

**IMPLEMENTASI IDENTIFIKASI PANDUAN DIET DENGAN METODE
FORWARD CHAINING BERBASIS ANDROID**

Oleh :

MEVICI DIANRESTI

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA KOMPUTER

pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG**

2017

Judul Skripsi : **IMPLEMENTASI IDENTIFIKASI
PANDUAN DIET DENGAN METODE
FORWARD CHAINING BERBASIS
ANDROID**

Nama Mahasiswa : **Mevici Dianresti**

Nomor Pokok Mahasiswa : **1317051040**

Jurusan : **Ilmu Komputer**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

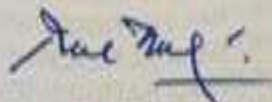


Aristoteles, S.Si., M.Si.
NIP 19810521 200604 1 002



dr. Tutik Ernawati, M.Gizi, Sp.GK.
NIP 19680606 200212 2 004

2. Ketua Jurusan Ilmu Komputer



Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji

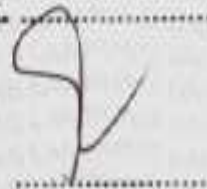
Ketua : **Aristoteles, S.Si., M.Si.**



Sekretaris : **dr. Tutik Ernawati, M.Gizi, Sp.GK.**



Penguji
Bukan Pembimbing : **Dr. Eng. Admi Syarif**



Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam


Prof. Warsito, S.Si., D.E.A., Ph.D.
NIP. 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi : **14 September 2017**

PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Implementasi Identifikasi Panduan Diet dengan Metode *Forward Chaining* Berbasis Android” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 14 September 2017



MEVICI DIANRESTI

NPM. 1317051040

RIWAYAT HIDUP



Penulis dilahirkan pada tanggal 14 Mei 1995 di Bandar Lampung, dengan Ibu bernama Efni Ekawati dan Ayah bernama Feri Fesdian, S.Pd.

Penulis menyelesaikan pendidikan formal pertama kali di TK Al-Azhar 2 Bandar Lampung tahun 2001, menyelesaikan Sekolah Dasar (SD) di SD Al-Azhar 1 Bandar Lampung tahun 2007, menyelesaikan

Sekolah Menengah Pertama (SMP) di SMP Negeri 29 Bandar Lampung tahun 2010, kemudian melanjutkan jenjang Sekolah Menengah Atas (SMA) di SMA Al-Azhar 3 Bandar Lampung dan lulus tahun 2013.

Pada tahun 2013, penulis terdaftar sebagai mahasiswi Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung (FMIPA) melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Pada bulan Januari – Februari 2016, penulis melakukan kerja praktik di Kantor Direksi PT. Perkebunan Nusantara 7 (PTPN 7) Lampung selama 40 hari. Kemudian pada bulan Juli 2016 penulis melakukan Kuliah Kerja Nyata (KKN) selama 40 hari di Desa Ngestirahayu Kecamatan Punggur Kabupaten Lampung Tengah.

Selama menjadi mahasiswa, penulis aktif dalam Organisasi AIESEC Unila dengan jabatan sebagai *Staff of Process and Quality Control Outgoing Exchange* pada tahun 2013-2014.

PERSEMBAHAN

*Puji dan syukur tak henti ku panjatkan kepada Allah SWT atas segala berkah-Nya
sehingga skripsi ini dapat terselesaikan.*

*Teruntuk yang terkasih Mama dan Papa kupersembahkan skripsi ini, hasil jerih
payah ku selama 4 tahun ini.*

Terima kasih untuk cinta kasih kalian untukku.

Terimakasih atas doa, dukungan, serta pengorbanan yang tiada henti.

*Teruntuk kakakku tersayang dan keluarga besar
yang telah memberikan doa dan apresiasi.*

Keluarga Ilmu Komputer 2013,

serta Almamater tercinta,

Universitas Lampung.

Motto

“Allah tidak membebani seseorang itu melainkan sesuai dengan kesanggupannya.”

(Q.S.Al-Baqarah: 286)

“Maka sesungguhnya bersama kesulitan itu ada kemudahan. Sesungguhnya bersama kesulitan itu ada kemudahan.”

(Q.S.Al-Insyirah: 5-6)

“Barangsiapa yang mempermudah kesulitan orang lain di dunia, maka Allah akan mempermudah urusannya di dunia dan akhirat.”

(HR. Muslim)

“Kadang-kadang Allah turunkan petir dan kilat, kita menangis mencari dimana hilangnya matahari, rupa-rupanya Allah hendak beri kita pelangi.”

(Anonim)

“No one is you, and that is your super power.”

(Anonim)

SANWACANA

Assalamualaikum wr., wb.

Alhamdulillah, segala puji bagi Allah SWT yang telah melimpahkan berkah, rahmat, hidayah, kesehatan, dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Implementasi Identifikasi Panduan Diet dengan Metode *Forward Chaining* Berbasis Android” dengan baik. Terima kasih penulis ucapkan kepada semua pihak yang telah membantu dan berperan besar dalam menyusun skripsi ini, antara lain:

1. Kedua orang tua tercinta, Mama, Efni Ekawati, dan Papa, Feri Fesdian S.Pd., yang tak henti-hentinya memberikan doa, kasih sayang, dukungan dan semangat yang tak terhingga serta memfasilitasi kebutuhan untuk menyelesaikan skripsi ini.
2. Bapak Aristoteles, S.Si., M.Si., sebagai pembimbing utama yang telah membimbing, memotivasi serta memberikan ide, kritik dan saran selama masa perkuliahan dan penyusunan skripsi sehingga penulis bisa sampai di tahap ini.
3. Ibu dr. Tutik Ernawati, M.Gizi, Sp.GK., pembimbing kedua yang telah membimbing dan memberikan bantuan, ide, kritik serta saran dalam penyusunan skripsi ini.
4. Bapak Dr. Eng. Admi Syarif, sebagai pembahas yang telah memberikan komentar dan masukan yang bermanfaat untuk perbaikan dalam penyusunan skripsi ini.

5. Bapak Prof. Warsito, S.Si., D.E.A., Ph.D., selaku Dekan FMIPA Universitas Lampung.
6. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Bapak Didik Kurniawan, S.Si., M.T., selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah banyak membantu penulis selama perkuliahan.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pengalaman hidup selama penulis menjadi mahasiswa.
9. Ibu Ade Nora dan Kak Irshan yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
10. Tiffany Dianresti, kakakku, dan M. Syifaul Qolbi, adikku tersayang, terima kasih atas doa, dukungan, dan semangat yang kalian berikan.
11. Jossi Chaerunisa, Frischandisany Faradina Putri, M. Olatta Fernanda, dan Layalia Jasmine Soraya, sepupu-sepupu tersayang, terima kasih atas doa dan semangat positif yang kalian berikan.
12. Fanny Prawaka, yang selalu memberikan dukungan dan doa serta tidak pernah jenuh menjadi tempatku berbagi masalah yang dihadapi selama perkuliahan dan penyusunan skripsi.
13. Engkong, Nenda, Ibu Ferra, Bapak Jhon, serta seluruh keluarga besar yang selalu memberikan dukungan dan doa.
14. Sahabat-sahabatku tercinta, Zakiah Nisrina, Mita Fuljana, Gita Ferdina Anggraini, Nisa Mashenta, Navia Yufitasari, Tika Oktavia, Ratu Mustika

Permata Lesi, Rizka Esa Basri, dan Rahmi Intan Amimy, terima kasih telah menjadi teman yang selalu memberikan hiburan dan semangat setiap saat.

15. Meita, Deka, Nisa, Kiki, Rendi, dan Nuril, terima kasih telah menjadi teman yang baik.
16. Sahabat seperjuangan Fani, yang tidak pernah jenuh menjadi tempat bertanya terutama di masa penyusunan skripsi ini.
17. Teman-teman Ilmu Komputer 2013 yang tidak bisa disebutkan satu per satu, terima kasih atas kebersamaannya selama ini.
18. Almamater tercinta, Universitas Lampung.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, akan tetapi sedikit harapan semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan terutama bagi teman-teman Ilmu Komputer.

Bandar lampung, 14 September 2017

Mevici Dianresti

DAFTAR ISI

	Halaman
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
II. TINJAUAN PUSTAKA	5
2.1 Kecerdasan Buatan	5
2.1.1 Bidang-Bidang Kecerdasan Buatan.....	6
2.2 Sistem Pakar	7
2.2.1 Ciri-Ciri Sistem Pakar	8
2.2.2 Keunggulan Sistem Pakar.....	8
2.2.3 Konsep Sistem Pakar	9
2.2.4 Arsitektur Sistem Pakar	11
2.2.5 Mekanisme Inferensi	13
2.2.5.1 Metode <i>Forward Chaining</i> (Runut Maju).....	14
2.2.5.2 Penanganan Data Tidak Lengkap.....	18
2.3 Probabilitas Klasik	19
2.4 Sistem Operasi Android	20
2.5 Panduan Diet dan Penyakit	21
2.5.1 Panduan Diet pada Ilmu Gizi	21
2.5.2 Penyakit	22
2.5.2.1 Tekanan Darah Tinggi.....	22
2.5.2.2 Osteo Arthritis	22
2.5.2.3 Gout	22

2.5.2.4	Gastritis	23
2.5.2.5	Infeksi Hati	23
2.5.2.6	Batu Empedu	23
2.5.2.7	Penurunan Fungsi Ginjal	23
2.5.2.8	Kencing Manis	24
2.6	Metode Pengembangan Sistem	24
2.6.1	Metode <i>Waterfall</i>	24
2.6.2	UML (<i>Unified Modelling Language</i>)	26
2.7	Teknik Pengujian Perangkat Lunak	27
2.7.1	<i>Equivalence Prartitioning</i> (EP)	28
2.7.2	Skala <i>Likert</i>	28
III.	METODE PENELITIAN	30
3.1	Waktu dan Tempat Penelitian	30
3.3	Tahapan Penelitian	31
3.3.1	Tahap Perumusan Masalah	31
3.3.2	Tahap Pengumpulan Data	32
3.3.3	Tahap Pengembangan Sistem	32
3.3.3.1	<i>Software Requirements Analysis</i>	32
3.3.3.2	<i>Design</i>	33
3.3.3.3	<i>Coding</i>	44
3.3.3.4	<i>Testing</i>	44
3.3.4	Analisis Hasil Pengujian	50
IV.	HASIL DAN PEMBAHASAN	51
4.1	Analisis Hasil Penelitian	51
4.2	Representasi Pengetahuan	51
4.3	Implementasi Sistem	52
4.4	Tampilan Aplikasi Sistem Pakar	53
4.4.1	Tampilan Halaman <i>Splash Screen</i>	53
4.4.2	Tampilan Menu Utama	53
4.4.3	Tampilan Menu Konsultasi	54
4.4.3.1	Analisis Perhitungan Presentase	56
4.4.4	Tampilan Menu Data Panduan Diet	58

4.4.5	Tampilan Menu Bantuan	58
4.4.6	Tampilan Menu Tentang	59
4.5	Pengujian Sistem	60
4.5.1	Pengujian Internal.....	60
4.5.1.1	Pengujian Fungsional	60
4.5.1.2	Pengujian Kepakaran Sistem.....	66
4.5.2	Pengujian Eksternal	67
4.5.2.1	Analisis Hasil Kuisioner.....	72
V.	KESIMPULAN DAN SARAN	78
5.1	Kesimpulan.....	78
5.2	Saran.....	79

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel	Halaman
1 Contoh Ilustrasi Aturan dan Konklusi.....	17
2 Tingkat Preferensi Jawaban Skala <i>Likert</i>	29
3 Daftar Pengujian Versi Android.....	45
4 Daftar Pengujian Resolusi Layar dan Densitas Layar	46
5 Daftar Pengujian <i>User Interface</i>	47
6 Daftar Pengujian Fungsi Menu Aplikasi	48
7 Daftar Pertanyaan Pengujian Non-Fungsional	49
8 Hasil Pengujian Versi Android.....	61
9 Hasil Pengujian Resolusi Layar dan Densitas Layar.....	62
10 Hasil Pengujian <i>User Interface</i>	63
11 Hasil Pengujian Fungsi Menu Aplikasi	64
12 Hasil Pengujian Kepakaran Sistem	66
13 Hasil Penilaian Kategori Responden I	69
14 Hasil Penilaian Kategori Responden II	70
15 Hasil Penilaian Kategori Responden III.....	71
16 Kriteria <i>Index</i> Penilaian Hasil Kuisioner	72

DAFTAR GAMBAR

Gambar	Halaman
1 Komponen Umum Sistem Pakar	10
2 Arsitektur Sistem Pakar	11
3 <i>Forward Chaining</i>	14
4 <i>Flowchart</i> Proses Inferensi	15
5 Penalaran Pelacakan ke Depan	15
6 <i>Graph</i> Pengetahuan	17
7 Metode <i>Waterfall</i>	24
8 Diagram Alir Penelitian	31
9 <i>Use-case Diagram</i>	32
10 <i>Sequence Diagram</i> Konsultasi	33
11 <i>Sequence Diagram</i> Data Panduan Diet	34
12 <i>Sequence Diagram</i> Bantuan	35
13 <i>Sequence Diagram</i> Tentang	35
14 <i>Activity Diagram</i> Konsultasi	36
15 <i>Activity Diagram</i> Data Panduan Diet	37
16 <i>Activity Diagram</i> Bantuan	38
17 <i>Activity Diagram</i> Tentang	38
18 <i>Class Diagram</i>	39
19 Rancangan <i>Layout Splash Screen</i>	40
20 Rancangan <i>Layout Beranda</i>	41
21 Rancangan <i>Layout Konsultasi</i>	41
22 Rancangan <i>Layout Pertanyaan Gejala</i>	42
23 Rancangan <i>Layout Hasil Diagnosis</i>	42
24 Rancangan <i>Layout Data Panduan Diet</i>	43
25 Rancangan <i>Layout Bantuan</i>	43
26 Rancangan <i>Layout Tentang</i>	44
27 <i>Splash Screen</i>	53
28 <i>Screen Informasi</i>	53

29 Tampilan Halaman Utama	54
30 Tampilan Halaman “Mulai”	55
31 Tampilan Menu Konsultasi	55
32 Tampilan Halaman Hasil Konsultasi	56
33 Tampilan Detail Panduan Diet Hasil Konsultasi	56
34 Tampilan Menu Data Panduan Diet	58
35 Tampilan Menu Bantuan	59
36 Tampilan Menu Tentang	60
37 Grafik Jawaban untuk Pertanyaan 1	73
38 Grafik Jawaban untuk Pertanyaan 2	73
39 Grafik Jawaban untuk Pertanyaan 3	74
40 Grafik Jawaban untuk Pertanyaan 4	75
41 Grafik Jawaban untuk Pertanyaan 5	75
42 Grafik Jawaban untuk Pertanyaan 6	76
43 Grafik Jawaban untuk Pertanyaan 7	77

DAFTAR LAMPIRAN

Lampiran	Halaman
1 Data Panduan Diet	85
2 Data Gejala	88
3 Tabel Keputusan	89
4 Pohon Keputusan	91
5 Basis Aturan	92

I. PENDAHULUAN

1.1 Latar Belakang

Ilmu gizi atau ilmu nutrisi adalah ilmu yang mempelajari proses dalam tubuh makhluk hidup untuk memanfaatkan makanan guna pembentukan energi, tumbuh kembang, dan pemeliharaan tubuh. Di dalam ilmu gizi dikenal istilah diet. Istilah diet sering disalahartikan oleh orang awam sebagai pembatasan makanan pada penderita suatu penyakit. Definisi diet yang sebenarnya adalah pengaturan jumlah dan jenis makanan yang dimakan setiap hari agar seseorang lebih terjaga kesehatannya. Panduan diet dimanfaatkan untuk menjaga kesehatan dan status gizi seseorang [1]. Berbagai keadaan penyakit harus menjadi perhatian dalam keputusan memberikan nutrisi seseorang [2].

Berbagai penyakit dapat diprediksi berdasarkan keluhan yang dialami seseorang. Meskipun tidak efektif, namun berbagai keluhan tersebut dapat mengarahkan ke kemungkinan terjadinya suatu penyakit. Hal ini dapat membantu memberikan arahan ke seseorang untuk memeriksakan diri ke dokter yang sesuai dengan kondisinya sehingga penyakitnya tidak cepat memburuk. Ketika seseorang mengkonsultasikan status gizinya, maka dapat diberikan nutrisi dengan panduan diet yang tepat sehingga lebih terjaga status gizinya serta dapat menekan progresivitas penyakit yang diderita [3].

Masyarakat pada umumnya masih bergantung kepada tenaga medis untuk memperoleh informasi mengenai suatu penyakit, mulai dari gejala hingga solusinya. Namun, terbatasnya waktu dan jumlah tenaga medis terkadang menjadi kendala bagi masyarakat untuk berkonsultasi dengan tenaga medis [4].

Penelitian sebelumnya yang mengkaji pengembangan sistem pakar berjudul Sistem Pakar Pengobatan Herbal, melakukan rancang bangun sistem pakar untuk membantu konsultasi dalam diagnosa penyakit dalam manusia dengan solusi terapi obat herbal berbasis desktop dengan Metode *Forward Chaining* [5].

Penelitian lainnya dengan judul Sistem Pakar Pada Konsultasi Jenis Senam, menggunakan metode *forward chaining* untuk melakukan konsultasi jenis senam yang tepat sesuai dengan kondisi dan kemampuan tubuh masing-masing orang [6]. Penelitian dengan judul Sistem Pakar Diagnosa Penyakit Pada Ikan Budidaya

Air Tawar, memanfaatkan sistem pakar untuk mengidentifikasi penyakit pada budidaya ikan tawar dengan metode *forward chaining* [7]. Penelitian tentang sistem pakar lainnya dengan judul Sistem Pakar Untuk Mendiagnosa Penyakit Dalam Pada Manusia menggunakan sistem pakar untuk membantu diagnosis penyakit dalam serta menentukan solusi dari penyakit tersebut dengan metode *forward chaining* dan *backward chaining* [4]. Penelitian dengan tema diagnosa

penyakit Tuberkulosis dilakukan untuk membangun sistem pakar berbasis web untuk mengenali diagnosa awal penyakit tuberkolosis pada manusia dewasa dengan metode probabilitas klasik [8]. Penelitian lainnya yang dilakukan oleh Madeso, Kabo, dan Batmetan melakukan rancang bangun sistem pakar untuk penentuan status gizi pada balita menggunakan metode *forward chaining* dengan *Visual Basic 6.0* [9]. Penelitian oleh Triyanto dan Fadlil menggunakan sistem

pakar untuk membantu diagnosa penyakit kelinci dengan metode penelusuran *forward chaining* dan metode kepastian *Theorema Bayes* [10]. Penelitian lainnya yang dilakukan oleh Kurmasela dan Sevani membangun sistem pakar untuk diagnosis gizi buruk pada balita menggunakan metode *certainty factor* dengan berbasis web [11].

Pada penelitian ini dilakukan pengembangan aplikasi sistem pakar untuk memudahkan masyarakat melakukan identifikasi panduan diet dengan Metode *Forward Chaining* berbasis android. Pada aplikasi ini, ditampilkan pertanyaan-pertanyaan gejala penyakit yang mungkin dialami seseorang. Dari gejala-gejala tersebut akan didefinisikan panduan diet manakah yang sesuai.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam penelitian ini adalah bagaimana mengembangkan sistem pakar untuk mengidentifikasi panduan diet dengan Metode *Forward Chaining* berbasis android.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Panduan diet yang dapat diidentifikasi adalah 8 jenis dengan 22 gejala penyakit.
2. Metode penalaran yang digunakan adalah *Forward Chaining*.
3. Metode ketidakpastian yang digunakan adalah Probabilitas Klasik.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengembangkan sebuah aplikasi berbasis android yang mampu membantu masyarakat umum dalam mengidentifikasi jenis panduan diet berdasarkan gejala-gejala yang diberikan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Manfaat praktis yang diharapkan adalah agar aplikasi dapat digunakan oleh masyarakat umum untuk mengidentifikasi panduan diet yang sesuai secara mudah.
2. Manfaat teoritis yang diharapkan adalah agar penelitian ini dapat berkontribusi bagi pelayanan gizi masyarakat pada umumnya.

II. TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan atau dalam bahasa inggris “*Artificial Intelligence*” adalah istilah yang merujuk kepada mesin yang memiliki kemampuan seperti manusia, yaitu dapat berpikir, mempertimbangkan tindakan yang akan diambil, dan dapat mengambil keputusan [12]. Kecerdasan buatan merupakan cabang ilmu komputer yang bertujuan membuat sebuah komputer dapat berpikir dan bernalar seperti manusia [10]. Kecerdasan buatan dapat membantu kegiatan manusia dalam mengambil keputusan, mencari informasi secara lebih akurat, serta membuat komputer lebih mudah digunakan dengan menggunakan bahasa alami yang mudah dipahami [12].

Sebagai makhluk yang cerdas, manusia dapat menyelesaikan masalah, membuat kesimpulan, mengenali pola, mengambil keputusan, menganalisa ucapan, dan sebagainya. Komputer yang diberikan basis pengetahuan dan inferensi akan dapat mengerjakan pekerjaan yang sebelumnya hanya dapat dikerjakan manusia. Dengan demikian, maka komputer akan menjadi alat yang lebih bermanfaat dan memberikan nilai tambah pada kemampuan manusia dalam rangka meningkatkan produktivitas dan efisiensi kerja manusia [13].

Kecerdasan merupakan kegiatan belajar, berpikir, dan mengamati dengan informasi sebagai objeknya. Untuk memecahkan masalah dengan kecerdasan buatan, maka digunakan basis pengetahuan, relasi, dan fakta dengan teknik pelacakan atau pencocokan pola. Komputasi pada kecerdasan buatan dilakukan dengan cara manipulasi simbol dan representasi. Pengetahuan ialah informasi yang terorganisir sehingga dapat dengan mudah dimengerti dan dapat diterapkan dalam pengambilan keputusan serta pemecahan masalah [13]. Sebagai bagian utama pada kecerdasan buatan, pengetahuan diperoleh melalui pendidikan dan pengalaman serta terdiri dari fakta, teori, pemikiran, prosedur, dan hubungannya satu dengan yang lain [5].

2.1.1 Bidang-Bidang Kecerdasan Buatan

Kecerdasan buatan merupakan kumpulan ide dan konsep yang berkaitan dengan sistem cerdas dan dapat dikembangkan dan diimplementasikan dalam area berbeda [14], yaitu:

- a. Sistem Pakar (*Expert System*)
- b. Komputasi Saraf
- c. Penerjemahan Bahasa
- d. Algoritma Genetika
- e. Agen Cerdas
- f. Sistem Robotik dan Sistem Sensor
- g. Pemrosesan Bahasa Alami/NLP (Natural Language Process)
- h. *Speech/Voice Understanding*
- i. *Computer Vision and Scene Recognition*
- j. *Intelligent Computer – Aided Instruction*

k. *Fuzzy Logic*

l. *Game Playing*

2.2 Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang dibuat untuk memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar atau ahli dalam suatu bidang tertentu dengan menggunakan pengetahuan, fakta, dan teknik penalaran. Pada dasarnya sistem pakar diterapkan untuk mendukung kegiatan pemecahan masalah seperti pembuatan keputusan (*decision making*), pemanduan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosa (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*), dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar [9].

Sistem pakar menyediakan penyelesaian dan solusi dengan kualitas pakar untuk masalah-masalah dalam suatu bidang spesifik. Sistem pakar meniru proses pemikiran serta pengetahuan pakar dalam menyelesaikan masalah tertentu. Pengetahuan sistem pakar umumnya diambil dari seseorang yang ahli dalam suatu bidang [14]. Pakar atau ahli adalah seseorang yang memiliki keahlian khusus untuk menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam, misalnya dokter atau montir. Dokter merupakan pakar yang mampu mendiagnosis penyakit yang diderita pasien dan dapat memberikan penatalaksanaan suatu penyakit. Sedangkan montir merupakan seseorang yang memiliki keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor atau mobil [15].

2.2.1 Ciri-Ciri Sistem Pakar

Ciri-ciri sistem pakar ialah [6]:

1. Terbatas pada domain atau bidang keahlian tertentu.
2. Mampu mengemukakan rangkaian alasan-alasan dengan cara yang dapat dipahami.
3. Dirancang untuk dapat dikembangkan secara bertahap.
4. Berdasarkan aturan atau kaidah tertentu yang telah dibuat.
5. *Output*-nya bersifat saran, anjuran, atau nasehat.
6. *Output*-nya tergantung pada dialog yang dilakukan sistem dengan *user*.
7. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.

2.2.2 Keunggulan Sistem Pakar

Keunggulan dari sistem pakar dibandingkan dengan seorang pakar adalah sebagai berikut [16]:

1. Sistem pakar dapat digunakan setiap hari sehingga tanpa istirahat karena menyerupai mesin, sedangkan seorang pakar tidak mungkin dapat bekerja terus menerus.
2. Sistem pakar dapat diperbanyak dan kemudian dibagikan ke berbagai lokasi berbeda untuk digunakan karena merupakan suatu perangkat lunak, sedangkan pakar hanya dapat bekerja pada satu tempat dan disaat yang tidak bersamaan.
3. Sistem pakar dapat diberikan sistem pengamanan untuk menentukan siapa saja yang memiliki hak akses menggunakannya dan jawaban yang diberikan bebas

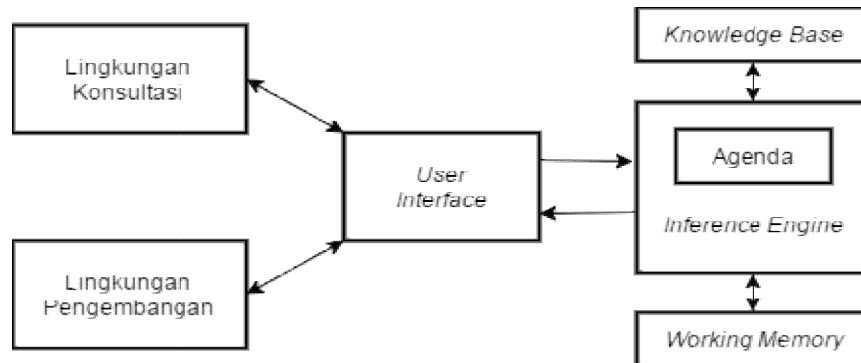
dari tekanan, sedangkan seorang pakar mungkin saja mendapat ancaman atau tekanan ketika melaksanakan tugas.

4. Dengan didukung pemeliharaan yang baik, pengetahuan suatu sistem pakar tidak akan hilang atau terlupa. Sedangkan pengetahuan seorang pakar manusia lambat laun akan hilang karena sakit, semakin tua, atau meninggal.
5. Sistem pakar memiliki kemampuan pemecahan masalah yang tidak terpengaruh dengan faktor luar seperti perasaan, intimidasi, dan faktor ekonomi. Sedangkan seorang pakar dalam menyelesaikan suatu masalah dapat saja dipengaruhi oleh faktor-faktor yang telah disebutkan di atas sehingga memberikan jawaban yang berbeda-beda dan mungkin tidak konsisten atas pertanyaan yang diajukan walaupun masalahnya sama.
6. Dalam menyelesaikan suatu masalah, sistem pakar relatif lebih cepat dibandingkan dengan seorang pakar manusia.
7. Biaya untuk memperkerjakan seorang pakar lebih mahal dibandingkan dengan penggunaan program sistem pakar (dengan asumsi program sistem pakar telah ada).

2.2.3 Konsep Sistem Pakar

Sistem pakar memiliki dua lingkungan pokok, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan merupakan lingkungan yang digunakan sebagai pembangun sistem pakar dari segi komponen maupun basis pengetahuan, sedangkan lingkungan konsultasi digunakan oleh seseorang yang bukan ahli atau

pakar untuk berkonsultasi. Berikut ini adalah komponen sistem pakar secara umum [17].



Gambar 1 Komponen Umum Sistem Pakar

1. *Knowledge Base* (Basis Pengetahuan)

Basis Pengetahuan merupakan komponen sistem pakar yang berisi pengetahuan-pengetahuan dalam pemecahan masalah. Pada proses ini, pengetahuan direpresentasikan menjadi basis pengetahuan dan basis aturan yang kemudian dikodekan, dikumpulkan dan dibentuk secara sistematis. Ada dua elemen utama basis pengetahuan, yaitu fakta dan aturan (*rule*).

2. *Inference Engine* (Mesin Inferensi)

Mesin inferensi merupakan suatu modul dalam sistem pakar yang menjadi tempat dilakukannya proses inferensi (menghasilkan informasi baru dari informasi yang telah diketahui). Pada mesin inferensi terdapat agenda, yaitu aksi-aksi potensial yang berhubungan dengan suatu masalah.

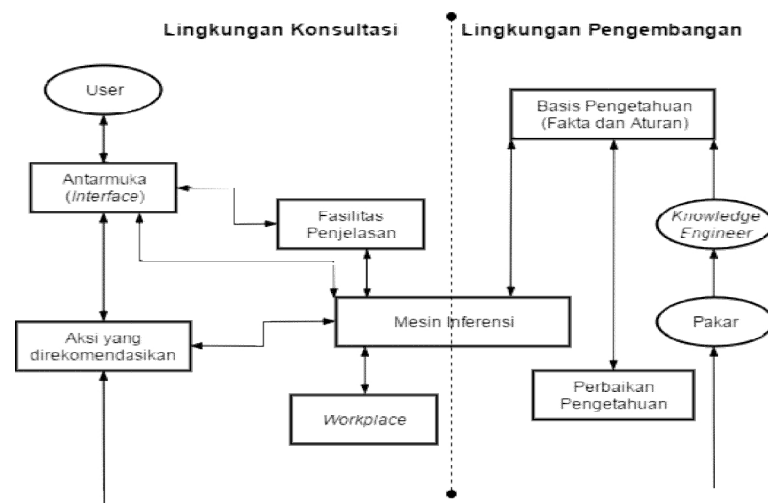
3. *Working Memory*

Working memory adalah bagian sistem pakar yang digunakan untuk merekam kejadian sementara, termasuk keputusan sementara. Isi dari bagian sistem pakar ini adalah fakta-fakta masalah yang ditemukan dalam suatu proses konsultasi. Struktur dari *working memory* mengikuti alur inferensi sistem pakar.

4. *User Interface* (Antarmuka Pengguna)

Antarmuka pengguna adalah bagian dari sistem pakar yang merupakan mekanisme atau media komunikasi antara pengguna (*user*) dengan sistem. Bagian ini juga memberikan fasilitas informasi dan keterangan yang mengarah pada penelusuran masalah sampai ditemukannya solusi.

2.2.4 Arsitektur Sistem Pakar



Gambar 2 Arsitektur Sistem Pakar

Komponen yang terdapat dalam sistem pakar pada Gambar 2 meliputi hal-hal berikut [16]:

1. Antarmuka Pengguna (*User Interface*)

User interface adalah bagian dari sistem pakar yang merupakan mekanisme yang digunakan untuk berinteraksi antara pengguna dengan sistem pakar. Antarmuka menerima informasi dari pengguna dan mengubahnya menjadi bentuk yang dimengerti sistem. Antarmuka juga memungkinkan sistem pakar untuk menerima perintah dan *input* dari pengguna serta menyajikan informasi (*output*) kepada pengguna.

2. Aksi yang Direkomendasikan

Bagian ini merupakan saran atau solusi yang direkomendasikan untuk permasalahan yang dihadapi pengguna atau *user*.

3. Basis pengetahuan (*Knowledge Base*)

Basis pengetahuan adalah bagian paling penting pada sistem pakar karena keahlian pakar disimpan didalamnya. Isi dari basis pengetahuan adalah fakta-fakta yang didapat dari ahli dan diimplementasikan pada sistem komputer dengan metode representasi pengetahuan tertentu.

4. *Knowledge Engineer*

Knowledge Engineer adalah spesialis sistem yang menerjemahkan pengetahuan yang diperoleh dari pakar atau ahli menjadi pengetahuan yang dapat disimpan ke dalam basis pengetahuan sistem pakar.

5. Akuisisi pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah proses memperoleh pengetahuan dari seorang pakar yang dilakukan oleh pengolah pengetahuan (*knowledge engineer*). Pengolah pengetahuan mewawancarai pakar dan mengumpulkan pengetahuan yang ada.

6. Mesin Inferensi

Mesin inferensi merupakan bagian dari sistem pakar yang bertugas untuk menemukan konklusi atau solusi yang tepat dari sekian banyak konklusi atau solusi yang ada. Proses dalam mesin inferensi adalah memberikan metodologi penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace* untuk membuat kesimpulan.

7. *Workplace* (Tempat Kerja)

Workplace ialah area kerja dari sekumpulan memori kerja (*working memory*) dimana masalah yang diperoleh dari input dideskripsikan. *Workplace* juga digunakan untuk perekaman hipotesis dan keputusan sementara.

8. Fasilitas Penjelasan Sistem

Fasilitas penjelasan sistem adalah komponen tambahan pada sistem pakar yang memberikan penjelasan tentang bagaimana menjalankan program, apa yang harus dijelaskan kepada *user*, memberikan saran kepada *user*, mengakomodasi kesalahan *user*, dan menjelaskan bagaimana suatu masalah terjadi.

9. Perbaikan Pengetahuan

Seorang pakar memiliki kemampuan perbaikan pengetahuan, yaitu mereka dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar dari pengetahuan tersebut, dan meningkatkan untuk konsultasi mendatang.

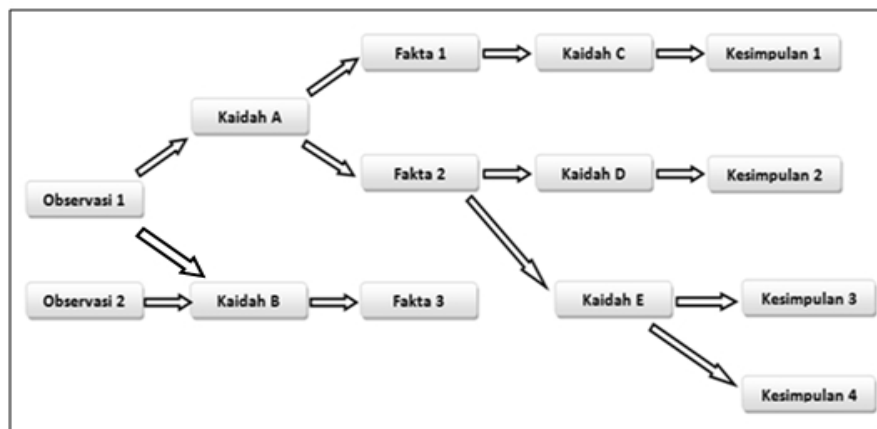
2.2.5 Mekanisme Inferensi

Mekanisme inferensi adalah bagian dari sistem pakar yang melakukan penalaran berdasarkan aturan dan pola tertentu. Selama proses konsultasi antara sistem dan pemakai, mekanisme inferensi menguji aturan satu demi satu sampai kondisi memenuhi pola aturan tertentu. Fungsi mekanisme inferensi adalah pembuktian hipotesis. Bila hipotesis telah dimasukkan ke dalam sistem pakar, maka mekanisme inferensi mengecek apakah hipotesis sudah ada dalam basis data atau belum. Jika sudah ada, maka hipotesis dianggap sebagai fakta yang telah dibuktikan sehingga operasi tidak perlu dilanjutkan. Ada dua metode inferensi

penting dalam sistem pakar, yaitu *Forward Chaining* dan *Backward Chaining* [7]. Dalam penelitian ini digunakan metode inferensi *Forward Chaining*.

2.2.5.1 Metode *Forward Chaining*(Runut Maju)

Forward chaining adalah strategi pengambilan keputusan yang dimulai dari pengambilan premis (fakta) menuju konklusi (kesimpulan akhir). Pemecahan masalah pada *forward chaining* dimulai dari fakta masalah yang diberikan dan kumpulan langkah-langkah sah atau aturan-aturan untuk berpindah *state*. Tanda-tanda atau kunci keberhasilan akan terkumpul dengan sendirinya ketika mulai mengarah ke kesimpulan. Contoh diagram *forward chaining* dapat dilihat pada Gambar 3 [9].



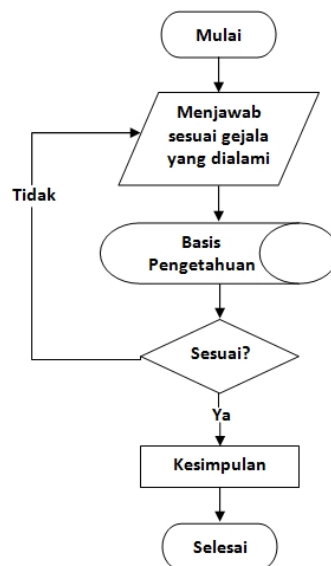
Gambar 3 *Forward Chaining*

Ciri-ciri dari metode *forward chaining* adalah [18] :

1. Dalam metode ini, dilakukan pemrosesan yang diawali dari sekumpulan data untuk selanjutnya dilakukan inferensi sesuai kaidah atau aturan yang ditetapkan sehingga didapatkan kesimpulan yang optimal.
2. Mesin inferensi dengan metode ini akan terus melakukan looping pada prosesnya untuk mencapai hasil yang sesuai.

3. Kelebihan metode ini ialah dapat menambahkan databaru ke dalam *database* inferensi dan dapat melakukan perubahan aturan inferensi.

Pencocokan fakta pada inferensi *forward chaining* dimulai dari bagian sebelah kiri atau IF terlebih dahulu. Misalnya, IF A AND B THEN C, artinya jika premis A dan B bernilai benar sesuai dengan situasi maka akan menghasilkan konklusi C. Proses inferensi adalah bagian dari sistem pakar yang melakukan penalaran atau pelacakan dengan menggunakan isi daftar aturan dan pola tertentu. Aliran proses inferensi dapat dilihat pada Gambar 4 [11].



Gambar 4 Flowchart Proses Inferensi

Pelacakan ke depan (*forward chaining*) adalah dimulai dari sekumpulan data menuju kesimpulan. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan [18].



Gambar 5 Penalaran Pelacakan ke Depan

Operasi dari sistem *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui ke dalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai *goal* atau tidak ada lagi premis yang cocok dengan fakta yang diketahui. Proses pencarian *forward chaining* dimulai dari premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi (*THEN*) atau dapat dimodelkan sebagai berikut.

1. Informasi masukan dapat berupa data, bukti, temuan, atau pengamatan.
2. Konklusi dapat berupa tujuan, hipotesa, penjelasan, atau diagnosis.

Sehingga jalannya penalaran *forward chaining* dapat dimulai dari data menuju tujuan, dari bukti menuju hipotesa, dari temuan menuju penjelasan, atau dari pengamatan menuju diagnosis [18].

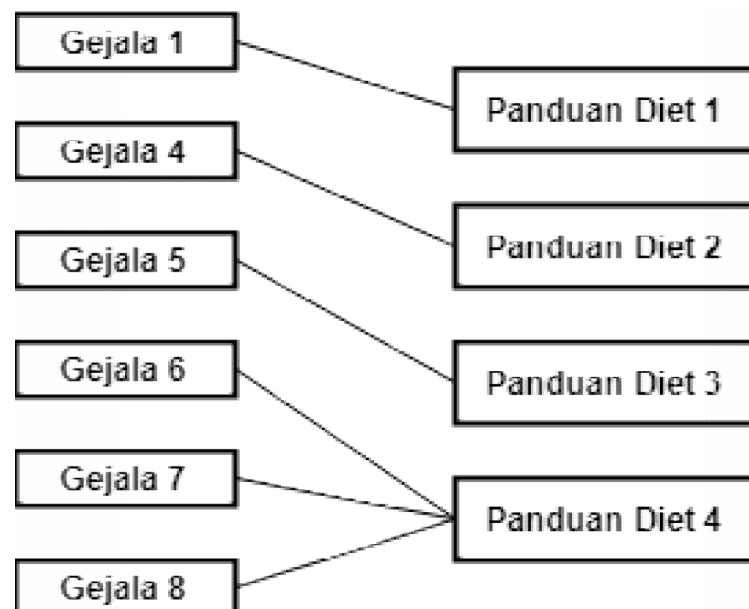
Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan manakah yang akan dijalankan, kemudian aturan tersebut dijalankan. Proses diulang sampai ditemukan suatu hasil [15]. Untuk memudahkan pemahaman mengenai metode ini, akan diberikan ilustrasi kasus sebagai berikut.

Tujuan: Diharapkan dapat memperoleh konklusi pedoman diet yang sesuai berdasarkan premis-premis dalam aturan dan fakta yang diberikan oleh user, berikut adalah daftar aturannya.

Tabel 1 Contoh Ilustrasi Aturan dan Konklusi

Aturan 1	Aturan 2	Aturan 3	Aturan 4
Jika Gejala 1	Jika Gejala 4	Jika Gejala 5	Jika Gejala 6
			Dan Gejala 7
			Dan Gejala 8
Maka Panduan Diet 1	Maka Panduan Diet 2	Maka Panduan Diet 3	Maka Panduan Diet 4

Penelusuran maju pada contoh kasus ini adalah untuk mengetahui apakah suatu fakta yang dialami oleh pengguna termasuk panduan diet 1, panduan diet 2, panduan diet 3, panduan diet 4 atau bahkan bukan salah satu dari konklusi tersebut, yang artinya sistem belum dapat mengambil kesimpulan karena keterbatasan aturan. Berikut adalah penggambaran aturan contoh kasus di dalam sebuah *graph* yang memetakan antara premis-premis dan konklusi-konklusi.

**Gambar 6** *Graph* Pengetahuan

Dalam penalaran ini, *user* diminta memasukkan premis-premis atau gejala-gejala yang dialami. Untuk memudahkan pengguna, sistem memunculkan pertanyaan

gejala dan *user* dapat memberikan umpan balik gejala mana yang dialami dengan memilih tombol “ya” atau “tidak”. Dalam contoh kasus ini, daftar gejalanya adalah:

Gejala 1, Gejala 4, Gejala 5,

Gejala 6, Gejala 7, Gejala 8

Berdasarkan premis-premis yang dipilih oleh *user*, maka sistem akan mencari aturan yang sesuai, sehingga akan diperoleh konklusinya. Apabila *user* memilih Gejala 1, maka aturan yang terpilih adalah aturan 1 dengan konklusinya adalah Panduan Diet 1. Apabila *user* memilih Gejala 6 dan Gejala 7, maka sistem akan mengarah pada aturan 4 dengan konklusinya adalah Panduan Diet 4. Tetapi, karena aturan 4 premisnya adalah Gejala 6, Gejala 7, dan Gejala 8, maka premis-premis yang dipilih oleh *user* tidak cukup untuk mengambil kesimpulan Panduan Diet 4 sebagai konklusi terpilih.

2.2.5.2 Penanganan Data Tidak Lengkap

Teknik penyelesaian *forward chaining* merupakan cara dengan asumsi bahwa untuk mengambil suatu kesimpulan semua premis harus terpenuhi. Untuk kasus-kasus tertentu, seorang pakar akan tetap dapat mengambil kesimpulan meskipun premis yang diberikan tidak lengkap. Contohnya dipilih Gejala 7 saja, maka sistem akan tetap dapat mengambil kesimpulan meskipun tingkat kemungkinan tidak 100%. Misalnya masalah yang biasa diputuskan oleh seorang pakar dalam penyakit tanaman. Untuk menyimpulkan bahwa tanaman mengalami suatu penyakit haruslah mengalami gejala, ada kalanya premis yang harusnya diketahui tidak dapat diketahui karena sebab tertentu. Oleh karena itu, pakar akan

mengambil kesimpulan tanaman mengalami penyakit tertentu dengan tingkat kemungkinan yang tidak 100% [15].

Dalam dunia medis, kita tidak dapat memutuskan tingkat akurasi dari suatu penyakit yang diderita oleh seseorang secara mutlak. Hal ini dikarenakan pemeriksaan medis memiliki tahapan yang saling berkesinambungan [3]. Dalam penelitian ini, probabilitas klasik digunakan untuk mengurutkan konklusi berdasarkan pengetahuan yang dimiliki sistem.

2.3 Probabilitas Klasik

Probabilitas adalah cabang ilmu matematika yang mempelajari kemungkinan terjadinya suatu kejadian diantara seluruh kejadian yang mungkin terjadi. Dalam pendekatan probabilitas klasik, dapat diketahui tingkat kepastian kemungkinan terjadinya suatu kejadian. Persamaan probabilitas klasik dalam diagnosis penyakit pada sistem pakar adalah sebagai berikut [8]:

$$P A = \frac{n(A)}{n}$$

Keterangan:

$P A$ = Peluang terjadinya penyakit A

$n A$ = Banyaknya gejala penyakit A yang muncul

n = Total gejala penyakit A

Probabilitas klasik digunakan untuk mendapatkan peluang kemungkinan dari suatu penyakit, sehingga untuk menghitung presentase kemungkinan penyakit adalah:

$$Presentase A = P A \times 100\%$$

2.4 Sistem Operasi Android

Android adalah sebuah sistem operasi perangkat *mobile* yang pada awalnya dikembangkan oleh Android, Inc. Android menyediakan *platform* terbuka bagi para *developer* atau pengembang untuk menciptakan aplikasi mereka sendiri. Tidak ada ketentuan yang tetap dalam konfigurasi *Software* dan *Hardware* di dalam Android [12]. Fitur-fitur yang didapat dalam Android antara lain [7]:

1. *Storage* - Menggunakan SQLite.
2. *Connectivity* - Mendukung GSM/EDGE, IDEN, CDMA, EV-Do, UMTS.
3. *Bluetooth* (termasuk A2DP dan AVRCP), WiFi, LTE, dan WiMax.
4. *Messaging* - Mendukung SMS dan MMS.
5. *Web Browser* - Berbasiskan *open-source* WebKit.
6. *Chrome's V8 JavaScript*
7. *Media Support* - Mendukung beberapa media dalam bentuk 3GP atau MP4, MP3, MIDI, Ogg Vorbis, WAV, JPEG, GIF, dan BMP.
8. *Hardware Support* - Akselerasi sensor, kamera, kompas digital, sensor kedekatan, GPS.
9. *Multi-touch* - Mendukung *multi-touch screens*.
10. *Multi-tasking* - Mendukung aplikasi *multi-tasking*.
11. *Flash-support* - Android 2.3 mendukung *Flash* 10.1
12. *Tethering* - Mendukung pembagian koneksi Internet sebagai *wired/wireless hotspot*.

13. *Playstore* - Layanan konten digital milik Google.
14. Lingkungan pengembangan yang luas, termasuk *emulator*, peralatan *debugging*, dan *plugin* untuk Eclipse IDE.

2.5 Panduan Diet dan Penyakit

2.5.1 Panduan Diet pada Ilmu Gizi

Istilah gizi dan ilmu gizi di Indonesia mulai dikenal sekitar tahun 1952-1955 sebagai terjemahan bahasa Inggris *nutrition*. Kata gizi berasal dari bahasa Arab “*ghidza*” yang berarti makanan. Menurut dialek Mesir, *ghidza* dibaca ghizi. Selain itu, orang mulai menerjemahkan *nutrition* dan mengejanya sebagai “nutrisi” [9]. Gizi (sering disebut juga nutrisi) diartikan sebagai proses dalam tubuh makhluk hidup untuk memanfaatkan makanan guna pembentukan energi, tumbuh kembang, dan pemeliharaan tubuh sedangkan ilmu gizi mempelajari proses tersebut [1]. Ilmu gizi berkaitan dengan makanan dan tubuh manusia. Di dalam ilmu gizi, dipelajari segala sesuatu tentang makanan dan hubungannya dengan kesehatan yang optimal [19].

Istilah diet sering disalahartikan oleh orang awam sebagai pembatasan makanan pada penderita suatu penyakit. Definisi sebenarnya dari istilah diet adalah pengaturan jumlah makanan yang dimakan setiap hari agar seseorang tetap sehat [1]. Makanan ialah bahan selain obat yang didalamnya terkandung zat-zat gizi serta unsur-unsur dan ikatan kimia yang dapat diubah menjadi zat gizi yang berguna untuk tubuh [19]. Panduan diet memanfaatkan pengetahuan gizi untuk mempercepat kesembuhan dan memperbaiki status gizi serta dapat menekan

progresivitas penyakit. Panduan diet tidak dapat dipisahkan dari penyembuhan suatu penyakit. Dalam ilmu kedokteran modern, hal ini dipandang bukan hanya sekedar pemberian makan, melainkan menjadi terapi medis seperti halnya pengobatan dan tindakan medis lainnya [1].

2.5.2 Penyakit

Dalam penelitian ini, penyakit yang dijadikan objek untuk identifikasi panduan diet meliputi Tekanan Darah Tinggi, Osteo Arthritis, Gout, Gastritis, Infeksi Hati, Batu Empedu, Penurunan Fungsi Ginjal, dan Kencing Manis. Pemilihan berbagai penyakit tersebut didasarkan pada pertimbangan penyakit yang sering dijumpai di masyarakat.

2.5.2.1 Tekanan Darah Tinggi

Tekanan Darah Tinggi atau Hipertensi adalah suatu keadaan ketika tekanan darah di pembuluh darah meningkat secara persisten/menetap diatas normal, yaitu $\geq 140/90$ mmHg [20]. Hipertensi seringkali tidak menimbulkan gejala dan apabila terjadi dalam jangka waktu lama dapat menimbulkan komplikasi, sehingga perlu deteksi lebih dini [21].

2.5.2.2 Osteo Arthritis

Osteo Arthritis merupakan kelainan degeneratif pada persendian yang lebih sering dijumpai pada orang tua dan kasus obesitas. Penyakit ini terutama menyerang sendi-sendi panjang atau sendi penyangga tubuh [1].

2.5.2.3 Gout

Gout ialah penyakit dimana kadar asam urat dalam darah di atas nilai normal dan merupakan sindrom klinis dengan gambaran khas peradangan sendi akut akibat

reaksi sendi terhadap pembentukan kristal urat. Gout berhubungan dengan gangguan metabolisme purin [1].

2.5.2.4 Gastritis

Gastritis adalah peradangan atau iritasi di lambung [3]. Terjadinya gastritis dapat disebabkan oleh pola makan yang tidak teratur, infeksi bakteri atau virus, efek samping penggunaan obat tertentu, dan stress [22].

2.5.2.5 Infeksi Hati

Infeksi Hati (Hepatitis) adalah peradangan hati yang dapat disebabkan oleh virus A, B, C, konsumsi alkohol yang berlebihan dan zat-zat kimia hepatotoksik seperti kloroform serta karbon tetraklorida [1]. Infeksi virus hepatitis terutama menular melalui jalur fekal-oral dan juga melalui air serta makanan yang terkontaminasi [23].

2.5.2.6 Batu Empedu

Batu Empedu adalah keadaan penyakit berupa terbentuknya batu di dalam kandung empedu maupun saluran empedu (duktus keledokus) [23]. Beberapa faktor gizi yang turut menyebabkan terjadinya penyakit ini antara lain adalah obesitas dan konsumsi makanan sumber lemak yang tinggi [1].

2.5.2.7 Penurunan Fungsi Ginjal

Penurunan Fungsi Ginjal adalah keadaan dimana ginjal mengalami penurunan fungsi akibat berbagai faktor, misalnya infeksi, tumor, kelainan bawaan, dan lain-lain [24]. Keadaan tersebut dapat menyebabkan gagal ginjal yang progresif [1].

2.5.2.8 Kencing Manis

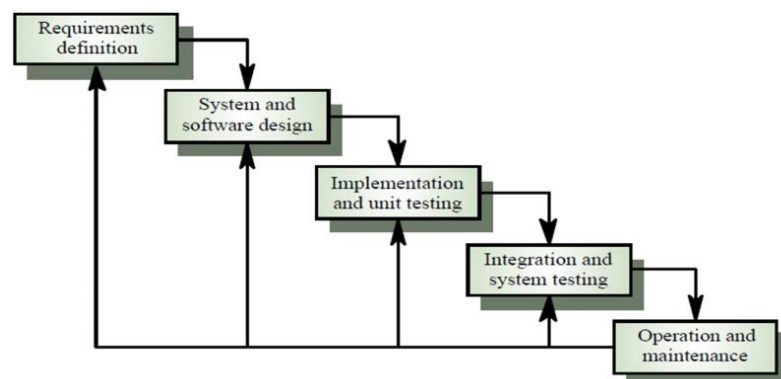
Menurut *American Diabetes Association* (ADA) tahun 2010, kencing Manis atau Diabetes Mellitus (yang dimaksud di sini adalah Diabetes Mellitus Tipe 2) adalah suatu kelompok penyakit metabolik dengan keadaan kadar gula darah lebih tinggi dari nilai normal (hiperglikemia) yang terjadi karena kelainan sekresi insulin, kerja insulin, atau keduanya [25].

2.6 Metode Pengembangan Sistem

Adapun metode pengembangan sistem yang digunakan dalam penelitian ini adalah Metode *Waterfall* dan desain menggunakan *Unified Modelling Language* (UML).

2.6.1 Metode *Waterfall*

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak dimana setiap tahapan yang dilalui harus menunggu tahapan yang sebelumnya selesai dikerjakan. Metode ini bersifat sistematis dan sekuensial. Tahapan yang dilakukan dalam pengembangan aplikasi pada penelitian ini sesuai dengan Metode *Waterfall* pada Gambar 7 [26].



Gambar 7 Metode *Waterfall*

Berikut merupakan penjelasan tahapan pengembangan aplikasi dengan Metode *Waterfall* [27].

1. *Requirements Definition* (Definisi/Analisis Kebutuhan Aplikasi)

Pada tahapan yang menjadi dasar pembuatan sistem ini dilakukan penetapan fitur, kendala, dan tujuan sistem melalui konsultasi dengan pengguna sistem. Hal-hal tersebut akan ditetapkan secara detail yang akan dijadikan spesifikasi dari sistem yang akan dibangun.

2. *System and Software Design* (Desain Aplikasi)

Pada tahapan ini akan dilakukan pembentukan arsitektur sistem berdasarkan persyaratan yang ditetapkan pada tahap sebelumnya. Selain itu, dalam tahapan ini juga dilakukan identifikasi serta penggambaran dasar sistem perangkat lunak beserta hubungan-hubungannya.

3. *Implementation* (Penerapan Desain dan Penulisan Kode Program)

Tahapan ini merupakan tahap penerjemahan desain sistem yang telah dibuat ke dalam bentuk perintah yang dimengerti komputer menggunakan bahasa pemrograman. Penelitian ini menggunakan basa pemrograman Java dan Eclipse sebagai *software* pengembangan aplikasinya. Hasil dari tahapan ini berupa *file installer* dengan ekstensi .apk.

4. *Integration and System Testing* (Pengujian Aplikasi)

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem telah memenuhi persyaratan. Pengujian aplikasi pada penelitian ini menggunakan metode *Blackbox Testing*. Pengujian dilakukan secara menyeluruh tanpa melihat struktur

internal aplikasi atau komponen yang diuji. *Blackbox Testing* berfokus pada kebutuhan fungsional aplikasi berdasarkan spesifikasi kebutuhan aplikasi.

5. *Operation and Maintenance* (Penerapan dan Perawatan Aplikasi)

Dalam tahapan ini, aplikasi telah siap diterapkan pada perangkat *mobile* dan siap digunakan sesuai dengan tujuan dibuatnya aplikasi. Perawatan, perbaikan, serta pengembangan dilakukan untuk menjaga kualitas dan kestabilan aplikasi dan memperbaiki error yang tidak ditemukan pada tahap pembuatan.

2.6.2 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi perancangan dan pendokumentasian sistem perangkat lunak. Bahasa pemodelan ini dapat digunakan untuk membuat model dari semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat dijalankan pada perangkat keras sistem operasi dan jaringan apapun, dan dapat ditulis menggunakan bahasa pemrograman apapun [28]. Beberapa diagram dalam UML (*Unified Modelling Language*) adalah sebagai berikut:

1. *Use-Case Diagram*, yaitu diagram yang memperlihatkan sekumpulan *use case* dan actor-aktor. Diagram ini penting untuk mengorganisasikan dan memodelkan perilaku suatu sistem yang dibutuhkan dan diharapkan oleh *user*.
2. *Activity Diagram*, yaitu diagram yang memperlihatkan organisasi serta kebergantungan komponen-komponen yang ada. Diagram ini berhubungan dengan *class diagram*, dimana komponen dipetakan ke dalam satu kelas atau lebih dan satu antarmuka atau lebih.

3. *Class Diagram*, yaitu diagram yang memperlihatkan kumpulan dari kelas-kelas, antarmuka, kolaborasi, serta relasi. *Class* digambarkan dengan segi empat yang dibagi beberapa bagian. Bagian atas merupakan nama dari *class* (*class name*), bagian tengah merupakan struktur dari *class* (atribut), dan bagian bawah merupakan sifat dari *class* (metode/*operations*).
4. *Sequence Diagram*, yaitu diagram yang menggambarkan interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

2.7 Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses eksekusi sistem atau program dengan tujuan menemukan *error* atau mengevaluasi atribut atau kemampuan sistem tersebut dan menentukan bahwa sistem telah memenuhi kebutuhan yang ditetapkan. Pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak dengan spesifikasinya [29]. Ada beberapa jenis pengujian perangkat lunak antara lain [30]:

1. *White Box*, yaitu pengujian yang didasarkan pada pengecekan detail perancangan dengan menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian.
2. *Black-Box*, yaitu pengujian yang berfokus pada spesifikasi fungsional suatu perangkat lunak.

Dalam penelitian ini, digunakan pengujian perangkat lunak dengan pendekatan *black-box*. Pengujian *black-box* bertujuan untuk menemukan kesalahan dalam

beberapa kategori, diantaranya: fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan performa, kesalahan inisialisasi, dan terminasi. *Black-box testing* adalah metode pengujian sistem dimana *tester* hanya mengetahui apa yang harus dilakukan suatu sistem. *Tester* atau penguji dalam metode ini tidak perlu mengetahui bagaimana cara sistem beroperasi. Dalam pengujian *black-box*, *tester* hanya memperhatikan hasil dari apa yang dimasukkan (*input*) tanpa harus mengetahui bagaimana atau mengapa bisa demikian [31].

2.7.1 Equivalence Partitioning (EP)

Equivalence Partitioning (EP) adalah teknik pengujian *black-box* yang membagi domain *input* dari program ke dalam kelas-kelas sehingga *test case* dapat diperoleh. *Equivalence Partitioning* mendefinisikan uji kasus untuk menemukan sejumlah kesalahan dan mengurangi jumlah uji kasus yang harus dibuat. Uji kasus yang didesain untuk *Equivalence Partitioning* adalah berdasarkan evaluasi kelas ekuivalensi untuk kondisi *input* yang menggambarkan kumpulan keadaan valid atau tidak valid. Kondisi *input* dapat berupa nilai numerik, *range* nilai, kumpulan nilai yang berhubungan, atau kondisi *Boolean* [12].

2.7.2 Skala Likert

Skala *Likert* adalah metode penskalaan pernyataan sikap dengan menggunakan respon yang didistribusikan sebagai dasar penentuan nilai skalanya. Nilai skala setiap pertanyaan ditentukan oleh distribusi *respons* setuju dan tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba [32]. Skala *Likert* berisi lima tingkat preferensi jawaban dengan pilihan pada Tabel 2.

Tabel 2 Tingkat Preferensi Jawaban Skala *Likert*

Angka Preferensi	Keterangan Preferensi
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Ragu-Ragu atau Netral
4	Setuju
5	Sangat Setuju

Rumus penentuan presentase penilaian berdasarkan kriteria pada Skala *Likert* adalah menggunakan rumus aritmatika *mean* berikut [32]:

$$P = \frac{Xi}{n N} (100\%)$$

Keterangan:

P = Presentase pernyataan

Xi = Nilai kualitatif total

n = Jumlah responden

N = Nilai kategori pernyataan terbaik

Selanjutnya, penentuan interval per kategori dihitung menggunakan rumus berikut:

$$I = \frac{100\%}{K}$$

Keterangan:

I = Interval

K = Kategori interval

III.METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung dan Rumah Sakit Umum Daerah (RSUD) DR. A. Dadi Tjokrodipo di Jl. Basuki Rahmat No.73, Sumur Putri, Teluk Betung Utara, Kota Bandar Lampung. Waktu penelitian dilaksanakan pada periode Semester Genap Tahun 2016/2017.

3.2 Perangkat Penelitian

Perangkat pendukung yang digunakan dalam penelitian ini adalah sebagai berikut.

A. Perangkat Keras

- *Notebook* ACER Aspire 4253
- *Smartphone* Android (Smartfren Andromax AD688G Android 4.3 API 18)

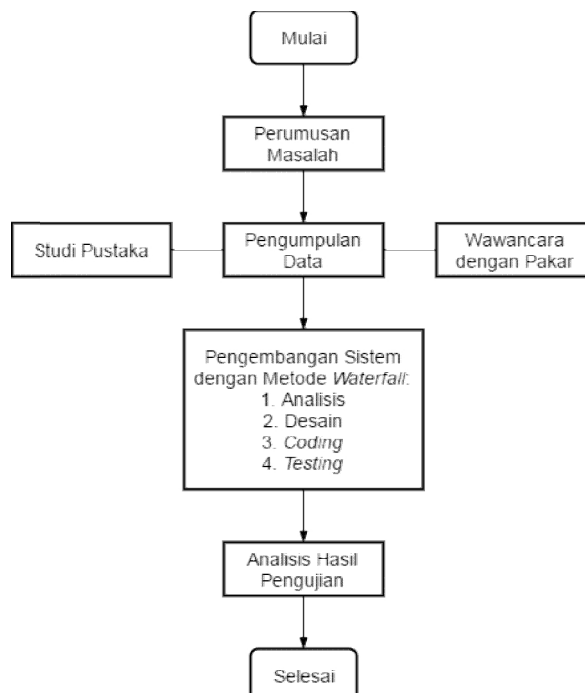
B. Perangkat Lunak

- Sistem Operasi Windows 7, 64-bit.
- *Android Studio*
- *Adobe Photoshop CS3*, sebagai media pembuatan *user interface*.

- *StarUML*, sebagai media pembuatan desain rancangan sistem.
- *SQLite Manager*, sebagai platform pembuatan database.

3.3 Tahapan Penelitian

Tahapan penelitian merupakan tahapan yang dilakukan peneliti dalam melaksanakan penelitian. Tahapan penelitian sistem pakar identifikasi gangguan gizi dengan metode *forward chaining* ini digambarkan pada Gambar 8.



Gambar 8 Diagram Alir Penelitian

3.3.1 Tahap Perumusan Masalah

Tahap perumusan masalah merupakan tahapan dimana peneliti merumuskan dan membatasi masalah yang akan diteliti. Perumusan dan pembatasan masalah diperlukan untuk mengarahkan peneliti dalam pembuatan sistem sehingga sistem yang dibuat tidak keluar dari batasan yang telah ditetapkan.

3.3.2 Tahap Pengumpulan Data

Tahap pengumpulan data dilakukan dengan menggunakan dua metode, yaitu:

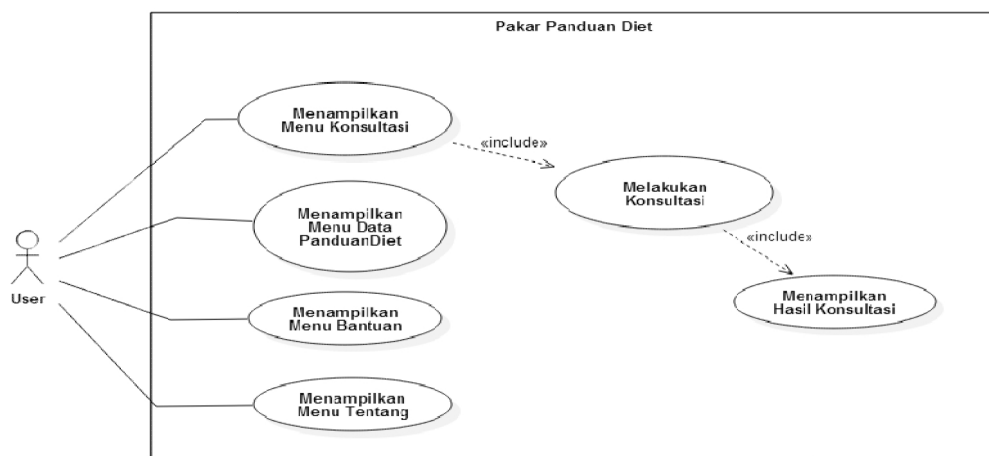
- a. Studi Pustaka, yaitu data yang dikumpulkan berasal dari literatur berupa buku, jurnal, dan dokumen yang berkaitan dengan penelitian.
- b. Wawancara, yaitu data yang dikumpulkan berasal dari proses wawancara dengan ahli/pakar. Wawancara bertujuan untuk mendapatkan data yang tidak ditemukan pada metode studi pustaka.

3.3.3 Tahap Pengembangan Sistem

Dalam tahap pengembangan sistem, digunakan metode *waterfall* yang terdiri dari empat tahapan sebagai berikut.

3.3.3.1 Software Requirements Analysis

Pada tahapan ini dilakukan analisis terhadap kebutuhan sistem. *Output* dari tahapan ini dalam model *use case diagram* untuk menggambarkan fungsi *interface* dari sisi pengguna atau *user*. Desain *use case diagram* dapat dilihat pada Gambar 9.



Gambar 9 *Use-case Diagram*

Use-case Diagram di atas menjelaskan bahwa *user* dapat mengakses empat buah aktivitas yang dilakukan sistem, yaitu menampilkan menu konsultasi (termasuk melakukan konsultasi dan menampilkan hasil konsultasi), menampilkan menu data panduan diet, menampilkan menu bantuan, dan menampilkan menu tentang.

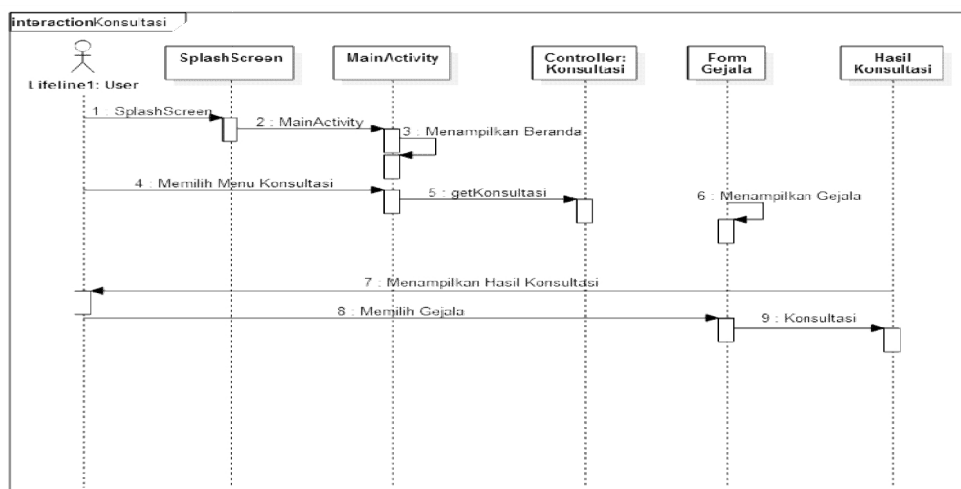
3.3.3.2 Design

Pada tahapan ini dilakukan perancangan desain sistem yang harus dapat mengimplementasikan kebutuhan yang telah didefinisikan pada tahapan sebelumnya. Perancangan pada tahapan ini menggunakan metode UML (*Unified Modelling Language*). *Output* dari tahapan ini berupa *sequence diagram* dan *activity diagram*, serta *interface* (antarmuka) aplikasi.

1. Sequence Diagram

Sequence diagram menggambarkan spesifikasi dari pilihan *button* sehingga *user* dapat memilih *button* tersebut dan akan ditampilkan *sub-menu* dari *button*. Pada aplikasi atau sistem yang akan dibangun ini terdapat empat *sequence diagram*.

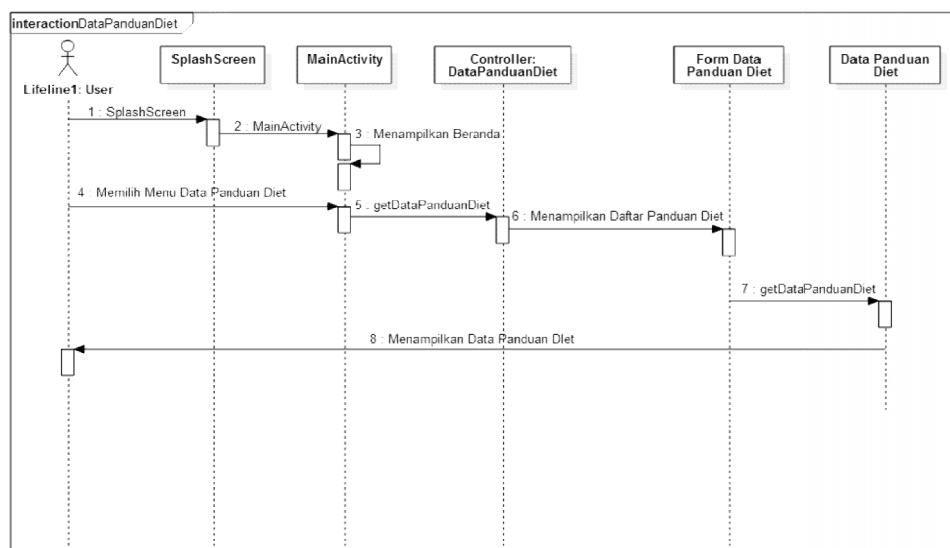
a. Sequence Diagram Konsultasi



Gambar 10 *Sequence Diagram* Konsultasi

User harus mengakses aplikasi terlebih dahulu untuk dapat mengakses menu konsultasi. Aplikasi akan menampilkan halaman *splash screen* selama beberapa detik dan selanjutnya akan muncul halaman beranda aplikasi dimana *user* dapat memilih Menu Konsultasi. Aplikasi akan menampilkan sub-menu mulai setelah *user* memilih Menu Konsultasi. Sistem akan menampilkan pertanyaan gejala dan sistem akan memproses jawaban *user* serta menampilkan hasil konsultasi.

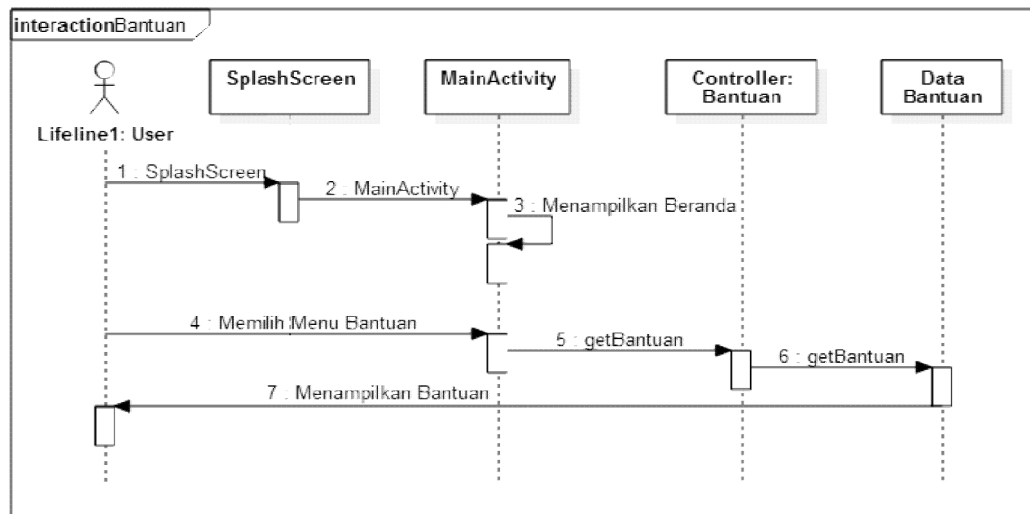
b. *Sequence Diagram* Data Panduan Diet



Gambar 11 *Sequence Diagram* Data Panduan Diet

User harus mengakses aplikasi terlebih dahulu untuk dapat memilih Menu Data Panduan Diet. Aplikasi akan menampilkan halaman *splash screen* selama beberapa detik dan selanjutnya akan muncul halaman beranda aplikasi dimana *user* dapat memilih Menu Panduan Diet dan aplikasi akan menampilkan daftar panduan diet yang dapat dilihat detailnya.

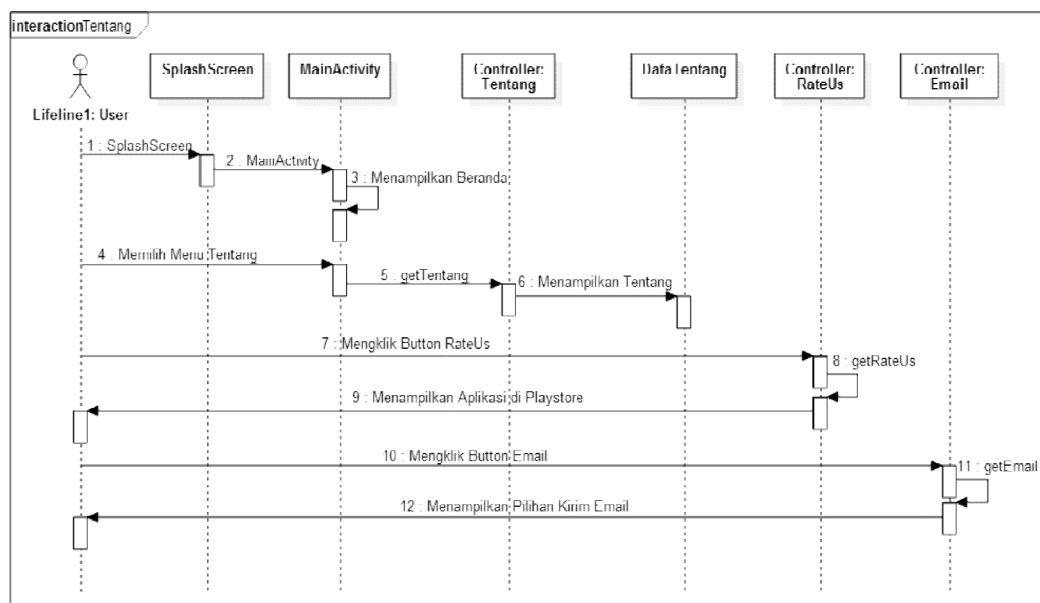
c. *Sequence Diagram Bantuan*



Gambar 12 *Sequence Diagram Bantuan*

User harus mengakses aplikasi terlebih dahulu untuk dapat memilih Menu Bantuan. Aplikasi akan menampilkan halaman *splash screen* selama beberapa detik dan selanjutnya akan muncul halaman beranda aplikasi dimana user dapat memilih Menu Bantuan dan aplikasi akan menampilkan halaman bantuan.

d. *Sequence Diagram Tentang*



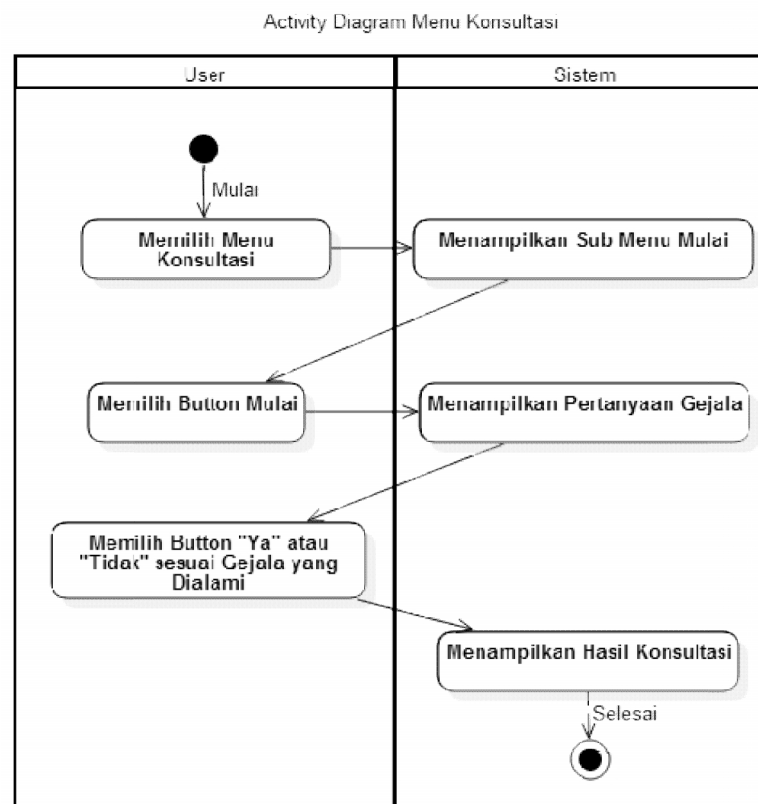
Gambar 13 *Sequence Diagram Tentang*

User harus mengakses aplikasi terlebih dahulu untuk dapat memilih Menu Tentang. Aplikasi akan menampilkan halaman *splash screen* selama beberapa detik dan selanjutnya akan muncul halaman beranda aplikasi dimana *user* dapat memilih Menu Tentang dan aplikasi akan menampilkan halaman tentang yang berisi tentang pengembang aplikasi, *button* untuk pemberian *rating* dan komentar di *Google Playstore*, dan *button* untuk mengirimkan *email* kepada pengembang aplikasi.

2. Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas sebuah sistem atau proses bisnis. Pada aplikasi atau sistem ini, terdapat empat *activity diagram*.

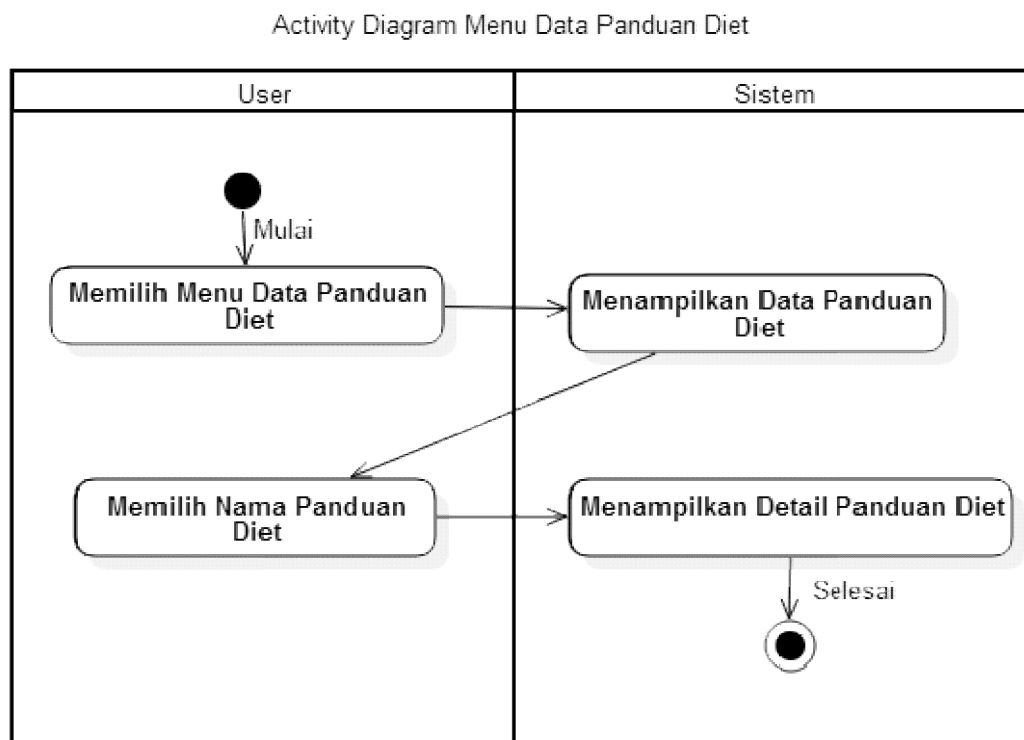
a. Activity Diagram Konsultasi



Gambar 14 Activity Diagram Konsultasi

Activity Diagram Konsultasi dimulai dengan *user* memilih Menu Konsultasi, kemudian sistem akan menampilkan pertanyaan gejala dan *user* menjawab pertanyaan tersebut. Sistem akan mencocokkan gejala yang dipilih dengan aturan yang ada lalu menampilkan hasil konsultasi.

b. *Activity Diagram* Data Panduan Diet

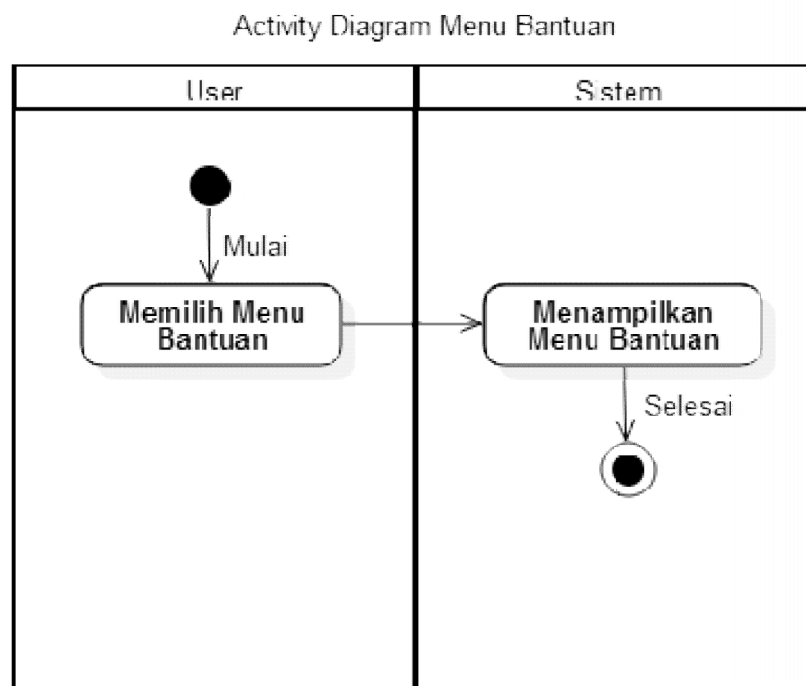


Gambar 15 *Activity Diagram* Data Panduan Diet

Activity Diagram Data Panduan Diet dimulai dengan *user* memilih Menu Data Panduan Diet, kemudian sistem akan menampilkan *listview* panduan diet. *User* dapat memilih nama panduan diet untuk melihat detailnya.

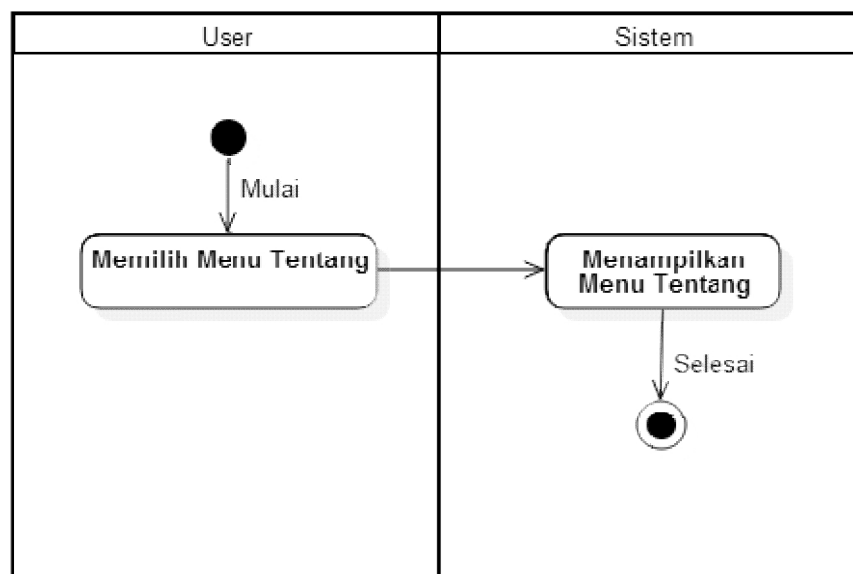
c. *Activity Diagram* Bantuan

Activity Diagram Bantuan dimulai dengan *user* memilih Menu Bantuan, kemudian sistem akan menampilkan informasi yang berkaitan dengan cara penggunaan aplikasi.



Gambar 16 *Activity Diagram* Bantuan

d. Activity Diagram Tentang

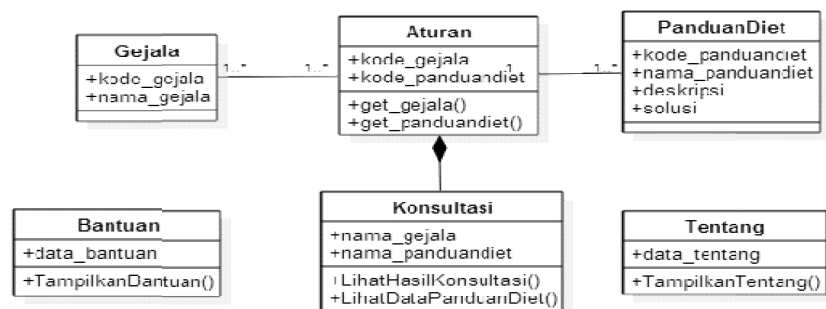


Gambar 17 *Activity Diagram* Tentang

Activity Diagram Tentang dimulai dengan *user* memilih Menu Tentang, kemudian sistem akan menampilkan informasi aplikasi dan pengembang.

3. Class Diagram

Class diagram merupakan diagram yang digunakan untuk menampilkan beberapa kelas serta paket pada sistem. Model *class diagram* sistem pada penelitian ini adalah sebagai berikut.



Gambar 18 *Class Diagram*

Pada Gambar 18 ditunjukkan bahwa *class diagram* Pakar Panduan Diet memiliki enam kelas sebagai berikut:

- Kelas Gejala, memiliki hubungan asosiasi dengan kelas aturan dengan nilai kardinalitas satu atau lebih. Hal ini berarti satu gejala dapat merupakan komponen pada satu aturan atau lebih. Begitu juga dengan kelas aturan yang dapat memiliki satu gejala atau lebih.
- Kelas Panduan Diet, memiliki hubungan asosiasi dengan kelas aturan dengan nilai kardinalitas hanya satu. Hal ini berarti setiap penyakit hanya memiliki satu aturan. Sedangkan aturan dapat memiliki satu penyakit atau lebih.
- Kelas Konsultasi, memiliki hubungan komposisi dengan kelas aturan. Hal ini berarti kelas konsultasi tidak dapat berdiri tanpa kelas aturan. Apabila kelas aturan hilang, maka kelas konsultasi juga akan hilang.

- Kelas Bantuan dan Tentang berdiri sendiri dan tidak ada hubungan dengan kelas lain.

4. Rancangan Antarmuka (*Interface*)

Antarmuka sistem merupakan proses penggambaran tampilan dari sebuah sistem sebagai penghujung antara sistem dengan *user*. Pada rancangan antarmuka ini, terdapat beberapa halaman yang dapat diakses *user* sebagai berikut.

a. *Layout Splash Screen*

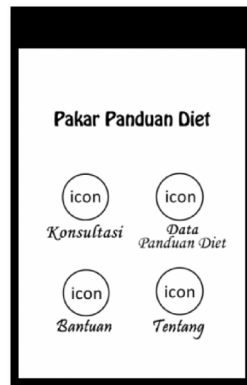
Layout Splash Screen merupakan layar kilat yang muncul ketika pertama kali membuka aplikasi. Rancangan *layout splash screen* disajikan pada Gambar 19.



Gambar 19 Rancangan *Layout Splash Screen*

b. *Layout Beranda (Menu Utama)*

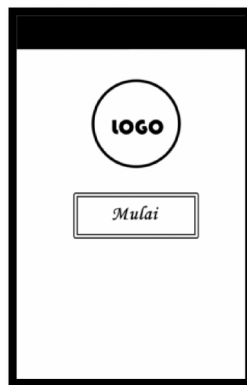
Beranda atau menu utama berisikan menu-menu pilihan (Menu Konsultasi, Menu Data Panduan Diet, Menu Bantuan, dan Menu Tentang) yang dapat digunakan oleh *user*. Rancangan *layout beranda* disajikan pada Gambar 20.



Gambar 20 Rancangan *Layout* Beranda

c. *Layout* Menu Konsultasi

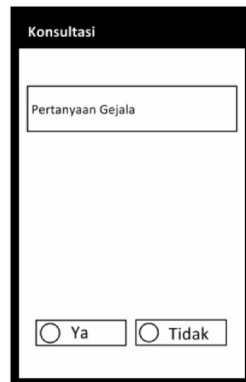
Layout Menu Mulai Konsultasi akan menampilkan submenu dengan *Button* Mulai. Rancangan *layout* konsultasi disajikan pada Gambar 21.



Gambar 21 Rancangan *Layout* Konsultasi

d. *Layout* Pertanyaan Gejala

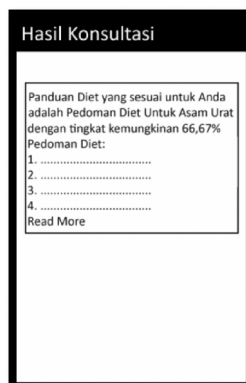
Layout Pertanyaan Gejala merupakan *sub-menu* dari Menu Konsultasi yang akan menampilkan pertanyaan-pertanyaan gejala yang harus dijawab oleh *user* dengan memilih *button* “YA” atau “TIDAK”. Rancangan *layout* pertanyaan gejala disajikan pada Gambar 22.



Gambar 22 Rancangan *Layout* Pertanyaan Gejala

e. *Layout* Hasil Konsultasi

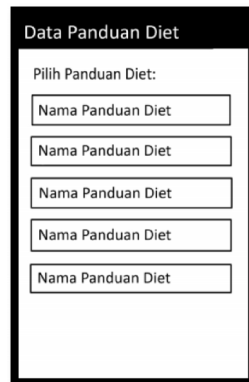
Layout Hasil Konsultasi merupakan *sub-menu* dari Menu Konsultasi yang akan menampilkan hasil panduan diet yang sesuai konsultasi gejala yang telah dilakukan. Rancangan *layout* hasil konsultasi disajikan pada Gambar 23.



Gambar 23 Rancangan *Layout* Hasil Diagnosis

f. *Layout* Data Panduan Diet

Layout Menu Data Panduan Diet dapat dilihat apabila *user* memilih menu ini. Halaman ini menampilkan informasi panduan diet. Rancangan *layout* data panduan diet disajikan pada Gambar 24.

A mobile application screen titled "Data Panduan Diet". Below the title, it says "Pilih Panduan Diet:". There are five identical input fields, each containing the placeholder text "Nama Panduan Diet".

Gambar 24 Rancangan *Layout* Data Panduan Diet

g. *Layout* Bantuan

Layout Bantuan menampilkan informasi bantuan dalam penggunaan aplikasi.

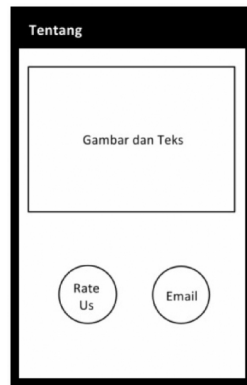
Rancangan *layout* bantuan disajikan pada Gambar 25.

A mobile application screen titled "Bantuan". It features a large rectangular box in the center with the placeholder text "Gambar dan Teks".

Gambar 25 Rancangan *Layout* Bantuan

h. *Layout* Tentang

Layout Tentang menampilkan informasi yang berkaitan dengan aplikasi dan pengembang aplikasi. Rancangan *layout* tentang disajikan pada Gambar 26.



Gambar 26 Rancangan *Layout* Tentang

3.3.3.3 Coding

Tahap *Coding* adalah proses menerjemahkan desain yang telah dirancang menjadi bahasa yang dimengerti oleh computer. Dalam penelitian ini, sistem atau aplikasi dibangun menggunakan bahasa pemrograman android yang terdiri dari bahasa *Java* dan XML menggunakan *framework Android Studio*.

3.3.3.4 Testing

Tahap *Testing* merupakan tahapan pengujian yang dilakukan setelah proses *coding* selesai. Dalam penelitian ini, pengujian yang dilakukan adalah pengujian internal dan pengujian eksternal.

A. Pengujian Internal

Pengujian internal adalah pengujian yang dilakukan oleh peneliti untuk menguji fungsional sistem serta kepakaran dari sistem dalam menentukan panduan diet berdasarkan fakta-fakta yang diberikan oleh *user*.

1. Pengujian Fungsional

Pengujian fungsional dilakukan dengan menggunakan pendekatan *black box* dengan metode *Equivalence Partitioning* (EP). Pengujian dengan metode ini tidak memperhatikan struktur logika internal atau *coding* perangkat lunak, namun

dengan membagi domain masukan dari program-program ke dalam kelas-kelas sehingga diperoleh *test case*. *Black box testing* dilakukan untuk memastikan apakah sistem yang dibuat telah bekerja sesuai ketentuan yang ditetapkan dan EP berusaha mendefinisikan kasus uji yang menemukan sejumlah kesalahan dan mengurangi jumlah kasus uji yang harus dibuat. Kelas uji pada pengujian ini adalah versi android, resolusi layar dan densitas layar, pengujian *user interface*, dan pengujian fungsi serta menu aplikasi.

a) Pengujian Versi Android

Pengujian versi android bertujuan untuk mengetahui apakah aplikasi dapat berjalan pada semua versi android yang diujikan. Pengujian ini menggunakan beberapa kriteria yang sudah ditentukan seperti kelas uji, daftar pengujian, skenario uji, dan hasil yang diharapkan. Pengujian versi android dapat dilihat pada Tabel 3.

Tabel 3 Daftar Pengujian Versi Android

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Versi Android	Pengujian Kompabilitas Versi OS Android	Pengujian pada Android Versi 4.3 (<i>Jelly Bean</i>)	Kompatibel dengan Android Versi 4.3 (<i>Jelly Bean</i>)
			Pengujian pada Android Versi 4.4 (<i>KitKat</i>)	Kompatibel dengan Android Versi 4.4 (<i>KitKat</i>)
			Pengujian pada Android Versi 5.0 (<i>Lollipop</i>)	Kompatibel dengan Android Versi 5.0 (<i>Lollipop</i>)
			Pengujian pada Android Versi 6.0 (<i>Marshmallow</i>)	Kompatibel dengan Android Versi 6.0 (<i>Marshmallow</i>)

b) Pengujian Resolusi Layar dan Densitas Layar

Pengujian resolusi layar dan densitas layar bertujuan untuk mengetahui apakah tampilan aplikasi dapat terlihat baik, proporsional, dan sesuai pada masing-masing resolusi android yang diujikan. Pengujian ini menggunakan beberapa kriteria yang sudah ditentukan seperti kelas uji, daftar pengujian, skenario uji, dan hasil yang diharapkan. Pengujian resolusi layar dan densitas layar dapat dilihat pada Tabel 4.

Tabel 4 Daftar Pengujian Resolusi Layar dan Densitas Layar

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Resolusi Layar dan Densitas Layar	Pengujian Resolusi Layar dan Densitas Layar	Pengujian pada android dengan resolusi 4 inch	Tampilan terlihat baik pada android dengan resolusi 4 inch
			Pengujian pada android dengan resolusi 4,5 inch	Tampilan terlihat baik pada android dengan resolusi 4,5 inch
			Pengujian pada android dengan resolusi 5 inch	Tampilan terlihat baik pada android dengan resolusi 5 inch
			Pengujian pada android dengan resolusi 5,5 inch	Tampilan terlihat baik pada android dengan resolusi 5,5 inch

c) Pengujian *User Interface*

Pengujian *user interface* bertujuan untuk mengetahui apakah tampilan sistem baik dan *user friendly* terhadap pengguna. Pengujian ini menggunakan beberapa

kriteria yang sudah ditentukan seperti kelas uji, daftar pengujian, skenario uji, dan hasil yang diharapkan. Pengujian *user interface* dapat dilihat pada Tabel 5.

Tabel 5 Daftar Pengujian *User Interface*

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	<i>User Interface</i>	Pengujian pada <i>icon</i> Aplikasi Pakar Panduan Diet	Klik <i>icon</i> Aplikasi Pakar Panduan Diet pada perangkat android	Menampilkan <i>splash screen</i> dan diikuti halaman utama aplikasi
			Klik tombol menu Konsultasi	Menampilkan <i>layout</i> menu Konsultasi
			Klik tombol menu Data Panduan Diet	Menampilkan <i>layout</i> menu Data Panduan Diet
			Klik tombol menu Bantuan	Menampilkan <i>layout</i> menu Bantuan
			Klik tombol menu Tentang	Menampilkan <i>layout</i> menu Tentang

d) Pengujian Fungsi Menu Aplikasi

Pengujian fungsi menu aplikasi bertujuan untuk mengetahui apakah fungsi yang diberikan pada masing-masing tombol menu aplikasi dapat berfungsi dengan baik. Pengujian ini menggunakan beberapa kriteria yang sudah ditentukan seperti kelas uji, daftar pengujian, skenario uji, dan hasil yang diharapkan. Pengujian fungsi menu aplikasi dapat dilihat pada Tabel 6.

Tabel 6 Daftar Pengujian Fungsi Menu Aplikasi

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Fungsi <i>layout</i> menu Konsultasi	Pengujian pada <i>layout</i> Konsultasi	Klik tombol menu Konsultasi	Menampilkan sub menu mulai
		Pengujian pada Menu Konsultasi	Klik tombol Mulai	Menampilkan <i>layout</i> pertanyaan yang disediakan sistem
			Klik tombol “Ya”	Menampilkan pertanyaan gejala selanjutnya atau hasil konsultasi
			Klik tombol “Tidak”	Menampilkan pertanyaan gejala selanjutnya atau hasil konsultasi
			Klik nama penyakit hasil konsultasi	Menampilkan detail panduan diet penyakit hasil konsultasi
2	Fungsi <i>layout</i> menu Data Panduan Diet	Pengujian pada <i>layout</i> Data Panduan Diet	Klik tombol Data Panduan Diet	Menampilkan <i>layout</i> data Panduan Diet berupa <i>listview</i>
		Pengujian melihat informasi data Panduan Diet	Pilih salah satu Panduan Diet	Menampilkan informasi Panduan Diet yang dipilih
3	Fungsi <i>layout</i> menu Bantuan	Pengujian melihat informasi Bantuan	Klik tombol Bantuan	Menampilkan <i>layout</i> bantuan
			<i>Swipe</i> pada layar	Menampilkan bantuan penggunaan aplikasi
4	Fungsi <i>layout</i> menu Tentang	Pengujian melihat informasi Tentang	Klik tombol <i>Rate Us</i>	Menampilkan aplikasi pada halaman <i>Playstore</i>
			Klik tombol <i>Email</i>	Menampilkan pilihan pengiriman <i>email</i>

2. Pengujian Kepakaran Sistem

Pengujian kepakaran sistem adalah pengujian yang bertujuan untuk melihat kemampuan sistem pakar pada aplikasi dalam mengidentifikasi jenis panduan diet berdasarkan fakta-fakta gejala yang diberikan. Pengujian kepakaran sistem dilakukan dengan membandingkan hasil diagnosis manual yang dilakukan oleh pakar dengan hasil konsultasi dengan sistem.

B. Pengujian Eksternal

Pengujian eksternal didasarkan pada pengujian yang melibatkan tampilan perangkat lunak (*user interface*), kinerja, dan kemudahan (*user friendly*). Pada pengujian ini, diberikan 5 (lima) preferensi jawaban dengan pilihan sebagai berikut: 5 = sangat setuju; 4 = setuju; 3 = cukup setuju; 2 = kurang setuju; 1 = tidak setuju. Daftar pertanyaan pengujian eksternal yang dilakukan disajikan pada Tabel 7.

Tabel 7 Daftar Pertanyaan Pengujian Non-Fungsional

No.	Pertanyaan	Kategori Penilaian				
		5	4	3	2	1
1	Apakah aplikasi dapat memberikan informasi yang bermanfaat bagi masyarakat?					
2	Apakah aplikasi mudah dipahami dan digunakan?					
3	Apakah aplikasi dan menu menu di dalamnya berjalan dengan baik dan sesuai fungsinya?					
4	Apakah aplikasi dapat membantu dalam identifikasi panduan diet dengan baik?					
5	Apakah identifikasi panduan diet sudah sesuai dengan fakta atau gejala yang ada?					
6	Apakah data-data yang tersedia sudah lengkap?					
7	Apakah aplikasi dapat memberikan solusi panduan diet berdasarkan gejala penyakit dalam yang dialami dengan baik?					

3.3.4 Analisis Hasil Pengujian

Tahap analisis hasil pengujian merupakan tahapan dimana hasil pengujian dianalisis. Tahap ini merupakan proses akhir dari penyelesaian sistem. Analisis yang dilakukan bertujuan untuk mengambil kesimpulan berdasarkan pengujian.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa:

1. Telah berhasil dibangun aplikasi “Pakar Panduan Diet” untuk membantu masyarakat umum dalam mengidentifikasi jenis panduan diet berdasarkan gejala-gejala yang ditemukan.
2. Sistem pakar yang dibangun dapat memberikan informasi yang bermanfaat bagi masyarakat berupa panduan diet untuk beberapa penyakit.
3. Hasil pengujian fungsional menunjukkan bahwa sistem pakar yang dibangun telah berjalan sesuai yang diharapkan.
4. Berdasarkan penilaian penggunaan aplikasi melalui pengisian kuisioner, dapat disimpulkan bahwa aplikasi “Pakar Panduan Diet” memperoleh presentase penilaian rata-rata sebesar 82,2% (sangat baik) menurut kategori responden I, memperoleh presentase penilaian rata-rata sebesar 80,6% (sangat baik) menurut kategori responden II, dan memperoleh presentase penilaian rata-rata sebesar 84,2% (sangat baik) menurut kategori responden III. Hasil rata-rata presentase penilaian dari ketiga kategori responden adalah 82,3%.

5.2 Saran

Beberapa saran untuk pengembangan lebih lanjut yang dapat diberikan setelah dilakukannya penelitian ini adalah sebagai berikut:

1. Penambahan data-data panduan diet untuk gejala-gejala penyakit lainnya.
2. Penyempurnaan desain *user interface* aplikasi.
3. Penyederhanaan bahasa pada gejala dan panduan diet agar lebih mudah dimengerti.
4. Pengembangan sistem untuk *platform* lainnya, seperti berbasis *web* atau berbasis *desktop*.

DAFTAR PUSTAKA

- [1] Andry Hartono, *Terapi Gizi dan Diet Rumah Sakit*, 2nd ed. Jakarta: Penerbit Buku Kedokteran EGC, 2006.
- [2] Pramono. (2010) Gizi Sehat. [Online].
<https://gizisehat.wordpress.com/2010/05/31/dukungan-nutrisi-pada-kasus-penyakit-dalam/>
- [3] Tutik Ernawati, *8 Jenis Panduan Diet dan 25 Data Gejala*. Bandar Lampung: Wawancara, Mei 2017.
- [4] Istri Sulistyowati and Arif Soeleman, "Sistem Pakar Untuk Mendiagnosa Penyakit Dalam Pada Manusia," *Jurnal Teknologi Informasi*, vol. 4, no. 1, April 2013.
- [5] Riki Andri Yusda and William Ramdhan, "Sistem Pakar Pengobatan Herbal," *Jurnal Teknologi dan Sistem Informasi*, vol. 1, no. 2, Maret 2015.
- [6] Anastasia Meyliana, Kusriani, and Emha Taufiq Luthfi, "Sistem Pakar Pada Konsultasi Jenis Senam Dengan Metode Forward Chaining," *Jurnal INFORMA Politeknik Indonusa Surakarta*, vol. 1, no. 3, 2016.
- [7] Ardhika Praseda Ageng Putra, Aristoteles, and Rara Diantari, "Sistem Pakar Diagnosa Penyakit pada Ikan Budidaya Air Tawar dengan Metode Forward Chaining Berbasis Android," *Jurnal Komputasi*, vol. 4, no. 1, pp. 92-98, 2016.
- [8] Anggun Marlina Puspitasari, Suhartono, and Kushartantya, "Sistem Pakar Berbasis Web Dengan Metode Probabilitas Klasik Untuk Diagnosa Penyakit Tuberkulosis Pada Manusia Dewasa," *Jurnal Masyarakat Informatika*, vol. 4, no. 7, 2013.
- [9] Lista Madeso, Don R. Kabo, and Johan R. Batmetan, "Rancang Bangun Sistem Pakar Penentuan Status Gizi pada Balita menggunakan Metode Forward Chaining," in *E-Jurnal Sariputra*, vol. 2, Tomohon, Juni 2015, pp. 16-24.

- [10] Sulis Triyanto and Abdul Fadlil, "Sistem Pakar Untuk Mendiagnosa Penyakit Kelinci," *Jurnal Sarjana Teknik Informatika*, vol. 2, no. 1, Februari 2014.
- [11] Grace Apriliany Presilia Kurmasela and Gisela Nina Sevani, "Aplikasi Diagnosa Gizi Buruk Pada Balita Menggunakan Metode Pembobotan (Studi Kasus: Kecamatan Nusaniwe, Kota Ambon)," *ComTech*, vol. 6, no. 1, Maret 2015.
- [12] Aristoteles , Kusuma Adhianto, and Puja Putri Abdullah, "Sistem Pakar Untuk Mendiagnosa Penyakit Pada Ayam Dengan Metode Certainty Factor Berbasis Android," *Jurnal Komputasi*, vol. 4, no. 1, 2016.
- [13] Jati Sasongko Wibowo, "Penerapan Sistem Pakar Dalam Bidang : Industri, Pendidikan, Bisnis," *Edisi Januari*, vol. VII, no. 1, Januari 2002.
- [14] Azis Sukma Dhiana, "Rancang Bangun Sistem Pakar Untuk Mendeteksi Gizi Buruk Pada Balita," Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah, Jakarta, 2010.
- [15] Kusrini , *Aplikasi Sistem Pakar Menentukan Faktor Kepastian Pengguna dengan Kualifikasi Pernyataan*, Dwi Prabantini, Ed. Yogyakarta: CV. ANDI OFFSET, 2008.
- [16] Anderias Eko Wijaya, "Sistem Pakar Berbasis Web Untuk Mendiagnosa Penyakit Pada Hewan Kurban Menggunakan Backward Chaining," *Jurnal Teknologi Informasi dan Komunikasi STMIK Subang*, 2013.
- [17] Agus Sasmito Aribowo and Siti Khomsah, "Sistem Pakar Dengan Beberapa Knowledge Base Menggunakan Probabilitas Bayes dan Mesin Inferensi Forward Chaining," in *Seminar Nasional Informatika (semnasIF)*, Yogyakarta, 2011, pp. 51-58.
- [18] Feriani Astuti Tarigan, "Sistem Pakar Untuk Penyusunan Jadwal Kuliah Berbasis Forward Chaining," *Jurnal TIME*, vol. II, no. 2, 2013.
- [19] Sunita Almitsier, *Prinsip Dasar Ilmu Gizi*. Jakarta: Gramedia Pustaka Utama, 2001.
- [20] Pusat Data dan Informasi Kementrian Kesehatan RI, *Hipertensi*. Jakarta, Mei 2014.

- [21] Bianti Nuraini, "Risk Factors of Hypertension," *J MAJORITY*, vol. 4, no. 5, Februari 2015.
- [22] Rona Sari Mahaji Putri, Hanum Agustin, and Wulansari, "Hubungan Pola Makan Dengan Timbulnya Gastritis Pada Pasien di Universitas Muhammadiyah Malang Medical Center (UMC)," *Jurnal Keperawatan*, vol. 1, no. 2, Juli 2010.
- [23] Perhimpunan Dokter Spesialis Penyakit Dalam Indonesia, *Buku Ajar Ilmu Penyakit Dalam*, VI ed. Jakarta: InternaPublishing, 2014, vol. II.
- [24] Badan Penelitian dan Pengembangan Kesehatan Kementerian Kesehatan RI, *RISKESDAS 2013*. Jakarta, Desember 2013.
- [25] Perkumpulan Endokrinologi Indonesia, *Konsensus Pengelolaan dan Pencegahan Diabetes Melitus Tipe 2 di Indonesia*, 4th ed. Jakarta: PB. PERKENI, 2011.
- [26] M. Rosidi Zamroni, Nizar Suryaman, and Ahmad Jalaluddin, "Rancang Bangun Aplikasi Permainan Untuk Pembelajaran Anak Menggunakan HTML 5," *Jurnal Teknika*, vol. 5, no. 2, September 2013.
- [27] Mia Rosmiati, "Analisis dan Perancangan E-Service Untuk Pelanggan Pada Jaya Bersama Konveksi," *IJSE - Indonesian Journal on Software Engineering*, vol. 1, no. 1, 2015.
- [28] Prastuti Sulistyorini, "Pemodelan Visual Dengan Menggunakan UML dan Rational Rose," *Jurnal Teknologi dan Informasi DINAMIK*, vol. XIV, no. 1, Januari 2009.
- [29] Zulkififli, "Model Prediksi Berbasis Neural Network Untuk Pengujian Perangkat Lunak Metode Black Box," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Yogyakarta, 2013, pp. 33-37.
- [30] M. Sidi Mustaqbal, Roeri Fajri Firdaus, and Hendra Rahmadi, "Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)," *JITTER - Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 1, no. 3, Agustus 2015.

- [31] Suhatati Tjandra and C. Pickerling, "Aplikasi Metode-Metode Software Testing Pada Configuration, Compatibility, dan Usability Perangkat Lunak," in *Seminar Nasional Inovasi Dalam Desain dan Teknologi - IDEaTech*, Surabaya, 2015, pp. 367-374.
- [32] Aristoteles , Rara Diantari, and Retno Monyka Agustianti, "Sistem Pakar Identifikasi Jenis Ikan Famili Cyprinidae Menggunakan Metode Forward Chaining dan Classic Probability," *Jurnal Komputasi*, vol. 4, no. 1, 2016.
- [33] Direktorat Bina Farmasi Komunitas dan Klinik Ditjen Bina Kefarmasian dan Alat Kesehatan Departemen Kesehatan, *Pharmaceutical Care Untuk Pasien Penyakit Arthritis Rematik*. Jakarta, 2006.