.textView1);
                var txt2lst =
e.View.findViewById<TextView>(Resou
rce.Id.textView2);
                var txt3lst =
e.View.findViewById<TextView>(Resou
rce.Id.textView3);
                edit1.Text = txt1lst.Text;
                edit2.Text = txt2lst.Text;
                edit3.Text = txt3lst.Text;
                Ubah = txt1lst.Text;
            };
            btn1.Click += delegate

```

```

        {
            tbl_penyakit Penyakit = new
tbl_penyakit()
        {
            Kode = edit1.Text,
            Penyakit = edit2.Text
        };
        db.insertPenyakit(Penyakit);
        loadData();
    };
    btn2.Click += delegate
    {
        tbl_penyakit Penyakit = new
tbl_penyakit()
        {
            Kode = edit1.Text,
            Penyakit = edit2.Text
        };
        db.updatePenyakit(Penyakit,
Ubah);
        loadData();
    };
    btn3.Click += delegate
    {
        db.deletePenyakit(Ubah);
        loadData();
    };
}

public void loadData()
{
    IstSource = db.selectPenyakit();
    var adapter = new
IstClassPenyakit(this, IstSource);

```

```

IstData.Adapter = adapter;

```

```

}

```

```

}

```

```

}

```

11. Tentang

```

using System;

```

```

using System.Collections.Generic;

```

```

using System.Linq;

```

```

using System.Text;

```

```

using Android.App;

```

```

using Android.Content;

```

```

using Android.OS;

```

```

using Android.Runtime;

```

```

using Android.Views;

```

```

using Android.Widget;

```

```

namespace App_Mata.mata

```

```

{

```

```

    [Activity(Label = "_Tentang")]

```

```

    public class _Tentang : Activity

```

```

    {

```

```

        protected override void

```

```

        OnCreate(Bundle savedInstanceState)

```

```

        {

```

```

        base.OnCreate(savedInstanceState);

```

```

        SetContentView(Resource.Layout.Tentang

```

```

        );

```

```

    }

```

