



**PERANCANGAN APLIKASI SISTEM PAKAR UNTUK  
MENDIAGNOSA BAKTERI ESCHERICHIA COLI  
(E-COLI) PADA AIR MINUM ISI ULANG  
DENGAN METODE CERTAINTY FACTOR**

Disusun dan Di ajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh  
Gelar Sarjana Komputer Pada Fakultas Sains dan Teknologi  
Universitas Pembangunan Panca Budi  
Medan

---

**SKRIPSI**

---

**OLEH**

**NAMA : ARIE SURYA DINATA**  
**N.P.M : 1414370113**  
**PROGRAM STUDI : SISTEM KOMPUTER**

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

## **ABSTRAK**

**ARIE SURYA DINATA**  
**PERANCANGAN APLIKASI SISTEM PAKAR UNTUK**  
**MENDIAGNOSA BAKTERI ESCHERICHIA COLI**  
**(E-COLI) PADA AIR MINUM ISI ULANG**  
**DENGAN METODE CERTAINTY FACTOR**  
**Tahun 2019**

Seiring dengan semakin meningkatnya kebutuhan air minum pada saat ini, banyak pula depot atau tempat pengisian ulang air minum yang bermunculan. Sebagian dari masyarakat belum mengetahui apakah air tersebut layak untuk dikonsumsi atau tidak. Air minum yang di konsumsi sehari-hari tersebut tanpa melalui proses pemasakan atau perebusan terlebih dahulu. Depot yang bermunculan saat ini banyak yang tidak mencantumkan surat dari dinas kesehatan setempat yang menyatakan bahwasanya air minum yang dijual layak untuk dikonsumsi. Biaya yang mahal dan urusan yang sangat sulit membuat para pemilik depot air minum tidak memperdulikan hal itu yang sebenarnya harus dimiliki. Salah satu hal yang dapat mengancam kesehatan melalui air minum adalah adanya bakteri *Escherichia Coli (E-Coli)*. Untuk mengetahui apakah air minum yang dikonsumsi mengandung bakteri *E-Coli* tidak mudah, karena ukurannya yang sangat kecil dan tidak kasat mata. Salah satu akibat yang dapat ditimbulkan dari bakteri *E-Coli* ini seperti sakit perut, muntaber, diare, tekanan darah tinggi, bahkan gangguan ginjal. Berdasarkan dari penyakit-penyakit yang bersumber dari air minum tersebut membuat penulis ingin membuat suatu program aplikasi sistem pakar yang dapat mendiagnosa air minum dengan cara melihat tanda-tanda pada air minum yang dijual dari depot air minum yang telah banyak dipasaran. Perancangan aplikasi yang akan dibuat berbentuk aplikasi *mobile* yang dapat digunakan oleh semua orang dan dapat dipasang pada semua *smartphone* yang menggunakan sistem operasi *android*. Sehingga dapat lebih mudah dalam menilai suatu kualitas air minum dari depot tersebut.

***Kata kunci*** : *Certainty Factor, Android, E-Coli, Sistem pakar.*

## DAFTAR ISI

Halaman

<b>HALAMAN JUDUL</b>	
<b>LEMBAR PERSETUJUAN</b>	
<b>LEMBAR PENGESAHAN</b>	
<b>ABSTRAK</b>	
<b>KATA PENGANTAR</b> .....	i
<b>DAFTAR ISI</b> .....	iii
<b>DAFTAR GAMBAR</b> .....	v
<b>DAFTAR TABEL</b> .....	vi
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
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
1.6. Metode Penelitian .....	5
1.7. Sistematika Penulisan .....	5
<b>BAB II LANDASAN TEORI</b> .....	<b>7</b>
2.1. Bakteri <i>Escherichia Coli</i> .....	7
2.2. Sistem Pakar .....	8
2.3. Metode <i>Certainty Factor</i> .....	11
2.4. Aplikasi <i>Mobile</i> .....	14
2.5. <i>Android</i> .....	15
2.6. Definisi <i>Visual Studio 2017</i> .....	16
2.7. Pengertian Basis Data (Sistem Basis Data) .....	18
2.8. <i>Database SQLite</i> .....	21
2.9. <i>Unified Modeling Language (UML)</i> .....	22
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM</b> .....	<b>28</b>
3.1. Analisis Yang Sedang Berjalan .....	28
3.2. Penerapan Metode <i>Certainty Factor</i> .....	29
3.3. Analisis Kebutuhan.....	33
3.4. <i>Use case</i> .....	34
3.5. Activity Diagram Aplikasi Mendiagnosa Bakteri E-Coli.....	37
3.6. <i>Diagram Sequence</i> Aplikasi Mendiagnosa Bakteri E-Coli .....	38
3.7. <i>Class Diagram</i> Aplikasi Mendiagnosa Bakteri E-Coli .....	39
3.8. Perancangan Aplikasi .....	40
3.9. Perancangan <i>Database</i> .....	46

<b>BAB IV IMPLEMENTASI SISTEM .....</b>	<b>48</b>
4.1. Penerapan Metode <i>Certainty Factor</i> .....	48
4.2. Implementasi Sistem yang Digunakan .....	52
4.3. Pengujian Aplikasi Diagnosa Bakteri E-Coli .....	57

<b>BAB V PENUTUP .....</b>	<b>60</b>
5.1. Kesimpulan .....	60
5.2. Saran .....	60

**DAFTAR PUSTAKA**

**BIOGRAFI**

**LAMPIRAN - LAMPIRAN**

## DAFTAR TABEL

	Halaman
Tabel 2.1 Nilai CF .....	13
Tabel 2.9.1 Simbol <i>Use Case Diagram</i> .....	23
Tabel 2.9.2 Simbol <i>ActivityDiagram</i> .....	25
Tabel 2.9.3 Simbol <i>Sequence Diagram</i> .....	26
Tabel 2.9.4 Simbol <i>Class Diagram</i> .....	27
Tabel 3.2 Pertanyaan Diagnosa .....	29
Tabel 3.2 Nilai MB dan MD.....	30
Tabel 3.4.2 Definisi Aktor.....	35
Tabel 3.4.3 Definisi <i>Use case</i> .....	35
Tabel 3.4.4.1 Skenario <i>Use case</i> Diagnosa .....	36
Tabel 3.4.4.2 Skenario <i>Use case</i> Riwayat Diagnosa .....	36
Tabel 3.4.4.3 Skenario <i>Use case</i> Tentang Bakteri E-Coli.....	37
Tabel 3.4.4.4 Skenario <i>Use case</i> Profil Pembuat .....	37
Tabel 3.8.2 Kriteria.....	47
Tabel 3.8.3 Riwayat.....	47
Tabel 4.1 Pertanyaan Diagnosa .....	48
Tabel 4.1. Nilai MB Dan MD.....	49

## DAFTAR GAMBAR

	Halaman
Gambar 1.6. Paradigma <i>Waterfall (Classic Life Cycle)</i> .....	5
Gambar 2.1. Struktur Skematis Sistem Pakar .....	11
Gambar 2.6. Tampilan Visual Studio 2017 .....	18
Gambar 3.4.1 <i>Use Case Diagram</i> Aplikasi Mendiagnosa Bakteri E-Coli	34
Gambar 3.5 <i>Activity Diagram</i> .....	38
Gambar 3.6 <i>SquenceDiagram</i> Aplikasi Mendiagnosa Bakteri E-Coli .	39
Gambar 3.7 <i>ClassDiagram</i> Aplikasi Mendiagnosa Bakteri E-Coli .....	39
Gambar 3.8.1.1 Rancangan Tampilan Menu Utama .....	41
Gambar 3.8.1.2 Rancangan Tampilan Tentang Bakteri E-Coli.....	42
Gambar 3.8.1.3 Rancangan Tampilan Diagnosa .....	43
Gambar 3.8.1.4 Rancangan Tampilan Riwayat Diagnosa.....	44
Gambar 3.8.1.5 Rancangan Tampilan Menu Profil.....	45
Gambar 3.8.1.6 Struktur Arsitektur <i>Navigasi</i> .....	46
Gambar 4.2.2.1.1 Tampilan Menu Utama.....	53
Gambar 4.2.2.1.2 Tampilan Tentang Bakteri E-Coli.....	54
Gambar 4.2.2.1.3 Tampilan Diagnosa .....	55
Gambar 4.2.2.1.4 Tampilan Hasil Diagnosa.....	56
Gambar 4.2.2.1.5 Tampilan Riwayat Diagnosa .....	56
Gambar 4.2.2.1.6 Tampilan Profil .....	57
Gambar 4.2.2.2 Diagnosa .....	58
Gambar 4.2.2.2 Hasil Diagnosa.....	59

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Air merupakan materi penting dalam kehidupan. Semua makhluk hidup membutuhkan air. Bagi manusia, kebutuhan akan air adalah mutlak karena 70% zat pembentuk tubuh manusia terdiri dari air. Kebutuhan air untuk keperluan sehari-hari berbeda untuk setiap tempat dan setiap tingkatan kehidupan. Biasanya semakin tinggi taraf kehidupan, semakin meningkat pula jumlah kebutuhan air. Diantara kegunaan-kegunaan air tersebut yang sangat penting adalah kebutuhan untuk minum termasuk untuk memasak (Mairizki, 2017).

Seiring dengan semakin meningkatnya kebutuhan air minum pada saat ini, banyak pula depot-depot atau tempat pengisian ulang air minum yang bermunculan. Terkadang banyak sebagian dari masyarakat belum mengetahui apakah air tersebut layak untuk dikonsumsi atau tidak. Alasannya karena air yang di konsumsi sehari-hari tersebut tanpa melalui proses pemasakan atau perebusan terlebih dahulu. Depot-depot yang bermunculan saat ini pun banyak yang tidak mencantumkan surat dari dinas kesehatan setempat yang menyatakan bahwasanya air minum yang dijual layak untuk dikonsumsi. Biaya yang mahal dan urusan yang sangat sulit membuat para pemilik depot air minum tidak memperdulikan hal terpenting yang sebenarnya harus dimiliki.

Salah satu hal yang dapat mengancam kesehatan melalui air minum adalah adanya bakteri *Escherichia Coli* (E-Coli). Untuk mengetahui apakah air minum yang dikonsumsi mengandung bakteri E-Coli tidak mudah, karena

ukurannya yang sangat kecil dan tidak kasat mata. Salah satu akibat yang dapat ditimbulkan dari bakteri E-Coli ini seperti sakit perut, muntaber, diare, tekanan darah tinggi, bahkan gangguan ginjal.

Berdasarkan dari penyakit-penyakit yang bersumber dari air minum tersebut membuat penulis ingin membuat suatu program aplikasi sistem pakar yang dapat mendiagnosa air minum dengan cara melihat tanda-tanda pada air minum yang dijual dari depot air minum yang telah banyak dipasaran. Alasan penulis ingin membuat suatu sistem ini adalah agar lebih dapat membantu dalam menilai suatu depot air minum yang telah banyak dipasaran dengan mudah menggunakan aplikasi. Penggunaan aplikasi ini hanya tinggal memilih nama depot air minum yang telah diinputkan sebelumnya dan kemudian menjawab pertanyaan yang diberikan oleh aplikasi untuk dapat melihat hasil kualitas dari depot air minum tersebut.

Perancangan aplikasi yang akan dibuat berbentuk aplikasi *desktop* yang dapat digunakan oleh semua orang. Berdasarkan latar belakang diatas maka penulis tertarik untuk memilih judul "**Perancangan Aplikasi Sistem Pakar Untuk Mendiagnosa *Bakteri Escherichia Coli* (E-Coli) Pada Air Minum Isi Ulang Dengan Metode *Certainty Factor***".

## **1.2 Rumusan Masalah**

Adapun permasalahan yang dihadapi dalam perancangan sistem pakar untuk mendiagnosa *bakteri escherichia coli* (e-coli) pada air minum isi ulang ini adalah :

1. Bagaimana cara mengetahui apakah depot air minum tersebut memiliki kualitas yang baik ?
2. Bagaimana cara kerja metode *certainty factor* dalam melihat kualitas air minum tersebut dan menentukan apakah air minum mengandung bakteri *E-Coli*?

### 1.3 Batasan Masalah

Dalam perancangan sistem pakar diagnosa untuk mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang ini, penulis membatasi masalah sebagai berikut :

1. Aplikasi yang dibuat hanya menentukan apakah air minum mengandung bakteri E-Coli.
2. Perancangan sistem menggunakan metode *certainty factor* dalam mengetahui kualitas depot air minum di kota Medan.
3. Sistem yang akan dibangun dibuat dengan menggunakan bahasa pemrograman *Visual Basic .Net 2012* dan *database MySQL*

### 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai penulis dalam merancang sistem pakar untuk mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang ini adalah :

1. Agar lebih mempermudah dalam menentukan kualitas air minum yang dijual pada depot air minum di kota Medan.
2. Agar lebih terhindar dari bahaya bakteri E-Coli yang dapat membuat penyakit yang bersumber dari air minum.

## 1.5 Manfaat Penelitian

Merancang aplikasi sistem pakar untuk mendiagnosa *bakteri escherichia coli* (e-coli) pada air minum isi ulang ini bermanfaat bagi penulis dan masyarakat luasantara lain :

1. Memudahkan masyarakat dalam mendiagnosa bakteri E-Coli tanpa harus ke lab untuk memastikan adanya bakteri E-Coli pada air minum yang akan dikonsumsi.
2. Aplikasi sebagai alternatif dalam mendiagnosa bakteri E-Coli pada air minum yang dibeli.

## 1.6 Metode Penelitian

### 1.6.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode deskriptif. Adapun teknik pengumpulan data dilakukan dengan cara sebagai berikut:

#### 1. Observasi

Pengumpulan data dengan melakukan pengamatan pada masyarakat yang yang belum mengetahui tentang bahaya bakteri *escherichia coli* (e-coli) pada air minum.

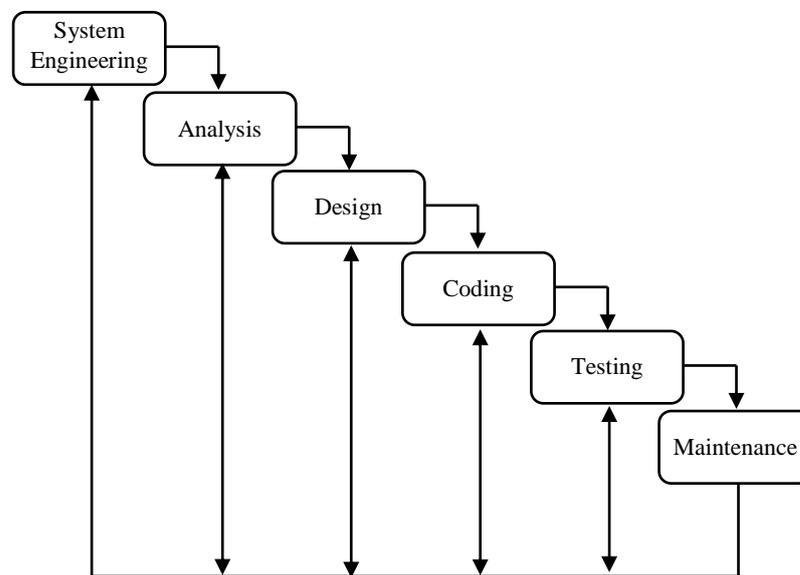
#### 2. Studi Literatur

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

### 1.6.2 Metode Pengembangan Sistem

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*.

Paradigma *Waterfall* dapat dilihat pada gambar berikut:



Gambar 1.6 Paradigma *Waterfall* (Classic Life Cycle)

Sumber : Iqbal, 2017

## 1.7 Sistematika Penulisan

Adapun struktur penulisan pada masing-masing bab dalam laporan skripsi ini adalah sebagai berikut:

### BAB I PENDAHULUAN

Membahas Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat Penelitian, Metodologi Penelitian dan Sistematika Penulisan.

## **BAB II LANDASAN TEORI**

Memaparkan teori-teori yang didapat dari sumber-sumber yang relevan untuk digunakan sebagai panduan dalam penelitian serta penyusunan skripsi. Dalam bab ini dijelaskan dengan penjelasan tentang sistem pakar, flowchart, activity diagram, use case diagram, class diagram, air minum dan bakteri *escherichia coli* (e-coli).

## **BAB III PERANCANGAN SISTEM**

Menjelaskan tentang gambaran sistem serta deskripsi dari hasil analisis sistem yang akan dijadikan sebagai petunjuk untuk perancangan sistem selanjutnya. Pada bab ini dijelaskan tentang perancangan sistem yang akan dirancang. Didalamnya terdapat rancangan aplikasi sistem pakar untuk mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum.

## **BAB IV IMPLEMENTASI SISTEM**

Bab ini menguraikan langkah-langkah dalam implementasi sistem, disertai dengan komponen-komponen kebutuhan sistem. Dalam bab ini dijelaskan dengan menampilkan data output dari sistem pakar diagnosa bakteri *escherichia coli* (e-coli) yang telah dirancang.

## **BAB V PENUTUP**

Mengemukakan kesimpulan yang diambil dari hasil penelitian dan perancangan sistem, serta saran-saran untuk pengembangan aplikasi sistem pakar selanjutnya, agar dapat dilakukan perbaikan-perbaikan dimasa yang akan datang.

## BAB II

### LANDASAN TEORI

#### 2.1 Bakteri *Escherichia Coli*

Bakteri *E.coli* adalah bakteri yang paling banyak digunakan sebagai indikator sanitasi karena bakteri ini adalah bakteri komensal pada usus manusia, umumnya merupakan patogen penyebab penyakit dan relatif tahan hidup di air sehingga dapat dianalisis keberadaannya di dalam air yang sebenarnya bukan medium yang ideal untuk pertumbuhan bakteri. *E.coli* dapat dipindahsebarakan melalui air yang tercemar tinja atau air seni orang yang menderita infeksi pencernaan, sehingga dapat menular pada orang lain. *E.coli* keluar dari tubuh bersama tinja dalam jumlah besar serta mampu bertahan sampai beberapa minggu. Kelangsungan hidup dan replikasi *E.coli* di lingkungan membentuk koliform. *E.coli* tidak tahan terhadap keadaan kering atau desinfektan biasa. Bakteri ini akan mati pada suhu 60°C selama 30 menit. (Widyaningsih, 2016)

Bakteri *coliform* adalah golongan bakteri *intestinal*, yaitu hidup didalam saluran pencernaan manusia. Bakteri *coliform* adalah bakteri indikator keberadaan bakteri patogenik lain. Lebih tepatnya, bakteri *coliform fekal* adalah bakteri indikator adanya pencemaran bakteri patogen. Penentuan *coliform fekal* menjadi indikator pencemaran dikarenakan jumlah koloninya pasti berkorelasi positif dengan keberadaan bakteri patogen. Selain itu, mendeteksi *coliform* jauh lebih murah, cepat, dan sederhana dari pada mendeteksi bakteri patogenik lain. Contoh bakteri *coliform* adalah, *Escherichia coli* dan *Enterobacter aerogenes*. Jadi, *coliform* adalah indikator kualitas air. Makin sedikit kandungan *coliform*, artinya

kualitas air semakin baik. Bakteri kelompok koliform meliputi semua bakteri berbentuk batang, gram negatif, tidak membentuk spora dan dapat memfermentasi laktosa dengan memproduksi gas dan asam pada suhu 37°C dalam waktu kurang dari 48 jam. Adapun bakteri E.Coli selain memiliki karakteristik seperti bakteri koliform pada umumnya juga dapat menghasilkan senyawa indole didalam air pepton yang mengandung asam *aminotriptofan*, serta tidak dapat menggunakan natrium sitrat sebagai satu-satunya sumber karbon.(Widyaningsih, 2016)

## 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).

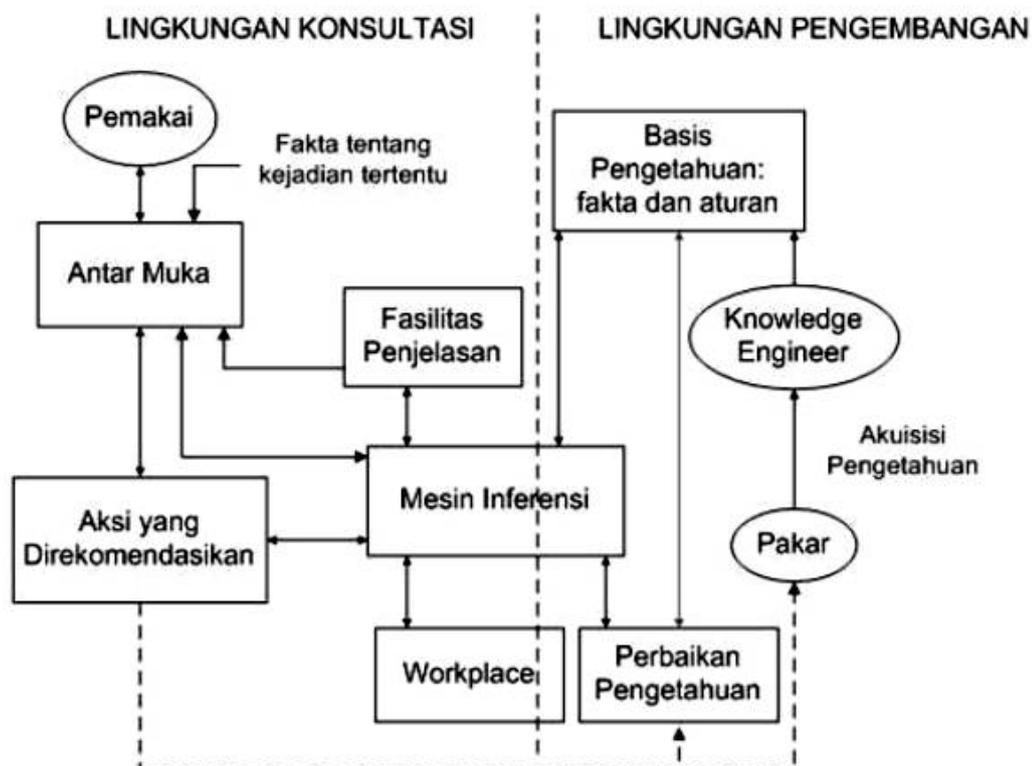
Proses inferensi dilakukan dalam suatu modul yang disebut *inference engine* (mesin inferensi). Ketika representasi pengetahuan pada bagian *knowledge base*(dasar pengetahuan)telah lengkap, atau paling tidak telah berada pada level cukup akurat, maka referensi pengetahuan tersebut telah siap digunakan. 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)

### 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) = *ertainty 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 diantaranya 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)

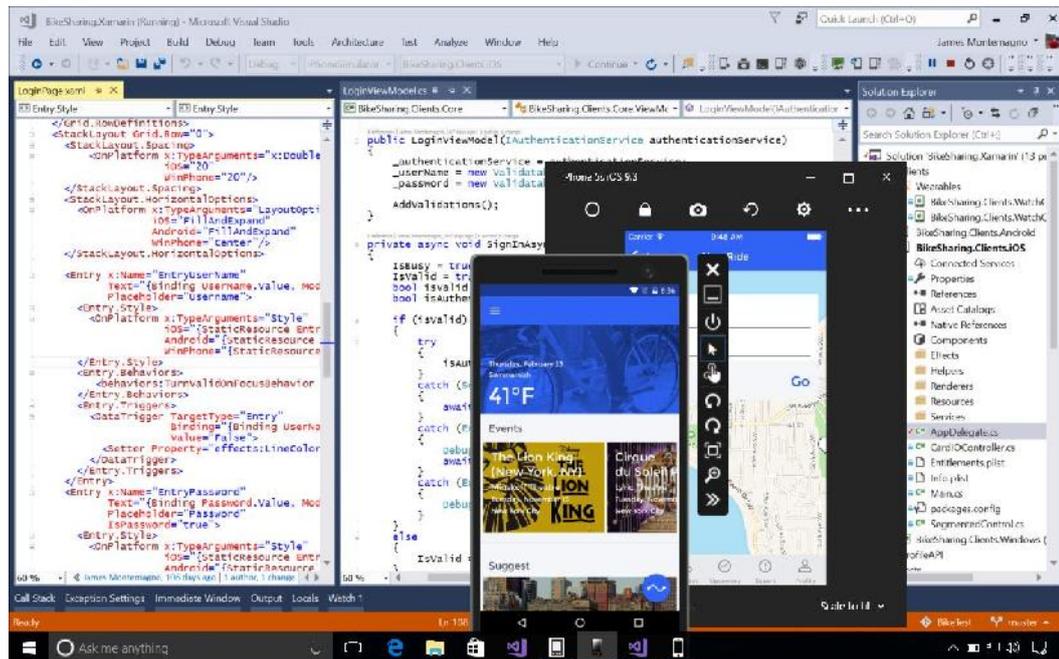
1. *Application Framework* yang memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. *Dalvik Virtual Machine*, yaitu mesin virtual yang dioptimalkan untuk perangkat *Mobile*.
3. *Graphic Library*, yang mendukung grafik 2D dan 3D berdasarkan OpenGL Library.
4. *Media Supported*, yang mendukung 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 Definisi Visual Studio 2017**

*Microsoft Visual Studio* adalah lingkungan pengembangan berdasarkan *.NET Framework* dengan C#, F#, *Visual Basic*, dan C++. Jika Anda ingin untuk membangun aplikasi untuk berbagai sistem operasi, *platform*, atau perangkat selain PC, Anda harus menggunakan alat pengembangan eksklusif dan kerangka kerja asli pada platform tertentu. Dalam beberapa tahun terakhir, *Microsoft* telah mengubah strateginya secara signifikan, membuka *platform* lain, merangkul *open source*, dan lebih fokus pada layanan *cloud*. Bahkan, perusahaan telah melakukan investasi yang signifikan untuk membawa teknologi, platform, alat pengembang, kerangka kerja, dan layanan ke sistem operasi lain seperti Linux dan Mac OS, dan ke khalayak non-*Microsoft* dengan berfokus pada layanan lebih dari di masa lalu.

Dalam strategi ini, *.NET Core*, sumber terbuka modular, *runtime* lintas platform, memungkinkan pengembang C# untuk menulis aplikasi yang berjalan di *Windows*, *Linux*, dan *Mac*. (Alessandro, 2017)

Dengan *Xamarin*, Anda dapat menulis aplikasi seluler yang berjalan di *Android*, *iOS*, dan *Windows* dengan basis kode C# tunggal. SQL Server 2016 sekarang memiliki pratinjau yang berjalan di Linux tonggak revolusioner untuk *Microsoft*. Pratinjau *Visual Studio* untuk *Mac* saat ini tersedia, dan sepenuhnya memungkinkan pengembang C# untuk menulis aplikasi lintas platform dengan *.NET Core* dan *Xamarin* di *Mac OS*. Dalam visi lintas platform dan lintas perangkat ini, awan bahkan lebih penting. Kenyataannya, *Azure* menghuni semua layanan baru dan yang ada yang ditawarkan *Microsoft*, dan tumbuh sesuai dengan apa yang diminta pasar misalnya, menyelenggarakan wadah *Docker* di *Linux*. Sebagai lingkungan pengembangan utama dari *Microsoft*, *Visual Studio 2017* sangat cocok dengan dunia pertama-*cloud-first mobile* ini. Pengembang dapat menggunakan *Visual Studio 2017* untuk membuat aplikasi yang berjalan di platform apa pun dan perangkat apa pun dengan bahasa dan kerangka kerja pilihan mereka. Misalnya, *Visual Studio 2017* memungkinkan Anda untuk menulis aplikasi *Node.js*, aplikasi *iOS* dan *Android* asli, dan aplikasi web yang berjalan di *Linux* dan *Mac OS*. Kabar baiknya adalah Anda, sebagai pengembang, masih dapat menggunakan alat canggih yang sama seperti yang Anda ketahui, seperti *debugger*, *IntelliSense*, dan *profiler* terhadap semua platform pengembangan yang didukung. (Alessandro, 2017)



**Gambar 2.6. Tampilan Visual Studio 2017**

Sumber : (*microsoft.com*)

## 2.7 Pengertian Basis Data (Sistem Basis Data)

DBMS adalah suatu koleksi dari datayang 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. *DataIsolation*.
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.

## 2.8 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 disain 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)

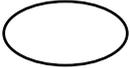
## **2.9 Unified Modeling Language (UML)**

### **2.9.1 Use Case Diagram**

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana suatu aktivitas berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa kegiatan. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas (Anwar, 2014).

Tabel 2.9.1 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)

### 2.9.2 Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku di dalam suatu bisnis. Activity diagram dapat dilihat sebagai sebuah sophisticated data flow diagram (DFD) yang digunakan pada analisis structural. Akan tetapi, berbeda dengan DFD, activity diagram mempunyai notasi untuk memodelkan aktivitas yang berlangsung secara paralel, bersamaan, dan juga proses pengambilan keputusan yang kompleks. (Suryasari, 2014)

Tabel 2.9.2 Simbol *ActivityDiagram*

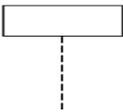
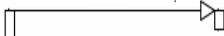
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final 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)

### 2.9.3 *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 2.9.3 Simbol *Sequence Diagram*

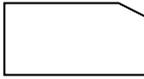
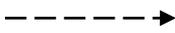
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi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

Sumber : (Gellysa Urva, 2015)

#### 2.9.4 *Class Diagram*

*Class diagram* adalah visualisasi kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Diagram ini memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dalam logical view dari suatu sistem. Kelas memiliki 3 area utama yaitu nama, atribut, dan operasi. Nama berfungsi untuk member identitas pada sebuah kelas, atribut fungsinya adalah untuk menunjukkan karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek. (Anwar, 2014)

Tabel 2.9.4 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

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisis Yang Sedang Berjalan

Seiring dengan semakin meningkatnya kebutuhan air minum pada saat ini, banyak pula depot-depot atau tempat pengisian ulang air minum yang bermunculan. Terkadang banyak sebagian dari masyarakat belum mengetahui apakah air tersebut layak untuk dikonsumsi atau tidak. Alasannya karena air yang di konsumsi sehari-hari tersebut tanpa melalui proses pemasakan atau perebusan terlebih dahulu. Depot-depot yang bermunculan saat ini pun banyak yang tidak mencantumkan surat dari dinas kesehatan setempat yang menyatakan bahwasanya air minum yang dijual layak untuk dikonsumsi. Biaya yang mahal dan urusan yang sangat sulit membuat para pemilik depot air minum tidak memperdulikan hal terpenting yang sebenarnya harus dimiliki. Salah satu hal yang dapat mengancam kesehatan melalui air minum adalah adanya bakteri *Escherichia coli* (E-Coli). Untuk mengetahui apakah air minum yang dikonsumsi mengandung bakteri E-Coli tidak mudah, karena ukurannya yang sangat kecil dan tidak kasat mata. Salah satu akibat yang dapat ditimbulkan dari bakteri E-Coli ini seperti sakit perut, muntaber, diare, tekanan darah tinggi, bahkan gangguan ginjal.

Berdasarkan hasil konsultasi yang dilakukan dengan BTKL (Badan Teknik Kesehatan Lingkungan) kota Medan terdapat beberapa ciri-ciri air minum yang tercemar bakteri *Escherichia coli* antara lain :

1. Berbau seperti selokan atau lumpur.
2. Terasa getir atau kelat dilidah.

3. Tidak jernih atau keruh.
4. Agak sedikit kekuningan.
5. Warna berubah dalam 2 hari.

Oleh karena itu berdasarkan analisis masalah yang terjadi, maka melalui sistem ini diharapkan menjadi pilihan alternatif dalam pengetahuan tentang bakteri *Escherichia coli* tersebut agar lebih efisien dan memudahkan masyarakat dalam mendiagnosa apakah air minum tersebut memiliki kualitas yang baik.

### 3.2 Penerapan Metode *Certainty Factor*

Setelah didapatkan ciri-ciri air minum yang tercemar bakteri *escherichia coli* dari BTKL (Badan Teknik Kesehatan Lingkungan), kemudian diberikan pertanyaan kepada pakar untuk mendapatkan nilai MB (*Measure of Belief*) dan MD (*Measure of increased Disbelief*) untuk setiap ciri-ciri air minum yang tercemar bakteri *escherichia coli*.

Berikut adalah tabel pertanyaan yang akan dijawab pengguna untuk mendapatkan hasil presentase dari depot air minum yang didiagnosa :

**Tabel 3.2 Pertanyaan Diagnosa**

Kode	Pertanyaan	Jawaban					
		TY	TT	KY	CY	Y	SY
C1	Apakah air minum isi ulang “Berbau seperti selokan atau lumpur” ?						
C2	Apakah air minum isi ulang “Terasa getir atau kelat di lidah” ?						
C3	Apakah air minum isi ulang “Tidak jernih atau keruh” ?						
C4	Apakah air minum isi ulang “Agak sedikit kekuningan” ?						
C5	Apakah air minum isi ulang “Warna berubah dalam 2 hari” ?						

Pada saat ingin menjawab pertanyaan akan diberikan 6 (enam) jawaban yang masing-masing memiliki bobot sebagai berikut :

1. TY (Tidak Yakin) : 0
2. TT (Tidak Tahu) : 0.2
3. KY (Kurang Yakin) : 0.4
4. CY (Cukup Yakin) : 0.6
5. Y (Yakin) : 0.8
6. SY (Sangat Yakin) : 1.0

Setelah semua pertanyaan dijawab, kemudian didapatkanlah nilai MB (*Measure of Belief*) dan MD (*Measure of increased Disbelief*) untuk setiap ciri-ciri dari air minum isi ulang yang tercemar bakteri *escherichia coli* seperti pada tabel 7 berikut ini

**Tabel 3.2 Nilai MB dan MD**

No	Kode	Ciri-Ciri Air	Nilai MB	Nilai MD
1	C1	Berbau seperti selokan atau lumpur	0.8	0.2
2	C2	Terasa getir atau kelat di lidah	0.8	0.2
3	C3	Tidak jernih atau keruh	0.8	0.2
4	C4	Agak sedikit kekuningan	0.6	0.4
5	C5	Warna berubah dalam 2 hari	0.6	0.4

Berdasarkan informasi ciri-ciri air serta nilai MB (*Measure of Belief*) dan MD (*Measure of increased Disbelief*) yang telah diperoleh maka pada kasus ini disusun sebuah kaidah produksi atau *rule base* yang berkaitan dengan bakteri *Escherichia coli*, kaidah tersebut adalah sebagai berikut :

IF berbau seperti selokan atau lumpur

AND terasa getir atau kelat di lidah

AND tidak jernih atau keruh

AND agak sedikit kekuningan

AND warna berubah dalam 2 hari

THEN mengandung bakteri *escherichia coli*.

Langkah untuk menentukan nilai CF untuk masing-masing ciri-ciri air adalah sebagai berikut :

CF (Berbau seperti selokan atau lumpur)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.8 - 0.2 = 0.6 \end{aligned}$$

CF (Terasa getir atau kelat di lidah)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.8 - 0.2 = 0.6 \end{aligned}$$

CF (Tidak jernih atau keruh)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.8 - 0.2 = 0.6 \end{aligned}$$

CF (Agak sedikit kekuningan)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.6 - 0.4 = 0.2 \end{aligned}$$

CF (Warna berubah dalam 2 hari)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.6 - 0.4 = 0.2 \end{aligned}$$

Kaidah atau *rule-rule* yang baru tersebut kemudian dihitung nilai CFnya dengan mengalikan  $CF(user)$  dengan  $CF(pakar)$  menjadi :

$$CF(H,E) = CF(Pakar)*CF(User)$$

$$CF1 = 0.6*1.0 = 0.6$$

$$CF2 = 0.6*0.8 = 0.48$$

$$CF3 = 0.6*0.8 = 0.48$$

$$CF4 = 0.2*0.6 = 0.12$$

$$CF5 = 0.2*0 = 0$$

Mengkombinasikan nilai CF dari masing-masing nilai yaitu kombinasikan CF1 dengan CF2 dengan rumus berikut :

$$\begin{aligned} CF_{combine}(CF_1, CF_2) &= 0.6 + 0.48*(1 - 0.6) \\ &= 0.6 + 0.192 \end{aligned}$$

$$CF_{old} = 0.792$$

$$\begin{aligned} CF_{combine}(CF_{old}, CF_3) &= 0.792 + 0.48*(1 - 0.792) \\ &= 0.792 + 0.09984 \end{aligned}$$

$$CF_{old2} = 0.89184$$

$$\begin{aligned} CF_{combine}(CF_{old2}, CF_4) &= 0.89184 + 0.12*(1 - 0.89184) \\ &= 0.89184 + 0.0129792 \end{aligned}$$

$$CF_{old3} = 0.9048192$$

$$\begin{aligned} CF_{combine}(CF_{old3}, CF_5) &= 0.9048192 + 0*(1 - 0.9437568) \\ &= 0.9048192 + 0 \end{aligned}$$

$$CF_{old4} = 0.9048192$$

Dari hasil perhitungan diatas, maka nilai CFnya adalah 0.9048192. maka persentase keyakinan =  $CF_{old4} * 100\% = 0.9048192 * 100\%$

$$= 90.48192\%$$
$$= 90.48\%.$$

Dengan demikian dapat dikatakan bahwa perhitungan *certainty factor* pada bakteri *escherichia coli* pada air minum isi ulang memiliki keyakinan persentase tingkat keyakinan sebesar 90.48%.

### 3.3 Analisis Kebutuhan

#### 1. Analisis Kebutuhan Fungsional

Terdapat dua alat penelitian dalam penyelesaian aplikasi mendiagnosa bakteri *escherichia coli (e-coli)* pada air minum isi ulang ini, yaitu :

##### a. *Smartphone Android*

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

- 1) *CPU* : *Qualcomm MSM8909 1,6 GHz, GPU*
- 2) *Memory Internal* : *2 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 Multi Touch Screen*
- 6) *Ukuran Layar* : *7200 x 1280 pixel*

##### b. *Hardware Komputer*

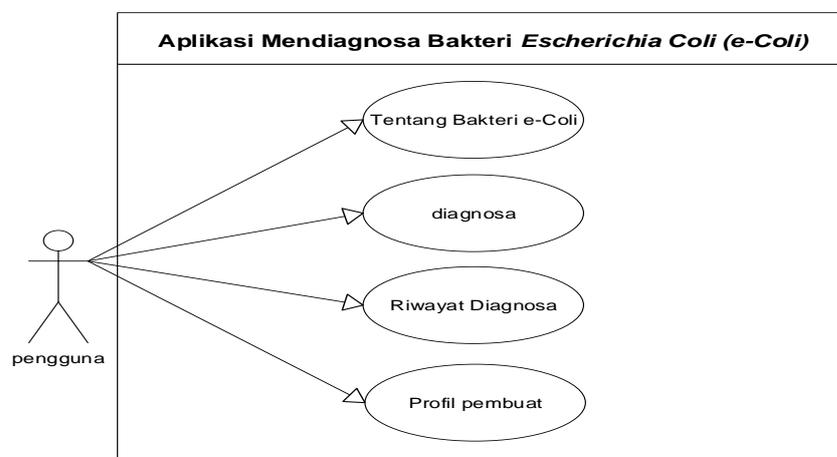
*Hardware* yang digunakan dalam perancangan aplikasi mendiagnosa bakteri *escherichia coli (e-coli)* pada air minum isi ulang memiliki spesifikasi sebagai berikut :

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

### 3.4 Use case

#### 3.4.1 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 mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang dapat dilihat dibawah ini :



**Gambar 3.4.1 Use Case Diagram Aplikasi Mendiagnosa Bakteri E-Coli**

### 3.4.2 Definisi Aktor

Berikut adalah deskripsi pendefinisian aktor pada aplikasi mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang:

**Tabel 3.4.2 Definisi Aktor**

Aktor	Deskripsi
Pengguna	Orang yang menggunakan aplikasi mendiagnosa bakteri <i>escherichia coli</i> (e-coli) pada air minum isi ulang.

### 3.4.3. Definisi Use case

Berikut adalah deskripsi pendefinisian *Use case* pada Aplikasi mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang:

**Tabel 3.4.3 Definisi Use case**

No	Use case	Deskripsi
1	Diagnosa	Merupakan menu yang berisi proses deteksi kualitas air minum dengan kriteria yang dialami.
2	Riwayat diagnosa	Merupakan menu untuk melihat data riwayat diagnosa pada depot air minum.
3	Tentang Bakteri E-Coli	Merupakan menu yang berisi tentang informasi mengenai bakteri e-coli ini
4	Profil pembuat	Merupakan menu yang berisi tentang informasi mengenai si pembuat aplikasi

### 3.4.4 Skenario Use case

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

### 3.4.4.1 Skenario *Use case* Diagnosa

Nama *Use case* : Diagnosa

Skenario :

**Tabel 3.4.4.1 Skenario *Use case* Diagnosa**

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu Diagnosa	
	2. Menampilkan form Diagnosayang berisi pertanyaan seputar kualias air yang didapat dari depot air minum.

### 3.4.4.2 Skenario *Use case* Riwayat Diagnosa

Nama *Use case* : Riwayat Diagnosa

Skenario :

**Tabel 3.4.4.2 Skenario *Use case*Riwayat Diagnosa**

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu Diagnosa	
	2. Menampilkan riwayat diagnosa tentang depot air minum yang telah didiagnosa.

### 3.4.4.3 Skenario *Use case* Tentang Bakteri E-Coli

Nama *Use case* : Tentang Bakteri E-Coli

Skenario :

**Tabel 3.4.4.3 Skenario *Use case* Tentang Bakteri E-Coli**

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu tentang bakteri e-coli	
	2. Menampilkan menu tentang bakteri e-coli untuk informasi tentang bahaya bakteri tersebut.

**3.4.4.4 Skenario *Use case* Profil Pembuat**

Nama *Use case* : Profil Pembuat

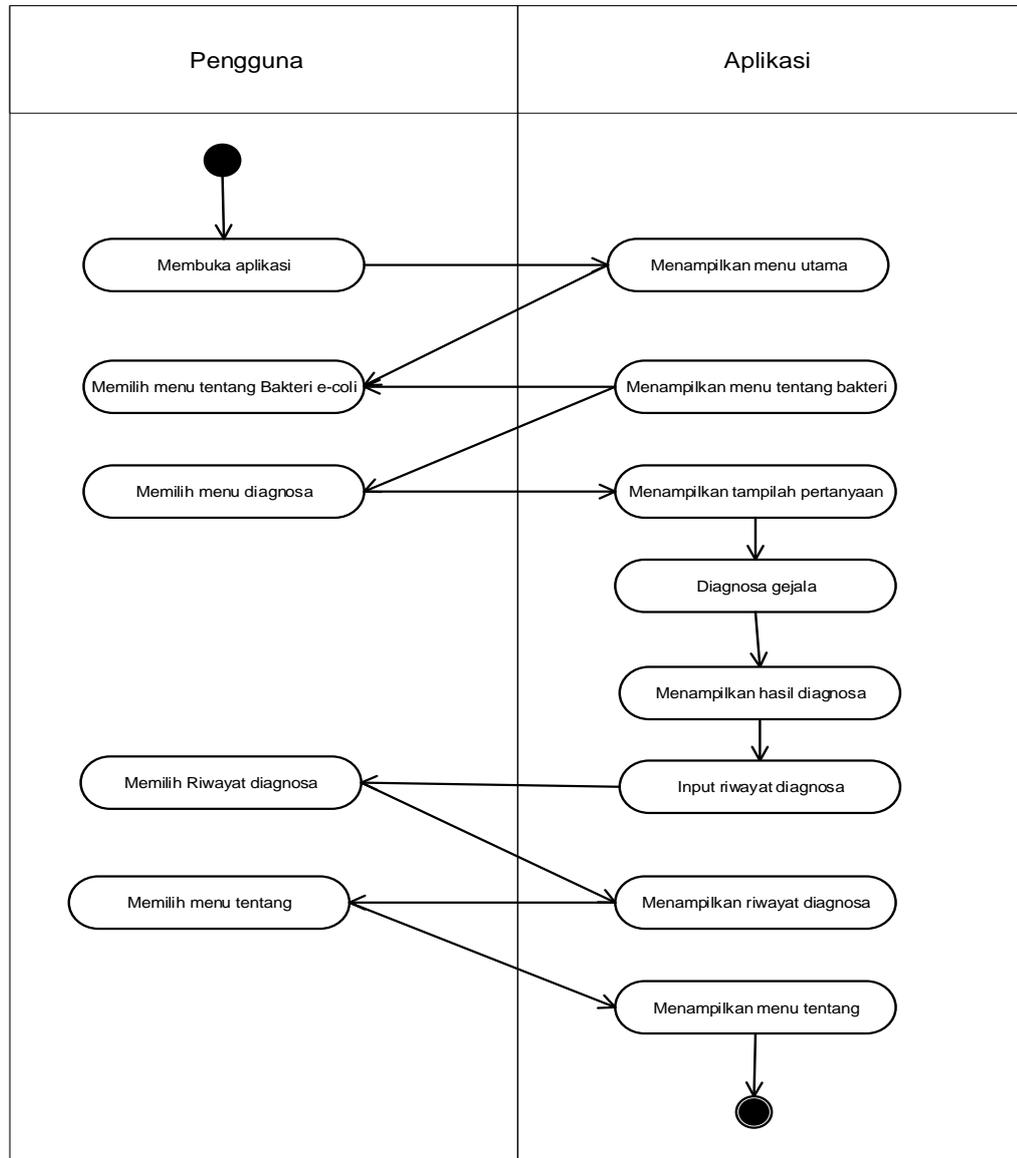
Skenario :

**Tabel 3.4.4.4 Skenario *Use case* Profil Pembuat**

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu profil pembuat	
	2. Menampilkan informasi tentang pembuat aplikasi mendiagnosa bakteri <i>escherichia coli</i> (e-coli) pada air minum isi ulang ini.

**3.5 Activity Diagram Aplikasi Mendiagnosa Bakteri E-Coli**

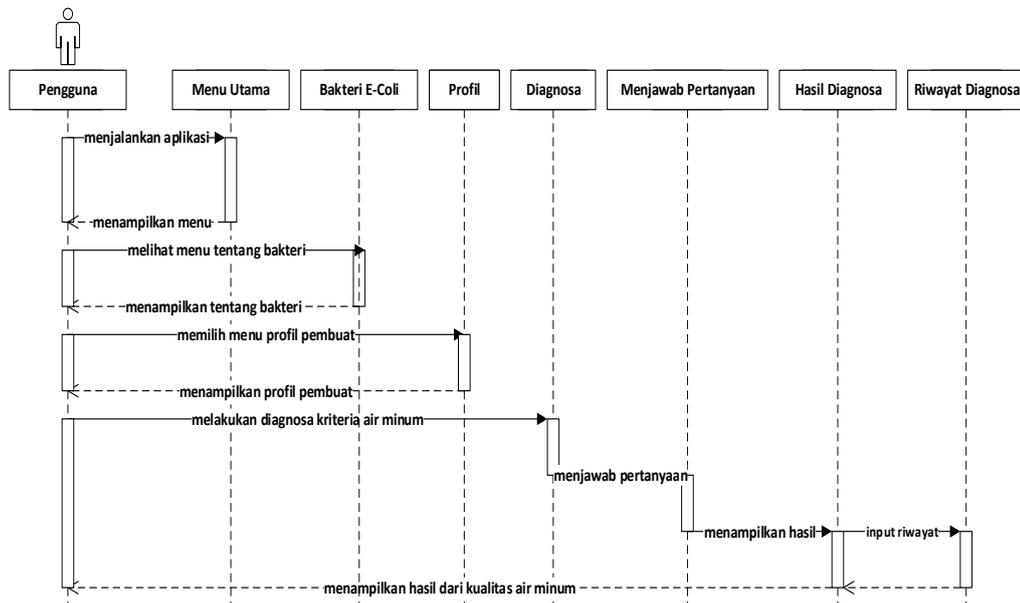
Berikut adalah *Activity diagram* aplikasi sistem pakar mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang:



**Gambar 3.5 Activity Diagram**

### 3.6 Diagram Sequence Aplikasi Mendiagnosa Bakteri E-Coli

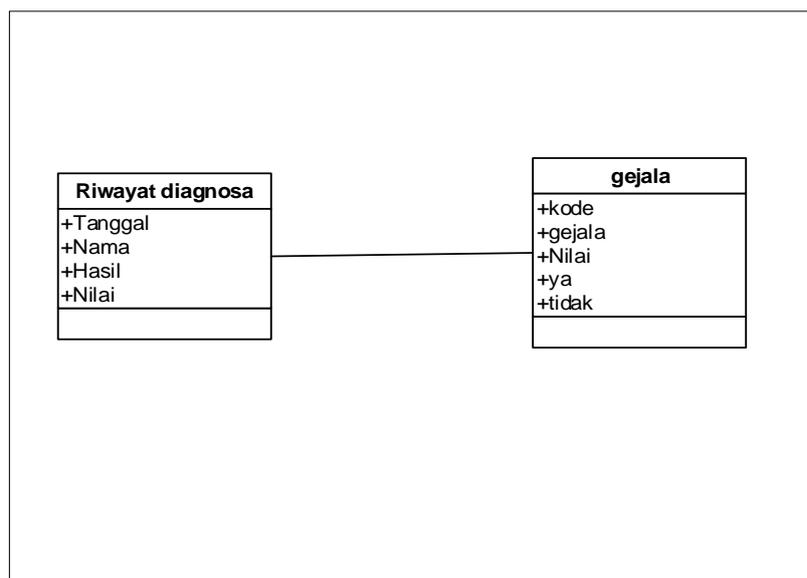
Berikut adalah *diagram sequence* aplikasi mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang:



**Gambar 3.6** *Sequence Diagram* Aplikasi Mendiagnosa Bakteri E-Coli

### 3.7 *Class Diagram* Aplikasi Mendiagnosa Bakteri E-Coli

Berikut adalah *classdiagram* aplikasi mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang:



**Gambar 3.7.** *Class Diagram* Aplikasi Mendiagnosa Bakteri E-Coli

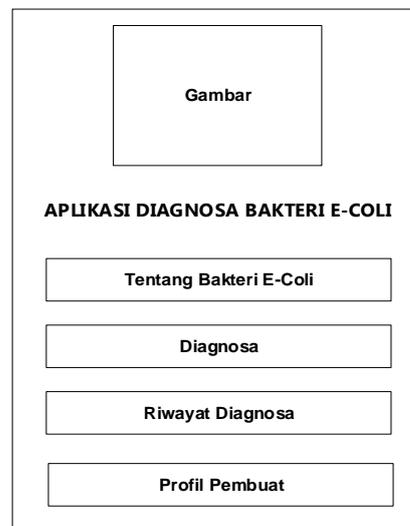
## **3.8 Perancangan Aplikasi**

### **3.8.1 Perancangan Antarmuka (*User Interface*)**

Perancangan Antarmuka 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 mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulang 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 :

#### **3.8.1.1 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 menu utama, yang dapat digunakan oleh pengguna untuk menuju ke tampilan yang diinginkannya dengan memilih menu yang ada pada tampilan. Menu utama memiliki logo, judul dan 3 tombol.



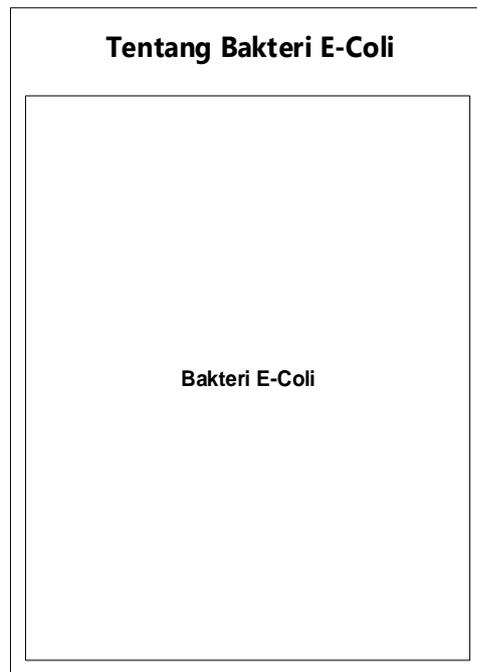
**Gambar 3.8.1.1 Rancangan Tampilan Menu Utama**

Berikut fungsi dari 4 tombol yang ada pada menu utama :

1. Tombol tentang bakteri e-coli berfungsi untuk mengetahui deskripsi tentang bahaya bakteri e-coli tersebut.
2. Tombol diagnosa berfungsi untuk menuju ke tampilan proses pertanyaan tentang kualitas air minum yang didapat dari depot air minum tersebut.
3. Tombol riwayat diagnosa berfungsi untuk melihat riwayat diagnosa terakhir oleh pengguna.
4. Tombol Profil pembuat berfungsi untuk menuju tampilan info dari data pembuat aplikasi.

### **3.8.1.2 Rancangan Tampilan Tentang Bakteri E-Coli**

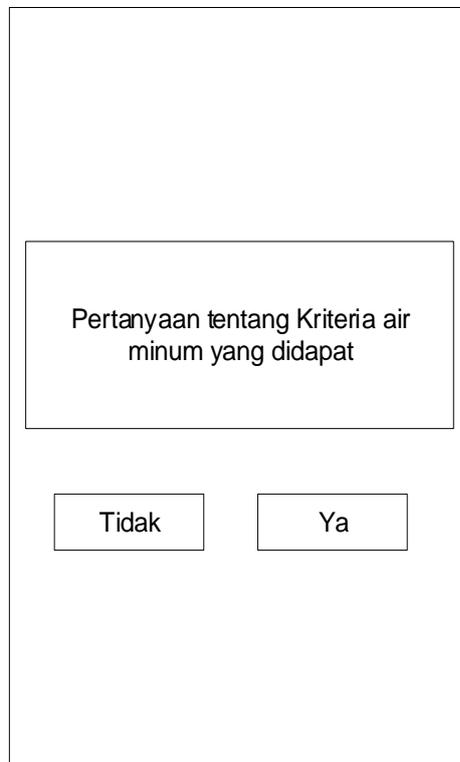
Rancangan ini adalah rancangan yang menampilkan tampilan tentang bahaya bakteri e-coli pada tubuh manusia.



**Gambar 3.8.1.2 Rancangan Tampilan Tentang Bakteri E-Coli**

### **3.8.1.3 Rancangan Tampilan Menu Utama Rancangan Tampilan Diagnosa**

Rancangan tampilandiagnosa adalah tampilan yang berisikan seputar pertanyaan tentang kualitas air minum yang didapat oleh masyarakat dari depot air minum tersebut. Tampilan diagnosa memiliki teks pertanyaan dan 2 tombol.



Pertanyaan tentang Kriteria air minum yang didapat

Tidak Ya

**Gambar 3.8.1.3 Rancangan Tampilan Diagnosa**

#### **3.8.1.4 Rancangan Tampilan Riwayat Diagnosa**

Rancangan tampilan riwayatdiagnosa adalah tampilan yang berisikan riwayat diagnosa depot air minum yang telah didiagnosa untuk mendapatkan hasil. Tampilan riwayat diagnosa dapat dilihat pada gambar dibawah ini.

Nama Depot	Hasil	Nilai

**Gambar 3.8.1.4 Rancangan Tampilan Riwayat Diagnosa**

#### **3.8.1.5 Rancangan Tampilan Menu Profil Pembuat**

Rancangan ini adalah rancangan yang menampilkan informasi dari si pembuat aplikasi diagnosa bakteri e-coli.

The image shows a wireframe for a user profile menu. At the top is a large square placeholder labeled "Gambar". Below it is the title "APLIKASI DIAGNOSA BAKTERI E-COLI". The user information is listed as follows:

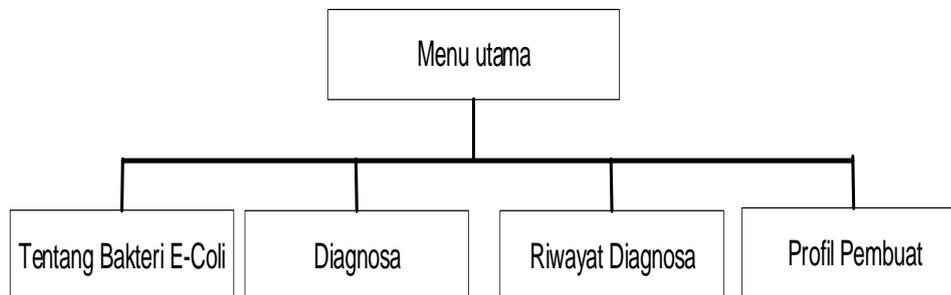
NAMA	:XXXXXXXXXXXX
N.P.M	:XXXXXXXXXXXX
PROGRAM STUDI	: SISTEM KOMPUTER

At the bottom, there are two side-by-side square placeholders, each labeled "Gambar".

**Gambar 3.8.1.5 Rancangan Tampilan Menu Profil**

### **3.8.1.6 Perancangan Arsitektur Navigasi**

Dari aplikasi mendiagnosa bakteri *escherichia coli* (e-coli) pada air minum isi ulangini, tampilan awalnya adalah tampilan *Menu Utamayang* didalamnya terdapat menu lain dan keseluruhan dari tampilan yang ada pada aplikasi ini, dapat dilihat pada gambar dibawah ini :



**Gambar 3.8.1.6 Struktur Arsitektur *Navigasi***

### **3.9 Perancangan *Database***

#### **3.8.1 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. Tabel-tabel yang ada di bawah ini tersimpan dalam suatu database yang bernama **db\_ecoli**.

### 3.8.2 Tabel Kriteria

**Tabel 3.8.2 Kriteria**

<b>Nama field</b>	<b>Type data</b>	<b>Size</b>	<b>Keterangan</b>
Kode	Varchar	10	Primary Key
Kriteria	Varchar	255	-
Nilai	Varchar	10	-
Ya	Varchar	100	-
Tidak	Varchar	20	-

### 3.8.3 Tabel Riwayat

**Tabel1 3.8.3 Riwayat**

<b>Nama_field</b>	<b>Type data</b>	<b>Size</b>	<b>Keterangan</b>
Kode	Varchar	10	Primary key
Nama Depot	Varchar	100	-
Hasil	Varchar	255	-
Nilai	Varchar	10	-

**BAB IV**  
**IMPLEMENTASI SISTEM**

**4.1 Penerapan Metode *Certainty Factor***

Berikut adalah tabel pertanyaan yang akan dijawab pengguna untuk mendapatkan hasil presentase dari depot air minum yang didiagnosa :

**Tabel 4.1 Pertanyaan Diagnosa**

Kode	Pertanyaan	Jawaban					
C1	Apakah air minum isi ulang “Berbau seperti selokan atau lumpur” ?	TY	TT	KY	CY	Y	SY
C2	Apakah air minum isi ulang “Terasa getir atau kelat di lidah” ?	TY	TT	KY	CY	Y	SY
C3	Apakah air minum isi ulang “Tidak jernih atau keruh” ?	TY	TT	KY	CY	Y	SY
C4	Apakah air minum isi ulang “Agak sedikit kekuningan” ?	TY	TT	KY	CY	Y	SY
C5	Apakah air minum isi ulang “Warna berubah dalam 2 hari” ?	TY	TT	KY	CY	Y	SY

Pada saat ingin menjawab pertanyaan akan diberikan 6 (enam) jawaban yang masing-masing memiliki bobot sebagai berikut :

- a. TY (Tidak Yakin) : 0
- b. TT (Tidak Tahu) : 0.2
- c. KY (Kurang Yakin) : 0.4

- d. CY (Cukup Yakin) : 0.6  
 e. Y (Yakin) : 0.8  
 f. SY (Sangat Yakin) : 1.0

Setelah semua pertanyaan dijawab, kemudian didapatkanlah nilai MB (*Measure of Belief*) dan MD (*Measure of increased Disbelief*) untuk setiap ciri-ciri dari air minum isi ulang yang tercemar bakteri *escherichia coli* seperti pada tabel 3 berikut ini

**Tabel 4.1 Nilai MB Dan MD**

No	Kode	Ciri-Ciri Air	Nilai MB	Nilai MD
1	C1	Berbau seperti selokan atau lumpur	0.8	0.2
2	C2	Terasa getir atau kelat di lidah	0.8	0.2
3	C3	Tidak jernih atau keruh	0.8	0.2
4	C4	Agak sedikit kekuningan	0.6	0.4
5	C5	Warna berubah dalam 2 hari	0.6	0.4

Berdasarkan informasi ciri-ciri air serta nilai MB (*Measure of Belief*) dan MD (*Measure of increased Disbelief*) yang telah diperoleh maka pada kasus ini disusun sebuah kaidah produksi atau *rule base* yang berkaitan dengan bakteri *Escherichia coli*, kaidah tersebut adalah sebagai berikut :

IF berbau seperti selokan atau lumpur

AND terasa getir atau kelat di lidah

AND tidak jernih atau keruh

AND agak sedikit kekuningan

AND warna berubah dalam 2 hari

THEN mengandung bakteri *escherichia coli*.

Langkah untuk menentukan nilai CF untuk masing-masing ciri-ciri air adalah sebagai berikut :

CF (Berbau seperti selokan atau lumpur)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.8 - 0.2 = 0.6 \end{aligned}$$

CF (Terasa getir atau kelat di lidah)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.8 - 0.2 = 0.6 \end{aligned}$$

CF (Tidak jernih atau keruh)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.8 - 0.2 = 0.6 \end{aligned}$$

CF (Agak sedikit kekuningan)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.6 - 0.4 = 0.2 \end{aligned}$$

CF (Warna berubah dalam 2 hari)

$$\begin{aligned} \text{CF(H,E)} &= \text{MB(H,E)} - \text{MD(H,E)} \\ &= 0.6 - 0.4 = 0.2 \end{aligned}$$

Kaidah atau *rule-rule* yang baru tersebut kemudian dihitung nilai CFnya dengan mengalikan  $\text{CF}(\text{user})$  dengan  $\text{CF}(\text{pakar})$  menjadi :

$$\begin{aligned} \text{CF(H,E)} &= \text{CF}(\text{Pakar}) * \text{CF}(\text{User}) \\ \text{CF1} &= 0.6 * 1.0 = 0.6 \end{aligned}$$

$$CF2 = 0.6 * 0.8 = 0.48$$

$$CF3 = 0.6 * 0.8 = 0.48$$

$$CF4 = 0.2 * 0.6 = 0.12$$

$$CF5 = 0.2 * 0 = 0$$

Mengkombinasikan nilai CF dari masing-masing nilai yaitu kombinasikan

CF1 dengan CF2 dengan rumus berikut :

$$CF_{combine}(CF1, CF2) = 0.6 + 0.48 * (1 - 0.6)$$

$$= 0.6 + 0.48 * 0.4$$

$$CF_{fold} = 0.792$$

$$CF_{combine}(CF_{fold}, CF3) = 0.792 + 0.48 * (1 - 0.792)$$

$$= 0.792 + 0.48 * 0.208$$

$$CF_{fold2} = 0.89184$$

$$CF_{combine}(CF_{fold2}, CF4) = 0.89184 + 0.12 * (1 - 0.89184)$$

$$= 0.89184 + 0.12 * 0.10816$$

$$CF_{fold3} = 0.9048192$$

$$CF_{combine}(CF_{fold3}, CF5) = 0.9048192 + 0 * (1 - 0.9437568)$$

$$= 0.9048192 + 0 * 0.562432$$

$$CF_{fold4} = 0.9048192$$

Dari hasil perhitungan diatas, maka nilai CFnya adalah 0.9048192. maka

$$\text{persentase keyakinan} = CF_{fold4} * 100\% = 0.9048192 * 100\%$$

$$= 90.48192\%$$

$$= 90.48\%.$$

Dengan demikian dapat dikatakan bahwa perhitungan *certainty factor* pada bakteri *escherichia coli* pada air minum isi ulang memiliki keyakinan persentase tingkat keyakinan sebesar 90.48%.

## **4.2 Implementasi Sistem yang Digunakan**

Tahapan implementasi yang dilakukan untuk menyelesaikan perancangan aplikasi diagnosa bakteri e-coli pada air minum isi ulang ini diperlukan informasi mengenai penyediaan perangkat keras (*Hardware*) dan perangkat lunak (*Software*).

Berikut disediakan perangkat keras dan perangkat lunak yang dibutuhkan.

### **4.2.1 Spesifikasi Perangkat Keras**

Aplikasi diagnosa bakteri e-coli pada air minum isi ulang 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 : 480 x 840 *pixel*

### **4.2.2 Spesifikasi Perangkat Lunak**

Aplikasi ini dijalankan pada perangkat lunak dengan spesifikasi sebagai berikut :

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

#### 4.2.2.1 Tampilan Aplikasi Diagnosa Bakteri E-Coli

Tampilan aplikasi diagnosa bakteri e-coli pada air minum isi ulang ini terdiri dari tampilan menu utama, tentang bakteri, diagnosa, riwayat diagnosa dan profil pembuat. Menu utama berisi menu-menu aplikasi yaitu menu utama, tentang bakteri, diagnosa, riwayat diagnosa dan profil pembuat.

Adapun tampilan menu-menu aplikasi diagnosa bakteri e-coli pada air minum isi ulang adalah sebagai berikut :

##### 4.2.2.1.1 Tampilan Menu Utama

Tampilan menu utama terdiri dari tentang bakteri, diagnosa, riwayat diagnosa dan profil pembuat.



**Gambar 4.2.2.1.1 Tampilan Menu Utama**

Berikut fungsi dari 4 tombol yang ada pada menu utama :

- a) Tombol tentang bakteri e-coli berfungsi untuk melihat tentang bakteri e-coli sebagai pengetahuan dasar bakteri yang berbahaya tersebut.
- b) Tombol diagnosa berfungsi untuk melakukan diagnosa bakteri e-coli pada air minum isi ulang pada depot air minum tertentu.
- c) Tombol riwayat diagnosa berfungsi untuk melihat data riwayat depot air minum yang telah didiagnosa sebelumnya.
- d) Tombol profil pembuat berfungsi untuk masuk ke menu tentang pembuat aplikasi diagnosa bakteri e-coli pada air minum isi ulang ini.

#### 4.2.2.1.2 Tampilan Tentang Bakteri E-Coli

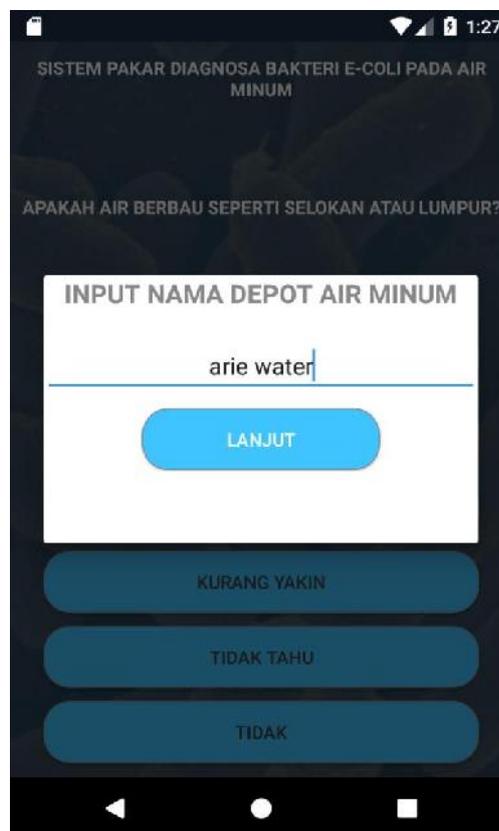
Tampilan tentang bakteri memiliki fungsi untuk melihat tentang pengertian dasar tentang bahaya bakteri e-coli pada tubuh manusia.



**Gambar 4.2.2.1.2 Tampilan Tentang Bakteri E-Coli**

#### 4.2.2.1.3 Tampilan Diagnosa

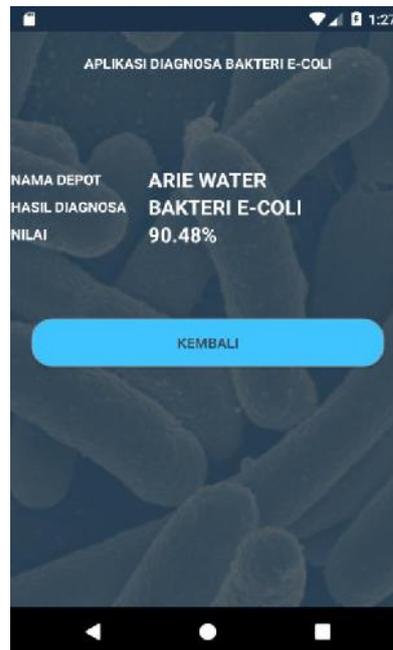
Tampilan diagnosa memiliki fungsi untuk diagnosa air minum yang didapat dari depot air minum tertentu untuk dapat menentukan nilai bakteri yang terdapat pada air minum tersebut. Dalam menu ini pengguna hanya tinggal menjawab pertanyaan yang diberikan aplikasi untuk dapat mendapatkan hasil persentase yang diberikan oleh aplikasi.



Gambar 4.2.2.1.3 Tampilan Diagnosa

#### 4.2.2.1.4 Tampilan Hasil Diagnosa

Tampilan hasil diagnosa memiliki fungsi sebagai memberikan hasil diagnosa air minum yang di indikasi mengandung bakteri e-coli pada air minum tersebut. Dalam menu ini pengguna akan ditampilkan hasil diagnosa dari jawaban pertanyaan seputar air minum yang didapat pada depot tertentu.



**Gambar 4.2.2.1.4 Tampilan Hasil Diagnosa**

#### 4.2.2.1.5 Tampilan Riwayat Diagnosa

Tampilan riwayat diagnosa memiliki fungsi untuk menampilkan riwayat diagnosa depot air minum yang telah didiagnosa sebelumnya.



**Gambar 4.2.2.1.5 Tampilan Riwayat Diagnosa**

#### 4.2.2.1.6 Tampilan Profil Pembuat

Tampilan profil pembuat berfungsi untuk melihat profil pembuat aplikasi diagnosa bakteri e-coli ini.

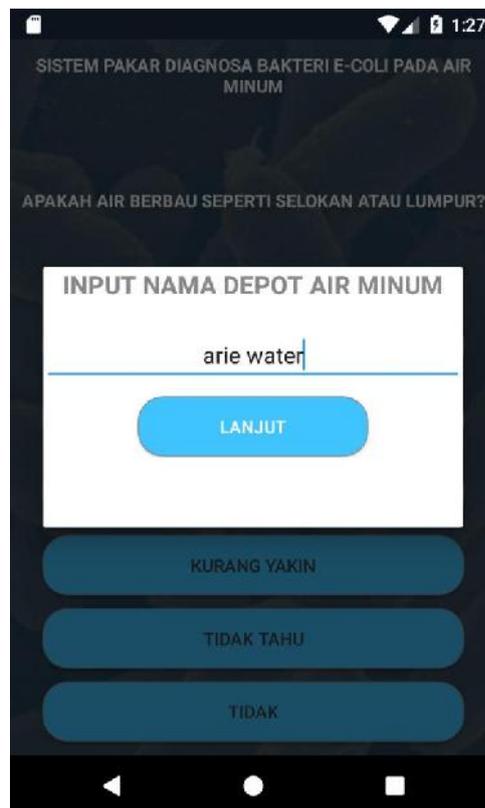


**Gambar 4.2.2.1.6 Tampilan Profil**

#### 4.2.2.2 Pengujian Aplikasi Diagnosa Bakteri E-Coli

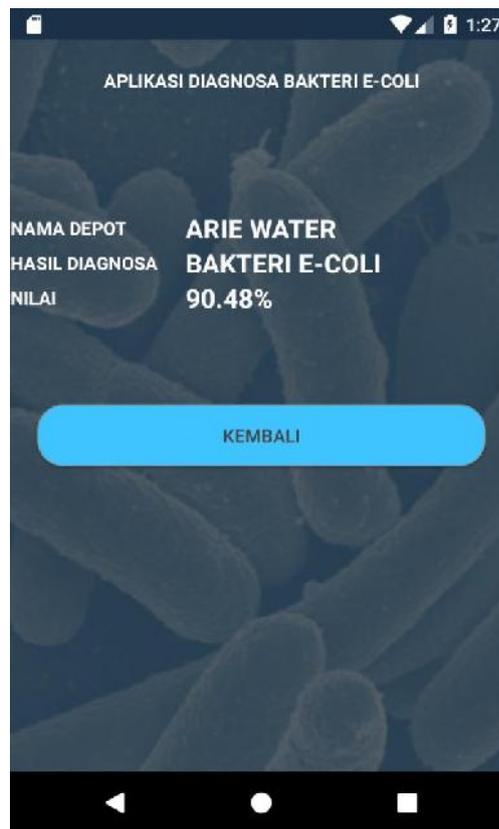
Pengujian aplikasi diagnosa bakteri e-coli pada air minum isi ulang digunakan untuk menguji sistem pada salah satu menu dimana data yang digunakan adalah proses menjawab pertanyaan seputar air minum yang didapat dari depot air minum tersebut. Cara menggunakan aplikasi diagnosa bakteri e-coli pada air minum isi ulang adalah sebagai berikut :

1. Langkah awalnya pengguna menjalankan aplikasi dan akan ditampilkan menu utama aplikasi.
2. Kemudian pengguna memilih menu diagnosa yang ada pada aplikasi.
3. Lalu akan tampil menu diagnosa air minum. Sebelum memulai diagnosa, pengguna diwajibkan untuk menginputkan nama depot air minum yang akan didiagnosa. Jika telah melakukan input nama depot air minumnya, dapat langsung melanjutkan dengan klik pada tombol lanjut yang telah tersedia pada aplikasi.



**Gambar 4.2.2.2 Diagnosa**

4. Setelah pengguna selesai dalam menjawab pertanyaan yang diberikan oleh aplikasi, maka akan ditampilkan tampilan baru yaitu tampilan hasil.



**Gambar 4.2.2.2 Hasil Diagnosa**

5. Dalam menu hasil diagnosa tersebut, pengguna akan ditampilkan hasil nilai bakteri e-coli yang terdapat pada air minum isi ulang pada depot air minum yang telah didiagnosa. Sehingga pengguna dapat lebih mengetahui apakah depot air minum tersebut airnya mengandung bakteri e-coli atau tidaknya dengan menggunakan aplikasi diagnosa bakteri e-coli ini dengan mudah.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pembuatan aplikasi diagnosa bakteri e-coli pada air minum isi ulang ini, maka didapat beberapa kesimpulan seperti berikut:

1. Penerapan metode *certainty factor* sangat sesuai digunakan pada sistem pakar mendiagnosa bakteri *escherichia coli* pada air minum isi ulang, sehingga dapat diketahui hasil tingkat kepastian diagnosa bakteri *escherichia coli* yang pada dasarnya pengguna aplikasi ini tidak mengetahui apakah air minum isi ulang yang akan dikonsumsi mengandung bakteri *escherichia coli* atau tidak.
2. Metode *certainty factor* dan pemrograman C# yang menghasilkan aplikasi berbasis android dapat membantu untuk mengetahui adanya bakteri *escherichia coli* pada air minum isi ulang menggunakan bantuan *smartphone android* serta sebagai sarana pencegahan agar air yang mengandung bakteri *escherichia coli* tersebut tidak dikonsumsi, supaya tidak menimbulkan penyakit.

#### **5.2 Saran**

Berikut adalah saran dari penulis agar aplikasi diagnosa bakteri e-coli pada air minum isi ulang ini dapat bermanfaat dan dikembangkan menjadi lebih baik lagi :

1. Bakteri yang disajikan dalam sistem pakar ini dibatasi hanya satu bakteri yaitu bakteri *escherichia coli*, perlu ditambahkan jenis bakteri lainnya pada air

minum isi ulang pada penelitian selanjutnya, sehingga sistem pakar ini dapat mendiagnosa lebih banyak bakteri pada air minum isi ulang.

2. Untuk pengembangan penelitian selanjutnya diharapkan sistem yang dirancang tidak hanya dapat digunakan pada *mobile*, tapi juga dapat digunakan diperangkat komputer *desktop* pada umumnya.

## DAFTAR PUSTAKA

- Alessandro, Sole Del. 2017. "Visual Studio 2017 Succinctly." *SynCFusion, Inc.*
- Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." Seminar Nasional Informatika (SNIf). Vol. 1. No. 1. 2017.
- Anwar, Sariyun Naja et al. 2014. "Desain Uml Aplikasi Navigasi Layanan Kesehatan Berbasis Android." (September).
- Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.
- Batubara, Supina, Sri Wahyuni, and Eko Hariyanto. "Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Dalam." Seminar Nasional Royal (SENAR). Vol. 1. No. 1. 2018.
- 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.
- FACHRI, Barany. Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. *Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika)*, 2018, 3: 98-102.
- Fahnun, Budi Utami, Rina Noviana, Lely Prananingrum, and Enlik Tjioe. 2013. "Informasi Kampus Berbasis Web Pada Android." *Seminar Nasional Teknologi Informasi dan Multimedia 2013*: 25–32.
- Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. *Int. J. Recent Trends Eng. Res*, 3(8), 58-64.
- Harto, Dodi. 2013. "Perancangan Sistem Pakar Untuk Mengidentifikasi Penyakit Pada Tanaman Semangka Dengan Menggunakan Metode Certainty Factor." *Pelita Informatika Budi Darma* IV: 22–27.

- Iqbal, Muhammad, Zarlis Muhammad, and T.H.F .Harumy. 2017. "Inovasi Aplikasi Check in Spot Nelayan Untuk Binaan Kecamatan Bagan Deli Medan." (November): 1–6.
- Khairul, K., IlhamiArsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). Implementasi Augmented Reality Sebagai Media Promosi Penjualan Rumah. In Seminar Nasional Royal (Senar) (Vol. 1, No. 1, pp. 429-434).
- Mairizki, Fitri. 2017. "Analisis Kualitas Air Minum Isi Ulang Di Sekitar Kampus Universitas Islam Riau." *jurnal Katalisator* 2(1): 9–19.
- Maulana, Mohammad Rochman Wahid. 2017. "Pengembangan Aplikasi Android Untuk Studi Bahasa Carakan Madura." *Journal Information Engineering and Educational Technology*
- Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." *Int. J. Recent Trends Eng. Res* 2.12 (2016): 140-151.
- Muharom, Arzan, and Rinda Cahyana. 2012. "Sistem Informasi Geografis Konsep-Konsep Dasar (Perspektif Geodesi & Informatika)." *Sekolah Tinggi Teknologi Garut* 6(1): 2–7.
- Nirmala. 2017. "Sistem Pakar Diagnosa Awal Penyakit Kulit Pada Sapi Bali Dengan Menggunakan Metode Forward Chaining Dan Certainty Factor." *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)* 3: 110–17.
- Noer, zeni muhammad, and Kafi Ngamali. 2017. "Aplikasi Perpustakaan Smk Siliwangi Ams Banjarsari Berbasis Android." 5(2): 160–65.
- Permana, A. I., and Z. Tulus. "Combination of One Time Pad Cryptography Algorithm with Generate Random Keys and Vigenere Cipher with EM2B KEY." (2020).
- Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." *Seminar Nasional Teknologi Informasi Dan Multimedia, ISSN.* 2015.
- Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. *Int. J. Secur. Its Appl*, 10(8), 173-180.
- Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." *INTECOMS: Journal of Information Technology and Computer Science* 1.1 (2018): 72-77.

- Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.
- Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." *Int. J. Eng. Trends Technol* 42.5 (2016): 218-222.
- Surahman, Surawijaya, and Eko Budi Setiawan. 2017. "Aplikasi Mobile Driver Online Berbasis Android Untuk Perusahaan Rental Kendaraan." *VIII(1)*: 35–42.
- Suryasari, Astrid Callista, and Juwita Sari. 2012. "Rancangan Aplikasi Customer Service Pada PT. Lancar Makmur Bersama." *Jurnal Sistem Informasi* 4(2): 468–76.
- Urva, Gellysa, and Helmi Fauzi Siregar. 2015. "Pemodelan UML E-Marketing Minyak Goreng." *Jurnal Teknologi dan Sistem Informasi* 1(9): 92–101.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu* 10.2 (2018): 1899-1902.
- Wamiliana, Deprianto, and Aristoteles. 2015. "Pengembangan Sistem Pakar Berbasis Web Mobile Untuk Mengidentifikasi Penyebab Kerusakan Telepon Seluler Dengan Menggunakan Metode Forward Dan Backward Chaining." *Jurnal Komutasi* 1(Sistem Pakar): 1–9.
- Widyaningsih, Wiwid et al. 2016. "Analisis Total Bakteri Coliform Di Perairan Muara Kali Wiso Jepara." 5: 157–64.
- Yulansari, Kiki, and 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.