

**ANALISIS SISTEM PAKAR PERTUMBUHAN ANAK
STANDAR WHO BERBASIS ANDROID DENGAN
METODE *FORWARD CHAINING***

SKRIPSI



Oleh:
Jefri
130210366

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2017**

**ANALISIS SISTEM PAKAR PERTUMBUHAN ANAK
STANDAR WHO BERBASIS ANDROID DENGAN
METODE *FORWARD CHAINING***

SKRIPSI

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



**Oleh:
Jefri
130210366**

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2017**

PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, 11 Februari 2017

Yang membuat pernyataan,

Materai

J e f r i

NPM : 130210366

**ANALISIS SISTEM PAKAR PERTUMBUHAN ANAK
STANDAR WHO BERBASIS ANDROID DENGAN
METODE *FORWARD CHAINING***

**Oleh
Jefri
130210366**

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

**Telah disetujui oleh Pembimbing pada tanggal
seperti tertera di bawah ini**

Batam, 11 Februari 2017

**Very Karnadi, S.Kom., M.Kom
Pembimbing**

ABSTRAK

Sistem Pakar adalah salah satu bagian dari Kecerdasan Buatan yang mengandung pengetahuan dan pengalaman yang dimasukkan oleh banyak pakar ke dalam suatu area pengetahuan tertentu, sehingga setiap orang dapat menggunakannya untuk memecahkan berbagai masalah yang dihadapi. Pertumbuhan balita bisa terjadi berdasarkan beberapa faktor, yaitu berdasarkan kelahirannya dan pertumbuhan gizi yang dikonsumsi. Dengan memanfaatkan metode *Forward Chaining*, dapat dihasilkan suatu aplikasi untuk menguraikan penilaian tingkat pertumbuhan balita berdasarkan standar *WHO (World Health Organization) 2005*. Desain sistem dilakukan menggunakan bantuan aplikasi *starUML*. Aplikasi berbasis *android* yang dirancang dengan bahasa pemrograman *JAVA, PHP, dan database MySQL*. Metode *Forward Chaining* digunakan dengan cara menjabarkan jawaban dari pertama awal hingga pertanyaan terakhir untuk mendapatkan sebuah hasil. Aplikasi yang di rancang dapat digunakan menguraikan tingkat pertumbuhan anak dan menyimpulkan saran yang tepat.

Kata kunci: Sistem pakar, pertumbuhan balita, *forward chaining*, *Android*, **World Health Organization (WHO) 2005**

ABSTRACT

Expert System is one part of the Artificial Intelligence containing knowledge and experience which included many experts in a particular area of knowledge, so that everyone can use it to solve the various problems encountered. Infant growth could occur by several factors, namely by birth and growth of nutrients consumed. By utilizing Forward Chaining method, to produce an application for assessment outlines the growth rate of infants based on standard WHO (World Health Organization), 2005. System design is done using the application's called StarUML. Android based application is developed by using programming language JAVA, PHP, and MySQL database. Forward Chaining method used to describe the way the first answer from the beginning to the last question to get a result. Applications are designed can be used decipher the growth rate of children and concluded that the right advice.

Keywords : Expert System, Child Growth, forward chaining, Android, World Health Organization (WHO) 2005

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Sanghyang Adi Buddha Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam.
2. Ketua Program Studi Andi Maslan, S.T., M.SI.
3. Bapak Very Karnadi, S.Kom., M.Kom selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. Kedua orang tua penulis yang telah membesarkan penulis dan menjadikan penulis orang yang berbakti kepada agama dan negara.

6. Bapak Dr. Wennas, SpA, Dr. Nopri Esmiralda dan Dr. Zufahni selaku pakar pertumbuhan anak dalam penelitian ini
7. Seluruh teman-teman dan sahabat seperjuangan selama kuliah yang namanya tidak bisa disebutkan satu-persatu, terima kasih telah menjadikan masa kuliah selama ini terasa indah dan menyenangkan.

Semoga Sanghyang Adi Buddha Tuhan Yang Maha Esa membalas kebaikan dan selalu mencurahkan hidayah serta taufikNya, sadhu sadhu sadhu.

Batam, Februari 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMBUNG DEPAN	
HALAMAN JUDUL	
HALAMAN PERNYATAAN.....	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	
1.1 Latar Belakang Penelitian.....	1
1.2 Identifikasi Masalah	4
1.3 Pembatasan Masalah	4
1.4 Perumusan Masalah.....	5
1.5 Tujuan Penelitian	6
1.6 Manfaat Penelitian.....	6
BAB II TINJAUAN PUSTAKA	
2.1 Teori Dasar	7
2.1.1 Kecerdasan Buatan	7
2.1.2 Sistem Pakar.....	9
2.1.3 Forward Chaining.....	16
2.2 Variabel	18
2.2.1 Umur.....	18
2.2.2 <i>Length/height-for-age</i>	18
2.2.3 <i>Weight-for-age</i>	20
2.3 Software Pendukung.....	20
2.3.1 Java	20
2.3.2 Eclipse	22
2.3.3 Android.....	23
2.3.4 <i>XAMPP (Apache PHP)</i>	28
2.3.5 MySQL	30
2.3.5.1 <i>Entity Relationship Diagram</i>	34
2.3.6 <i>Unified Modeling Language (UML)</i>	35
2.3.6.1 <i>Class Diagram</i>	36
2.3.6.2 <i>Use Case Diagram</i>	38
2.3.6.3 <i>Activity Diagram</i>	41
2.3.6.4 <i>Sequence Diagram</i>	42

2.4	Penelitian Terdahulu.....	45
2.5	Kerangka Pemikiran	49

BAB III METODOLOGI PENELITIAN

3.1	Desain Penelitian.....	51
3.2	Teknik Pengumpulan Data	52
3.3	Operasional Variabel.....	53
3.4	Perancangan Sistem.....	59
3.4.1	UML (Unified Modelling Language).....	59
3.4.1.1	<i>Class Diagram</i>	59
3.4.1.2	<i>Use Case Diagram</i>	60
3.4.1.3	<i>Activity Diagram</i>	64
3.4.1.4	<i>Sequence Diagram</i>	69
3.4.2	Desain Basis Data	74
3.4.3	Prototype.....	77
3.5	Lokasi dan Jadwal Penelitian.....	82
3.5.1	Lokasi Penelitian.....	82
3.5.2	Jadwal Penelitian.....	82

BAB IV HASIL DAN PEMBAHASAN

4.1	Hasil Penelitian	84
4.2	Pembahasan	98
4.2.1	Pengujian Validasi.....	99

BAB V SIMPULAN DAN SARAN

5.1	Simpulan.....	101
5.2	Saran.....	101

DAFTAR PUSTAKA

RIWAYAT HIDUP

SURAT KETERANGAN PENELITIAN

LAMPIRAN

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol pada <i>Class diagram</i>	37
Tabel 2.2 Simbol pada <i>Use case diagram</i>	39
Tabel 2.3 Simbol pada <i>Activity diagram</i>	41
Tabel 2.4 Simbol pada <i>Sequence diagram</i>	43
Tabel 3.1 Operasional Variabel	53
Tabel 3.2 Kategori dan Ambang Batas Status Gizi Berdasarkan Variabel	54
Tabel 3.3 Tabel Gejala / Sifat	54
Tabel 3.4 Tabel Keputusan	55
Tabel 3.5 Deskripsi <i>Use Case</i> Pengguna	61
Tabel 3.6 Deskripsi <i>Use Case</i> Admin	63
Tabel 3.7 Rincian tabel <i>user</i>	75
Tabel 3.8 Rincian tabel umur	75
Tabel 3.9 Rincian tabel beratbadan	75
Tabel 3.10 Rincian tabel tinggi	76
Tabel 3.11 Rincian tabel saran	76
Tabel 3.12 Rincian tabel datapengguna	77
Tabel 3.13 Jadwal Penelitian	83
Tabel 4.1 Hasil Pengujian Validasi	99

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Komponen-komponen penting dalam sebuah sistem pakar	13
Gambar 2.2 Tampilan Utama Aplikasi Eclipse	23
Gambar 2.3 Logo Android Versi 5.1	27
Gambar 2.4 Logo XAMPP	28
Gambar 2.5 Logo PHP.....	29
Gambar 2.6 Logo Apache.....	29
Gambar 2.7 Tampilan Utama Aplikasi MySQL Workbench 6.3 CE	34
Gambar 2.8 Tampilan Utama Aplikasi StarUML	36
Gambar 2.9 Kerangka Pemikir	50
Gambar 3.1 Desain Penelitian	51
Gambar 3.2 Pohon keputusan.....	57
Gambar 3.3 Class diagram	59
Gambar 3.4 Use case diagram pengguna biasa	60
Gambar 3.5 Use case diagram admin	62
Gambar 3.6 Activity diagram daftar	64
Gambar 3.7 Activity diagram login	65
Gambar 3.8 Activity diagram penilaian tingkat pertumbuhan anak.....	66
Gambar 3.9 Activity diagram data pertumbuhan anak	67
Gambar 3.10 Activity diagram ganti basis data	68
Gambar 3.11 Sequence diagram daftar	69
Gambar 3.12 Sequence diagram login	70
Gambar 3.13 Sequence diagram penilaian tingkat pertumbuhan anak	71
Gambar 3.14 Sequence diagram data pertumbuhan anak	72
Gambar 3.15 Sequence diagram ganti basis data	73
Gambar 3.16 Rancangan database	74
Gambar 3.17 Tampilan halaman utama atau login.....	78
Gambar 3.18 Halaman pendaftaran	78
Gambar 3.19 Halaman home (pengguna).....	78
Gambar 3.20 Halaman home (admin)	79
Gambar 3.21 Halaman tanggal lahir	79
Gambar 3.22 Halaman jenis kelamin	79
Gambar 3.23 Halaman berat badan.....	80
Gambar 3.24 Halaman tinggi atau panjang	80
Gambar 3.25 Halaman data pengguna	80
Gambar 3.26 Halaman ganti basis data.....	81
Gambar 3.27 Halaman ganti data berat.....	81
Gambar 4.1 Halaman utama atau login.....	84
Gambar 4.2 Halaman register	85
Gambar 4.3 Halaman tentang kami	86
Gambar 4.4 Halaman home – pengguna biasa	87

Gambar 4.5 Halaman <i>home</i> – admin	88
Gambar 4.6 Halaman tanggal lahir	89
Gambar 4.7 Halaman jenis kelamin	90
Gambar 4.8 Halaman berat badan	91
Gambar 4.9 Halaman tinggi	92
Gambar 4.10 Halaman hasil berat berdasarkan umur	93
Gambar 4.11 Halaman hasil tinggi berdasarkan umur	94
Gambar 4.12 Halaman data pengguna	95
Gambar 4.13 Halaman ganti basis data	96
Gambar 4.14 Halaman ganti data berat	97
Gambar 4.15 Halaman ganti data tinggi	98

BAB I

PENDAHULUAN

1.1 Latar Belakang Penelitian

Perkembang teknologi membuat semua hal yang dilakukan oleh manusia bagaikan bisa dilakukan melalui teknologi. Seperti pembelian pakaian maupun makanan, hanya membutuhkan pesanan *online*, benda yang diinginkan akan didapatkan. Sama seperti media konsultasi yang memungkinkan seseorang untuk mencari bantuan secara online untuk mendapatkan solusi yang diinginkan. Media konsultasi merupakan sebuah media yang diakses secara *online* untuk berkomunikasi dengan seorang pakar. Akan tetapi, media konsultasi ini memiliki sebuah persoalan yakni seorang pakar tidak bisa selalu dalam kondisi '*online*' untuk memberikan solusi. Hal ini akan menyebabkan waktu pengguna untuk mendapatkan solusi menjadi panjang. Maka, terciptanya Sistem Pakar yang bisa bertindak seperti pakar dalam penyelesaian masalah maupun pemberian solusi.

Anak adalah penerus bangsa masa depan kita. Masa depan anak – anak tidak hanya tergantung pada etika dan pendidikan. Kesehatan juga merupakan salah satu faktor penting. Kesehatan disini berarti tahap pertumbuhan anak masa balita. Maka, tumbuh kembang harus menjadi perhatian penting bagi orang tua. Proses tumbuh kembang anak dapat berlangsung secara alamiah, tetapi proses tersebut sangat tergantung kepada orang tua atau penjaga. Periode penting dalam tumbuh

kembang anak adalah masa balita. Karena pada masa ini pertumbuhan dasar akan mempengaruhi dan menentukan perkembangan anak selanjutnya.

Pertumbuhan anak yang berada pada gizi normal atau dikatakan ideal berdasarkan standar WHO 2005 akan terhindar dari kelainan-kelainan pada balita. Seperti tulang kaki yang membengkok karena pertumbuhan berat badan melebihi pertumbuhan tinggi badan. Mengakibatkan kaki tulang tidak mampu mendukung berat badan. Kedua, diabetes pada anak yang dinamakan *baby sugar* yang diakibatkan orang tua menginginkan anaknya gemuk, sehingga berat badan melebihi batas normal. Hal-hal itu bisa di hindari dengan menggunakan aplikasi sistem pakar ini.

Penelitian Shaid, dkk. (2015:37) menyatakan “Pertumbuhan balita bisa terjadi berdasarkan beberapa factor, yaitu berdasarkan kelahirannya dan pertumbuhan gizi yang dikonsumsi. Dengan memanfaatkan metode Case-base Reasoning, dapat dihasilkan suatu aplikasi untuk mengidentifikasi pertumbuhan balita. Dengan harapan sistem ini nantinya dapat digunakan sebagai sarana atau sebagai pengetahuan dalam menjaga kestabilan pertumbuhan balita dan membantu anda untuk mendapatkan hasil yang optimal dalam menjaga pertumbuhan setiap balita. Metode *Case Based Reasoning* (CBR) digunakan dalam aplikasi Pertumbuhan Balita dengan menggunakan Perhitungan Nearest Neighbor, Dimana data kasus baru akan dibandingkan perhitungannya dengan data kasus lama yang ada di *database*, dan kemudian dihitung kriteria kemiripannya berdasarkan rumus atau ketentuan yang berlaku.” Dimana Shaid, dkk. (2015:44) menyarankan “Program ini dapat lebih disempurnakan, khususnya

pada bagian proses pencarian data kasus. Hal ini akan memberikan hasil konsultasi yang diberikan perangkat lunak lebih baik.”

Penelitian Ernawati, dkk. (2014:109) menyatakan “Hasil penelitian menunjukkan sebanyak 9,5 persen bayi dengan berat badan lahir rendah (BBLR) dan 22 persen stunting. Nilai z-skor panjang badan terhadap umur pada bayi baru lahir berkorelasi dengan perkembangan motorik dan sosial emosi sejak bayi berumur nol bulan, yaitu $\rho=0,33$; $p=0,004$ untuk motorik dan $\rho=0,244$ dengan $p=0,036$ untuk sosial emosi.”

Penelitian Zamroni, dkk. (2013:510) menyatakan “Sistem pakar yang dibuat dapat membantu masyarakat yaitu orang awam dalam mendiagnosa awal perkembangan pada anak tersebut sehingga mempermudah dalam penanganannya.” Dikarenakan hasil penelitian menunjukkan (Zamroni, dkk., 2013:507) “Banyaknya bayi yang lahir pada saat ini. Tetapi dengan bertambahnya tersebut masih ada saja orang tua yang belum faham atau kurang mengertinya tentang perkembangan anak secara baik dan sesuai dengan usia yang benar”. Serta, menyarankan (Zamroni, dkk., 2013:510) “Perlu diadakan penambahan data untuk jenis perkembangan usia selanjutnya sehingga informasi yang dimiliki akan semakin luas dan banyak.”

Berdasarkan latar belakang permasalahan di atas, penulis tertarik untuk melakukan penelitian terhadap permasalahan berkaitan dengan pertumbuhan balita untuk mengingatkan masyarakat akan pengawasannya. Oleh karena itu, maka penulis bermaksud untuk merancang sebuah aplikasi sistem pakar yang mampu digunakan sebagai sebuah indikator pada pertumbuhan balita dan

memberikan sebuah saran. Tugas Akhir berjudul “**Analisis Sistem Pakar Pertumbuhan Balita Standar WHO berbasis Android dengan Metode *Forward Chaining***”.

1.2 Identifikasi Masalah

Setelah mempelajari dan mengamati topik yang diangkat dan sistem yang akan dibangun maka ditemukan beberapa permasalahan dalam analisis dan perencanaan pengembangan sistem yang akan dibangun yaitu:

1. Orang tua belum mengetahui tingkat pertumbuhan anaknya tanpa perbandingan yang jelas dan benar.
2. Belum adanya suatu sistem yang membantu masyarakat untuk mengetahui tingkat pertumbuhan anak.
3. Sulitnya berkonsultasi secara tatap muka dengan dokter spesialis anak mengenai tingkat pertumbuhan anak.

1.3 Pembatasan Masalah

Sesuai dengan permasalahan yang diangkat pada latar belakang penulisan, maka masalah yang akan dibahas dapat dirumuskan sebagai berikut:

1. Tingkat Pertumbuhan anak dinilai melalui umur yang berkisar dari 0 sampai 1 tahun, *Length/height-for-age* (tinggi anak berdasarkan usia) dan *weight-for-age* (berat anak berdasarkan usia berdasarkan Standar WHO tahun 2005).

2. Aplikasi ini digunakan untuk mengetahui tingkat pertumbuhan anak yang tidak mengalami kelainan hormon maupun genetik.
3. Aplikasi ini berbasis Android dengan menggunakan bahasa pemrograman Java.
4. Penanganan *database* akan menggunakan *database MySQL*.
5. Pembangunan sistem pakar menggunakan metode *Forward Chaining*.
6. Pengguna aplikasi harus memiliki *smartphone* berbasis android versi 5.0 *Lollipop*.

1.4 Perumusan Masalah

Berdasarkan batasan masalah yang dirumuskan, maka batasan masalah yang akan dibahas sebagai berikut:

1. Bagaimana merancang dan membangun sistem pakar yang bisa digunakan untuk penilaian tingkat pertumbuhan anak?
2. Bagaimana sistem pakar bisa menyimpulkan saran yang tepat terhadap pertumbuhan anak?
3. Bagaimana mengimplementasi sistem pakar dengan menggunakan bahasa pemrograman berbasis Android yang mencegah kesulitan konsultasi dengan pakar?

1.5 Tujuan Penelitian

Berdasarkan uraian latar belakang masalah, maka tujuan penyusunan Tugas Akhir ini adalah:

1. Menghasilkan sistem pakar yang bisa digunakan untuk menguraikan penilaian tingkat pertumbuhan anak.
2. Sistem pakar ini bisa menyimpulkan saran yang tepat terhadap pertumbuhan anak.
3. Dengan sebuah sistem pakar berbasis android, maka mencegah akan kesulitan berkonsultasi dengan pakar.

1.6 Manfaat Penelitian

Adapun manfaat dari penyusunan Tugas Akhir ini adalah sebagai berikut:

1. Secara Teoritis

Penelitian ini diharapkan dapat menambah pengetahuan bagi pembaca tentang sistem pakar pertumbuhan anak berstandar World Health Organization (*WHO*) tahun 2005.

2. Secara Praktis

Penelitian ini diharapkan dapat menjadi bantuan terhadap masyarakat untuk mengetahui pertumbuhan anak berstandar World Health Organization (*WHO*) tahun 2005.

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Menurut Sudaryono (2015:13) teori adalah generalisasi atau kumpulan generalisasi yang dapat digunakan untuk menjelaskan berbagai fenomena secara sistematis. Generalisasi adalah proses penalaran yang bertolak dari fenomena individual menuju kesimpulan umum.

2.1.1 Kecerdasan Buatan

Menurut Sutojo, dkk. (2011:1) kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

Herbert Alexander Simon (June 15, 1916-February 9, 2001): “Kecerdasan buatan (*Artificial Intelligence*) merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas” (Sutojo, dkk., 2011:2).

Bidang - bidang yang menggunakan kecerdasan buatan antara lain sebagai berikut.

1. Jaringan Saraf Tiruan (JST)

Menurut Sutojo, dkk. (2011: 396), jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis,

seperti proses informasi pada otak manusia. Cara kerja JST seperti cara kerja manusia, yaitu belajar melalui contoh.

2. Sistem Pakar (*Expert System*)

Menurut Sutojo, dkk. (2011: 204) sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia di mana pengetahuan tersebut dimasukkan ke dalam sebuah computer dan kemudian digunakan untuk menyelesaikan masalah – masalah yang biasanya membutuhkan kepakaran. Beberapa metode dalam sistem pakar adalah sebagai berikut.

a. *Forward Chaining*

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*.

b. *Backward Chaining*

Backward Chaining adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada dibagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian *IF*.

3. Logika *Fuzzy* (*Fuzzy Logic*)

Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol (Sutojo, dkk., 2011: 276).

Menurut Sutojo, dkk. (2011: 233) terdapat beberapa metode dalam logika fuzzy antara lain:

a. Metode *Tsukamoto*

Secara umum bentuk model *fuzzy Tsukamoto* adalah:

IF (X IS A) and (Y IS B) THEN (Z IS C)

Di mana A, B dan C adalah himpunan *fuzzy*.

b. Metode *Mamdani*

Metode Mamdani paling sering digunakan dalam aplikasi-aplikasi karena strukturnya yang sederhana, yaitu menggunakan operasi *MIN-MAX* atau *MAX-PRODUCT*.

c. Metode *Sugeno*

Dalam metode *Sugeno*, output sistem berupa konstanta atau persamaan linear. Secara umum bentuk model *fuzzy Sugeno* adalah:

IF (X₁ IS A₁) [] [] • • (X_N IS A_N) THEN Z = f(x,y)

2.1.2 Sistem Pakar

Menurut Sutojo, dkk. (2011:159) Sistem pakar merupakan cabang dari *Artificial Intelligence* (AI) yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver* (GPS) yang dikembangkan oleh Newell dan Simon. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti MYCIN untuk diagnosis penyakit, DENDRAL untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON dan XSEL untuk membantu konfigurasi

sistem komputer besar, SOPHIE untuk analisis sirkuit elektronik, Prospector digunakan di bidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik diesel dan sebagainya.

Menurut Hayadi (2016:1) Sistem pakar atau *Expert System* biasa disebut juga dengan *Knowledge Based System* yaitu suatu aplikasi komputer yang ditujukan untuk membantu pengambilan keputusan atau pemecahan persoalan dalam bidang yang spesifik. Sistem ini bekerja dengan menggunakan pengetahuan dan metode analisis yang telah didefinisikan terlebih dahulu oleh pakar yang sesuai dengan bidang keahliannya.

Menurut Jackson (1999, p3) “Sistem pakar adalah program komputer yang merepresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memberikan saran” (Sutojo, dkk., 2011:160).

Konsep dasar sistem pakar meliputi enam hal berikut (Sutojo, dkk., 2011:163).

1. Kepakaran (*Expertise*)

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

- a. Fakta-fakta tentang bidang permasalahan tertentu
- b. Teori-teori tentang bidang permasalahan tertentu

- c. Aturan *heuristic* yang harus dikerjakan dalam suatu situasi tertentu
- d. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya
- e. Strategi global untuk memecahkan permasalahan
- f. Pengetahuan tentang pengetahuan (*meta knowledge*).

2. Pakar (*Expert*)

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

- a. Mengenali dan memformulasikan masalah.
- b. Memecahkan masalah dengan cepat dan tegas.
- c. Menerangkan pemecahannya.
- d. Belajar dari pengalaman.
- e. Merekstrukturasikan pengetahuan.
- f. Memecahkan aturan-aturan.
- g. Menentukan relevansi

3. Pemindahan Kepakaran (*Transferring Expertise*)

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar kedalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar.

Proses ini melibatkan empat kegiatan, yaitu:

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

4. Inferensi (*Inferencing*)

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

5. Aturan-aturan (*Rule*)

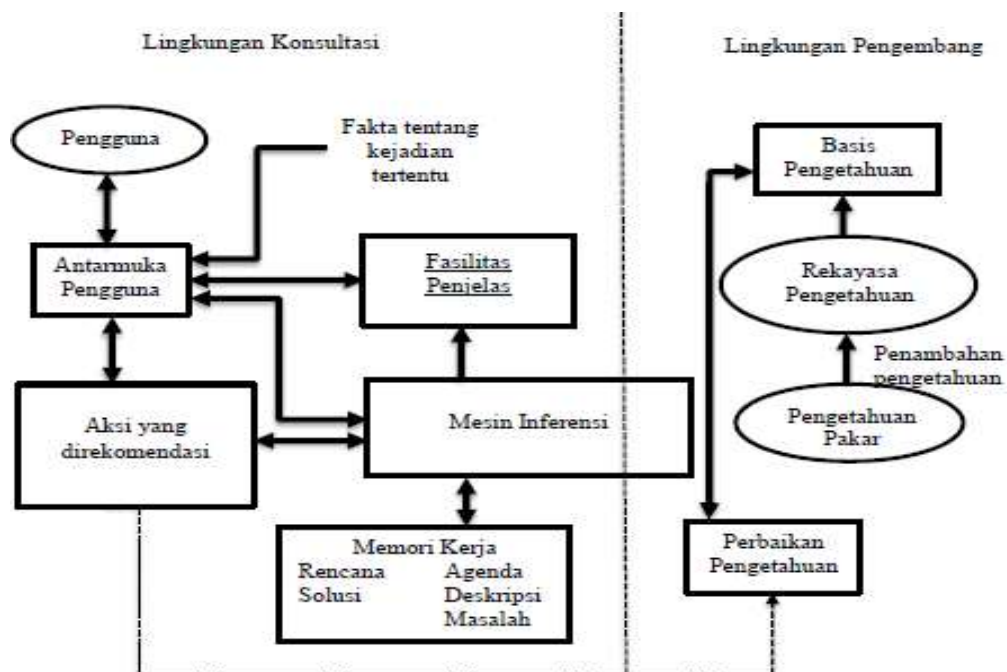
Kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis *rule* (*rule based systems*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

6. Kemampuan menjelaskan (*Explanation Capability*)

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikan oleh sistem pakar. Penjelasan dilakukan

dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi operasinya.

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memasukkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan sistem pakar (Sutojo, dkk., 2011:166). Gambar berikut menunjukkan komponen-komponen yang penting dalam sebuah sistem pakar:



Gambar 2.1 Komponen-komponen penting dalam sebuah sistem pakar
(Sumber: Sutojo, dkk., 2011:167)

Keterangan :

1. Akuisisi pengetahuan: digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku, dokumen multimedia, basis data, laporan riset khusus, dan informasi yang terdapat di *web*.
2. Basis pengetahuan (*knowledge base*): mengandung pengetahuan yang diperlukan untuk memahami, merumuskan dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:
 - a. Fakta, misalnya situasi, kondisi atau permasalahan yang ada.
 - b. Aturan (*rule*), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.
3. Mesin Inferensi (*Inference Engine*): adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan proses penalaran. Ada tiga teknik pengendalian yang digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik tersebut.

4. Daerah kerja (*blackboard*): untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi, sistem pakar membutuhkan *blackboard*, yaitu area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard* yaitu:
 - a. Rencana : bagaimana menghadapi masalah
 - b. Agenda : aksi-aksi potensial yang sedang menunggu untuk dieksekusi
 - c. Solusi : calon aksi yang akan dibangkitkan
5. Antarmuka pemakai (*user interface*): digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.
6. Sistem perbaikan pengetahuan (*knowledge refining system*): Subsistem penjelas berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.
7. Sistem perbaikan pengetahuan (*knowledge refining system*): Kemampuan memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat digunakan di masa mendatang. Kemampuan evaluasi diri seperti itu diperlukan oleh program

agar dapat menganalisis alasan-alasan kesuksesan dan kegagalannya dalam mengambil kesimpulan. Cara ini dapat menghasilkan basis pengetahuan yang lebih baik dan penalaran yang lebih efektif.

8. Pengguna (*User*): pada umumnya pengguna sistem pakar bukanlah seorang pakar yang membutuhkan solusi, saran, atau pelatihan (*training*) dan berbagai permasalahan yang ada.

2.1.3 Forward Chaining

Menurut Sutojo, dkk. (2011:171) Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database. Setiap kali pencocokan, dimulai dari rule teratas. Setiap rule hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi.

Menurut Russel S, Norvig P, 2003 Metode Forward Chaining adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan (Hayadi 2016:9).

Menurut Hayadi (2016:11) langkah-langkah yang harus dilakukan dalam membuat sistem forward *chaining* berbasis aturan, yaitu:

1. Pendefinisian Masalah

Tahap ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.

2. Pendefinisian Data Input

Sistem *forward chaining* memerlukan data awal untuk melakukan inferensi.

3. Pendefinisian Struktur Pengendalian Data

Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.

4. Penulisan Kode Awal

Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur aturan yang baik.

5. Pengujian Sistem

Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.

6. Perancangan Antarmuka

Antarmuka adalah suatu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.

7. Pengembangan Sistem

Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan prototipe sistem.

8. Evaluasi Sistem

Pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

2.2 Variabel

Menurut Sugiyono (2012:38) variabel penelitian adalah suatu atribut atau sifat atau nilai dari orang, obyek atau kegiatan yang mempunyai variasi tertentu yang diterapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya.

Menurut Sudaryono (2015:16) variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi dan kesimpulannya.

Berdasarkan Keputusan Menteri Kesehatan Republik Indonesia Nomor : 1995/Menkes/SK/XII/2010 tentang Standar Antropometri Penilaian Status Gizi Anak, indeks yang digunakan sebagai penentu kategori status gizi sebagai berikut:

2.2.1 Umur

Umur dihitung dalam bulan penuh. Contoh : umur 2 bulan 29 hari dihitung sebagai umur 2 bulan (Kemenkes, 2010:3).

2.2.2 *Length/height-for-age*

Ukuran panjang badan digunakan untuk anak umur 0 sampai 24 bulan yang diukur telentang. Bila anak umur 0 sampai 24 bulan diukur berdiri, maka hasil pengukurannya dikoreksi dengan menambahkan 0.7 cm (Kemenkes, 2010:3).

Standar untuk pertumbuhan linear memiliki bagian berdasarkan panjang (panjang-untuk-usia, 0 sampai 24 bulan) dan satu lagi di ketinggian (tinggi-untuk-

umur, 2 sampai 5 tahun). Dua bagian dibangun menggunakan model yang sama tetapi kurva akhir mencerminkan perbedaan rata-rata antara panjang berbaring dan berdiri tinggi. Dengan desain, anak-anak antara 18 dan 30 bulan dalam komponen *cross-sectional* dari MGRS (*Multicentre Growth Reference Study*) mengambil pengukuran panjang dan tinggi. Rata-rata perbedaan antara dua pengukuran pada 1625 anak-anak adalah 0,73 cm. Untuk cocok dengan model tunggal untuk rentang usia utuh, 0,7 cm itu ditambahkan ke nilai ketinggian penampang sebelum penggabungan dengan data panjang. Setelah model disesuaikan, kurva median dialihkan kembali ke bawah 0,7 cm untuk usia di atas dua tahun, dan koefisien kurva variasi disesuaikan dengan nilai-nilai median baru untuk membangun kurva pertumbuhan tinggi-untuk-usia. Daya transformasi pada usia diterapkan untuk meregangkan skala usia untuk masing-masing jenis kelamin sebelum disesuaikan dengan splines kubik untuk menghasilkan kurva pertumbuhan masing-masing. Kurva anak laki-laki diperlukan model dengan tingkat kebebasan yang lebih untuk menyesuaikan kedua median dan koefisien kurva variasi. Data untuk kedua jenis kelamin mengikuti distribusi normal (Onis, *et al.*, 2008:xviii).

Pengukuran tinggi anak dibagi menjadi dua, yaitu Panjang Badan (PB) dan Tinggi Badan (TB). Panjang Badan digunakan untuk anak umur 0 sampai 24 bulan yang diukur telentang. Bila anak umur 0 sampai 24 bulan diukur berdiri, maka hasil pengukurannya dikoreksi dengan menambahkan 0,7 cm. Ukuran Tinggi badan (TB) digunakan untuk anak umur di atas 24 bulan yang diukur berdiri. Bila anak umur 24 bulan diukur telentang, maka hasil pengukurannya dikoreksi dengan mengurangi 0,7 cm (Kemenkes, 2010:3).

2.2.3 *Weight-for-age*

Bobot dari sampel longitudinal dan *cross-sectional* digabung tanpa penyesuaian dan model tunggal disesuaikan untuk menghasilkan satu set kurva yang berdasarkan jenis kelamin masing-masing terhadap standar berat-untuk-usia. Daya transformasi yang sama diterapkan pada usia anak laki-laki dan perempuan sebelum disesuaikan dengan model konstruksi kurva. Data berat badan untuk kedua jenis kelamin yang miring, sehingga dalam menentukan model, parameter yang terkait dengan kemiringan dipasang di samping median dan koefisien perkiraan variasi. Dalam pemodelan kemiringan kurva gadis-gadis diperlukan tingkat kebebasan untuk menyesuaikan kurva untuk parameter ini (Onis, *et al.*, 2008:xviii).

2.3 Software Pendukung

2.3.1 Java

Java, dalam ilmu computer, merupakan bahasa pemrograman berorientasi objek yang diperkenalkan pada tahun 1995 oleh Sun Microsystems, Inc., yang saat Java diciptakan, dipimpin oleh James Gosling. Java memfasilitasi penyebaran baik data maupun program aplikasi kecil, yang dinamakan *applet*, lewat internet. Aplikasi-aplikasi Java tidak berinteraksi dengan CPU (*Central Processing Unit*) atau sistem operasi computer yang digunakan sehingga ia bisa bersifat mandiri

terhadap platform komputer, baik platform perangkat lunak maupun perangkat keras (Nugroho, 2008:4).

Keunggulan dari Java adalah dapat berjalan di banyak platform perangkat keras dan perangkat lunak sehingga pengembang aplikasi dan pemrogram dapat menuliskan program dengan Java (sekali dan hanya sekali saja) kemudian dengan relative mudah bisa menjalankannya di mana saja; di platform sistem operasi apapun serta di kebanyakan computer yang ada saat ini. Istilah populernya adalah “*Write Once, Run Anywhere*” (Nugroho, 2008:4).

Kepopuleran bahasa pemrograman Java juga dapat dimengerti karena Java memiliki bentuk dan sifat yang dimiliki bahasa lain yang terkemuka, seperti C, C++, Objective-C, Smalltalk serta Common Lisp. Dari Smalltalk, Java meminjam konsep model ekstensibilitas saat eksekusi, manajemen memori dinamis serta pengekseskuan program secara bersamaan (*multi-threading*). Dari C/C++, Java meminjam konsep-konsep operator yang sama sehingga mempermudah pemrogram C/C++ beralih ke Java (Nugroho, 2008:5).

Dalam bahasa pemrograman Java, kelas serta metode merupakan blok bangunan dasar. Jadi dengan bahasa pemrograman Java kita dapat menerapkan konsep pemrograman berorientasi objek dengan tuntas. Apalagi Java dilengkapi dengan pustaka kelas (*library*) yang beragam. Dalam Java, alokasi memori dinamis diatur secara otomatis oleh bagian kompiler Java yang dinamakan JVM (*Java Virtual Machine*) sehingga pemrograman agak dimudahkan. Keuntungan lain, karena ukuran JVM ini relative kecil maka kita tidak akan terlalu dipusingkan oleh batasan memori komputer yang kita miliki (Nugroho, 2008:6-7).

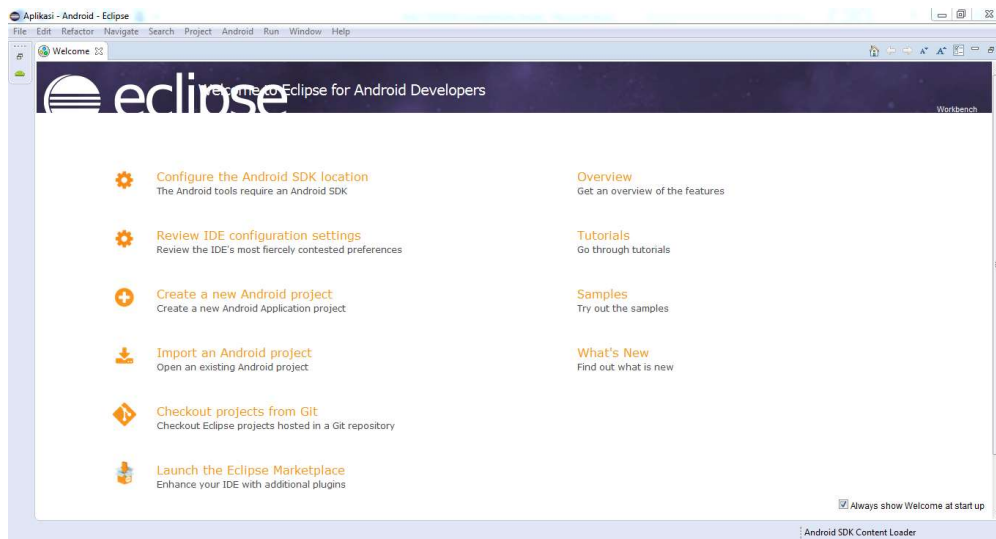
2.3.2 Eclipse

EMS (2015:35) mengatakana untuk IDE (*Integrated Development Environment*) yang dipakai dengan pemrograman Android, baik berbasis Java atau pemaketan dengan HTML dengan Cordova, Anda bisa menggunakan Eclipse. Sebenarnya *tool* yang bisa dipakai tidak hanya Eclipse, tapi ada juga *tool* lainnya seperti Android Studio. Tapi yang paling terkenal dan mudah konfigurasinya adalah Eclipse.

Menurut Safaat H (2015:4-5) Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

1. **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. **Multi-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.



Gambar 2.2 Tampilan Utama Aplikasi Eclipse
(Sumber: Data Penelitian, 2016)

2.3.3 Android

Menurut EMS (2015:1) Android secara sederhana bisa diartikan sebagai sebuah software yang digunakan pada perangkat mobile yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh Google. Sehingga Android mencakup keseluruhan sebuah aplikasi, mulai dari sistem operasi sampai pada pengembangan aplikasi itu sendiri. Pengembangan aplikasi pada platform

Android ini menggunakan dasar bahasa pemrograman Java. Tapi secara sempit, Android biasanya mengacu pada sistem operasinya saja .

Platform pengembangan aplikasi Android ini bersifat *open-source* atau terbuka, sehingga Anda dapat mengembangkan kemampuan untuk membangun aplikasi yang kaya dan inovatif. Bahkan seorang pengembang (*developer*) Android dapat membuat aplikasi yang bervariasi (EMS, 2015:2).

Android Lollipop (Hilmi Masruri & Creativity, 2015:20) adalah versi stabil terbaru dari sistem operasi Android yang dikembangkan oleh Google, yang pada saat ini mencakup versi antara 5.0 dan 5.1. Fitur yang ditawarkan Android Lollipop 5.0, antara lain :

1. Desain Material

Desain antarmuka yang berani, penuh warna, dan responsif untuk pengalaman yang konsisten dan intuitif di semua perangkat Anda. Responsif, gerak alami, pencahayaan dan bayangan realistis, dan elemen visual yang sudah familier memudahkan Anda untuk menavigasi perangkat. Warna baru yang terang, tipografi, dan gambar dari tepi ke tepi membantu memfokuskan perhatian Anda.

2. Notifikasi

Cara baru untuk mengontrol waktu dan cara menerima pesan, hanya pada saat yang Anda inginkan. Lihat dan balas pesan langsung dari layar kunci. Anda juga dapat menyembunyikan konten sensitif untuk notifikasi tersebut. Untuk meminimalkan gangguan, aktifkan mode Prioritas melalui tombol volume perangkat agar hanya orang dan notifikasi tertentu yang dapat masuk. Atau

jadwalkan masa tidak aktif secara berulang, misalnya dari jam 10 malam hingga 8 pagi, agar notifikasi Prioritas saja yang dapat masuk. Dengan Lollipop, panggilan telepon masuk tidak akan mengganggu film yang sedang Anda tonton atau game yang Anda mainkan. Anda dapat memilih untuk menjawab panggilan atau terus melakukan aktivitas. Kontrol notifikasi aplikasi, sembunyikan konten sensitif, dan prioritaskan atau nonaktifkan notifikasi aplikasi. Peringkat notifikasi yang lebih cerdas berdasarkan pengirimnya dan jenis komunikasi. Lihat semua notifikasi di satu tempat dengan mengetuk bagian atas layar.

3. Baterai

Daya untuk aktivitas sepanjang hari. Fitur penghemat baterai yang memperpanjang masa pakai perangkat hingga 90 menit. Perkiraan sisa waktu sampai baterai terisi penuh ditampilkan saat perangkat dicolokkan. Perkiraan sisa waktu sebelum mengisi daya perangkat kini dapat dilihat di setelan baterai.

4. Keamanan

Jaga agar file Anda tetap aman. Penerapan SELinux pada semua aplikasi untuk perlindungan yang lebih baik terhadap kerentanan dan perangkat lunak perusak. Gunakan Android Smart Lock untuk mengamankan ponsel atau tablet dengan menyandingkannya dengan perangkat tepercaya seperti Android Wear atau bahkan mobil.

5. Berbagi Perangkat

Berbagi dengan keluarga dan teman jadi lebih fleksibel. Fitur banyak pengguna untuk ponsel. Jika kelupaan ponsel, Anda tetap dapat menelepon teman

(atau mengakses pesan, foto, dll) dengan masuk ke ponsel Android lain yang menjalankan Lollipop. Fitur ini juga cocok untuk anggota keluarga yang ingin menggunakan ponsel bersama-sama tanpa berbagi informasi pribadi. Melalui fitur pengguna tamu di ponsel dan tablet, Anda dapat meminjamkan perangkat dan menjaga informasi pribadi tetap aman. Pin ke layar: pasang pin agar pengguna lain hanya dapat mengakses konten tertentu tanpa mengotak-atik item lainnya.

6. Setelan Cepat Baru

Kontrol baru yang praktis seperti senter, hotspot, pemutaran layar, dan layar transmisi.

7. Konektivitas

Penyempurnaan handoff jaringan yang mengakibatkan terbatasnya gangguan konektivitas. Misalnya, Anda dapat melanjutkan obrolan video atau panggilan VoIP tanpa gangguan saat keluar rumah dan beralih dari Wi-Fi rumah ke seluler.

8. Waktu Proses dan Performa

ART (*Android Run Time*), waktu proses Android yang benar-benar baru, menyempurnakan performa dan daya respons aplikasi. Memadatkan aplikasi dan layanan latar belakang agar Anda dapat melakukan lebih banyak hal sekaligus. Dukungan untuk perangkat 64 bit, seperti Nexus 9, menghadirkan CPU kelas desktop ke Android.

9. Media

Masukan audio latensi rendah memastikan aplikasi musik dan komunikasi dengan persyaratan penundaan yang ketat memberikan pengalaman langsung yang menakjubkan. Dengan campuran aliran audio multi-saluran, aplikasi audio profesional kini dapat mencampur hingga delapan saluran, termasuk saluran 5.1 dan 7.1. Dengan dukungan Audio USB, Anda dapat mencolokkan mikrofon atau pengeras suara USB, dan banyak perangkat audio USB lainnya seperti amplifier dan mixer ke perangkat Android. Teknologi video yang canggih dengan dukungan HEVC untuk pemutaran video UHD 4K, video yang disalurkan untuk pemutaran video berkualitas tinggi di Android TV, dan penyempurnaan dukungan HLS untuk streaming.

10. Aksesibilitas

Penyempurnaan kemampuan untuk pengguna yang mengalami gangguan penglihatan dan buta warna. Meningkatkan kontras teks atau inversi warna untuk meningkatkan keterbacaan. Sesuaikan tampilan untuk meningkatkan diferensiasi warna.



Gambar 2.3 Logo Android Versi 5.1

(Sumber : <https://goo.gl/images/tf1LrT>)

2.3.4 XAMPP (*Apache PHP*)

XAMPP adalah sebuah paket *web server* yang mengandung *Apache* dan *PHP*, serta *MySQL*. *XAMPP* tersedia untuk multisystem operasi, seperti *Windows* dan *Linux* (EMS, 2016:11).



Gambar 2.4 Logo *XAMPP*

(Sumber : <https://goo.gl/images/9UJ9Vk>)

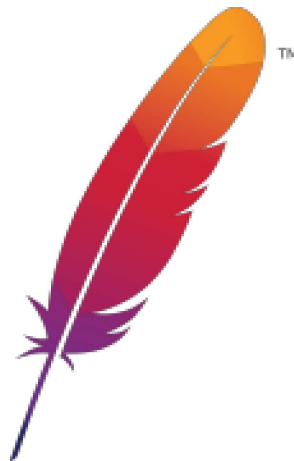
PHP adalah singkatan dari *PHP Hypertext Preprocessing*. Merupakan bahasa *scripting* untuk *web* dimana bisa membuat *web* dinamis dengan menyelipkan *script* kode-kode *HTML* yang merupakan bahasa *markup* standar untuk dunia *web*. *PHP* 5.0 memasukkan model pemrograman berorientasi objek ke dalam *PHP* untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek (EMS, 2016:1 - 4).



Gambar 2.5 Logo *PHP*

(Sumber : <https://goo.gl/images/zSfClB>)

Apache digunakan sebagai *web server* dari aplikasi yang akan kita buat. *Apache* berorientasi *web* sehingga sangat cocok untuk dunia *web*, kecepatan responnya tinggi. Komunikasi *PHP* dan *MySQL* sangat baik serta kompatibilitasnya tinggi. Bisa dikostumisasi karena kode sumber keduanya terbuka (EMS, 2016:10 - 11).



Gambar 2.6 Logo *Apache*

(Sumber : <https://goo.gl/images/3zjlDv>)

2.3.5 MySQL

Menurut Raharjo (2011:3) *database* didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat.

Menurut Raharjo (2011:21) MySQL merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Beberapa alasan menggunakan MySQL sebagai server database untuk aplikasi-aplikasi yang dikembangkan (Raharjo, 2011:22) :

1. Fleksibel

MySQL dapat digunakan untuk mengembangkan aplikasi *desktop* maupun aplikasi web dengan menggunakan teknologi yang bervariasi. Ini berarti bahwa MySQL memiliki fleksibilitas terhadap teknologi yang akan digunakan sebagai pengembang aplikasi, apakah itu PHP, JSP, Jav, Delphi, C++, maupun yang lainnya dengan cara menyediakan *plug-in* dan *driver* yang spesifik untuk masing-masing teknologi tersebut. Dalam database MySQL juga memiliki dukungan terhadap *stored procedure*, fungsi, *trigger*, *view*, SQL standar ANSI, dll, yang akan mempermudah dan mempercepat proses pengembangan aplikasi.

2. Performa Tinggi

MySQL memiliki mesin *query* dengan performa tinggi, dengan demikian proses transaksional dapat dilakukan dengan sangat cepat. Hal ini terbukti dengan digunakannya MySQL sebagai *database* dari beberapa aplikasi *web* yang memiliki *traffic* (lalu lintas) sangat tinggi.

3. Lintas Platform

MySQL dapat digunakan pada *platform* atau lingkungan (dalam hal ini Sistem Operasi) yang beragam, bisa Microsoft Windows, Linux, atau UNIX. Ini menyebabkan proses migrasi data (bila dibutuhkan) antarsistem operasi dapat dilakukan secara lebih mudah.

4. Gratis

MySQL dapat digunakan secara gratis.

5. Proteksi Data yang Handal

Perlindungan terhadap keamanan data merupakan hal nomor satu yang dilakukan oleh para professional di bidang *database*. MySQL menyediakan mekanisme yang *powerfull* untuk menangani hal tersebut, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data, dan lain sebagainya.

6. Komunitas Luas

Karena penggunaannya banyak maka MySQL memiliki komunitas yang luas. Hal ini berguna jika kita menemui suatu permasalahan dalam proses pengolahan data menggunakan MySQL.

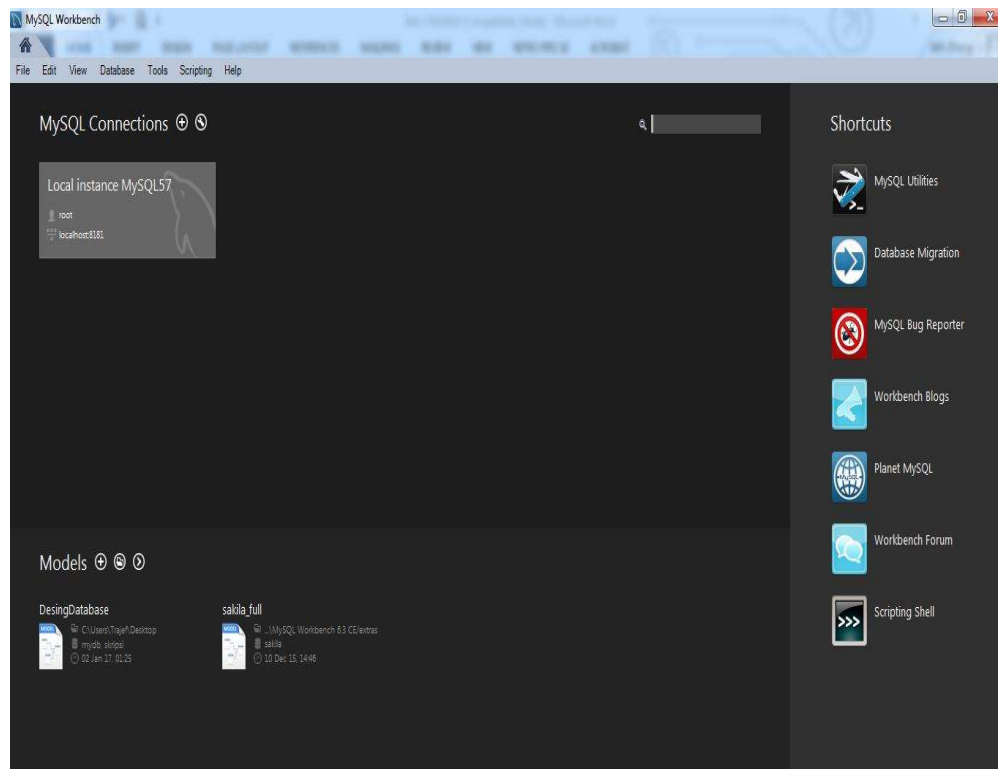
Menurut Hendry (2015: 7) MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis di bawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukkan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Menurut Hendry (2015: 7) MySQL memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, Mac Os X Server, dan masih banyak lagi.
2. **Perangkat lunak sumber terbuka.** MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. **Multi-user.** MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. **Performance tuning,** MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. **Ragam tipe data.** MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. **Perintah dan Fungsi.** MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. **Keamanan.** MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. **Skalabilitas dan Pembatasan.** MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. **Konektivitas.** MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
10. **Lokalisasi.** MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. **Antar Muka.** MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. **Klien dan Peralatan.** MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. **Struktur tabel.** MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.



Gambar 2.7 Tampilan Utama Aplikasi MySQL Workbench 6.3 CE
(Sumber: Data Penelitian, 2016)

2.3.5.1 *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) merupakan salah satu alat bantu (berupa gambar) dalam model *database* relasional yang berguna untuk

menjelaskan hubungan atau relasi antar tabel yang terdapat di dalam *database*. Dalam ERD kita juga dapat melihat daftar kolom yang menyusun masing-masing tabel (Raharjo, 2011:57).

2.3.6 Unified Modeling Language (UML)

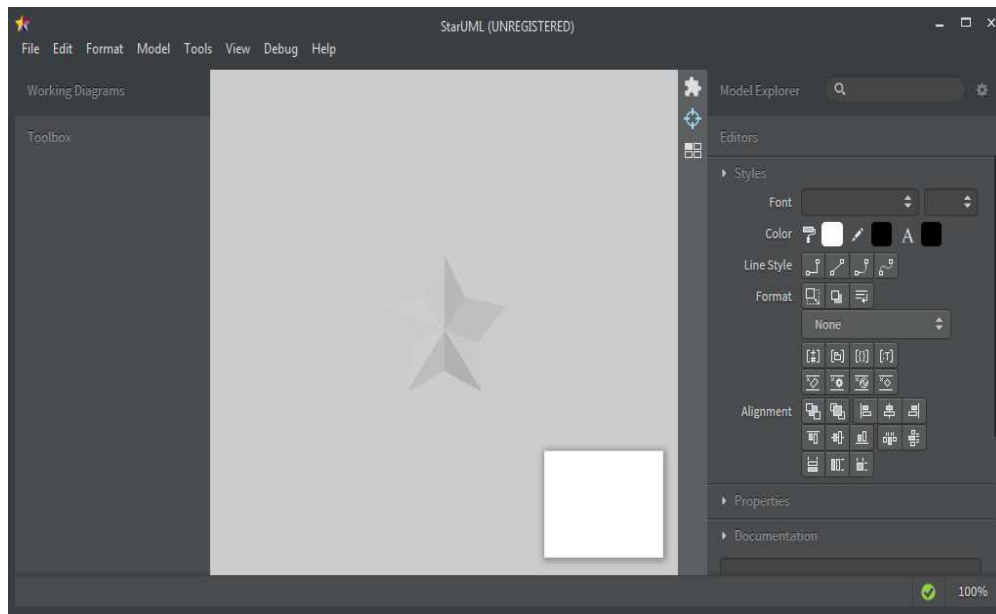
Menurut A.S dan Shalahuddin (2016:136) perangkat pemodelan adalah suatu model yang digunakan untuk menguraikan sistem menjadi bagian-bagian yang dapat diatur dan mengomunikasikan ciri konseptual dan fungsional kepada pengamat. Peran perangkat pemodelan :

1. Perangkat pemodelan bisa digunakan sebagai alat komunikasi antara pemakai dengan analis sistem maupun *developer* dalam pengembangan sistem.
2. Sebuah eksperimentasi yang bersifat "*trial and error*".
3. Model yang dibuat bisa meramalkan bagaimana suatu sistem akan bekerja.

Menurut A.S dan Shalahuddin (2016:133) UML (*Unified Modelling Language*) adalah salah satu standar bahasa yang digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Perangkat lunak yang membantu dalam mengrekayasa *UML* yang dinamakan CASE (*Computer-Aided Software Engineering*) tools. *CASE tools* mulai digunakan oleh pelaku rekayasa perangkat lunak sejak tahun 1980-an dalam hal ini mereka focus pada tahap awal pengembangan. Salah satu *tools* atau

perangkat lunak tersebut adalah StarUML. StarUML digolongkan kedalam kelompok *upper Case tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak (A.S dan Shalahuddin, 2016:122-123).



Gambar 2.8 Tampilan Utama Aplikasi StarUML
(Sumber: Data Penelitian, 2016)

2.3.6.1 *Class Diagram*

Class diagram berada pada bagian struktur diagram yang berguna menggambarkan kelas-kelas yang akan dibuat untuk membangun sebuah sistem. Sebuah kelas terdiri dari atribut dan operasi atau metode. Atribut adalah variabel-variabel atau kolom yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (A.S dan Shalahuddin, 2016:141-142).

Menurut A.S dan Shalahuddin (2016:142) susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem (*View*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

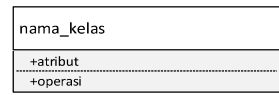


Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.


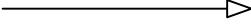
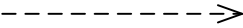
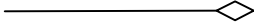
4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas, bisa di lihat pada tabel di bawah ini :

Tabel 2.1 Simbol pada *Class diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

Asosiasi berarah / <i>direct association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan kelas
Agresi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: A.S dan Shalahuddin (2016:146-147)

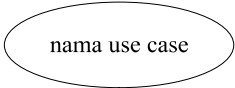
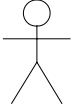

2.3.6.2 Use Case Diagram

Use case atau diagram *use case* berfungsi untuk menjelaskan fungsi yang ada pada suatu sistem dan siapa yang berhak menggunakan fungsi tersebut. *Use case* diagram merupakan bagian dari diagram kelakuan (*behavior*) (A.S dan Shalahuddin, 2016:155).

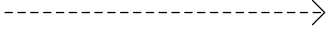
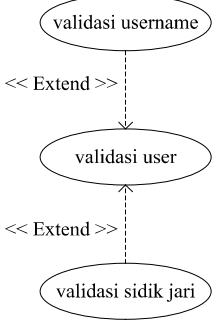

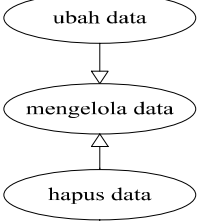
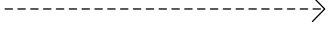

Menurut A.S dan Shalahuddin (2016:155) ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

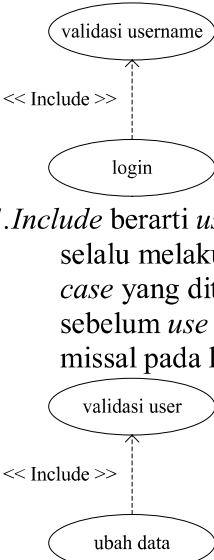
Tabel 2.2 Simbol pada *Use case diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
<p>Aktor / <i>actor</i></p>  <p>Nama aktor</p>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi.
<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<p>Ekstensi / <i>extend</i></p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya:

Tabel 2.2 Lanjutan

<p><< Extend >></p> 	 <p>arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>Generalisasi / <i>generalization</i></p> 	 <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasainya (umum).</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><< Include >></p>  <p><< Uses >></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>: <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>

Tabel 2.2 Lanjutan


	 <pre> graph TD login((login)) -.-> << Include >> validasi_username((validasi username)) ubah_data((ubah data)) -.-> << Include >> validasi_user((validasi user)) </pre> <p>1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p> <p>Kedua sudut pandang diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan kebutuhan</p>
--	---





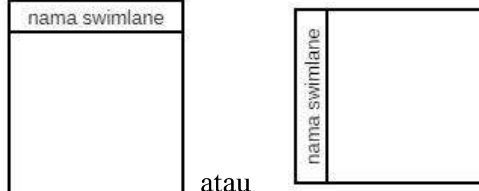
Sumber: A.S dan Shalahuddin (2016:156-158)

2.3.6.3 Activity Diagram

Activity diagram juga merupakan bagian dari diagram kelakuan (*behavior*). *Activity diagram* berguna untuk mendeskripsikan *workflow* (aliran kerja) dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. *Activity diagram* berbeda dengan *use case diagram* karna dia tidak menggambarkan apa yang dilakukan aktor, tetapi menggambarkan aktivitas yang dilakukan sistem (A.S dan Shalahuddin, 2016:161).

Tabel 2.3 Simbol pada *Activity diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal

Tabel 2.3 Lanjutan	
Aktifitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

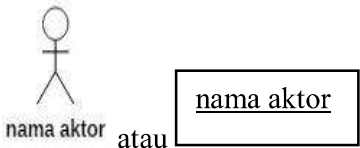
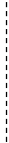
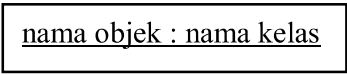

Sumber: A.S dan Shalahuddin (2016:162-163)

2.3.6.4 *Sequence Diagram*

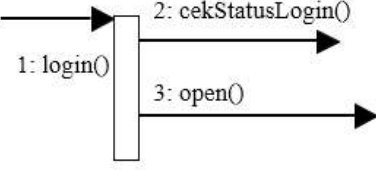
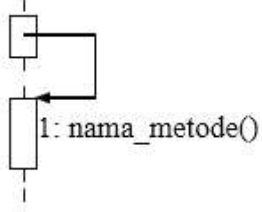
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Diagram sekuen yang sudah dibuat bisa digunakan untuk melihat

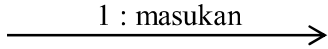
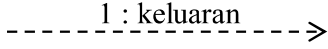
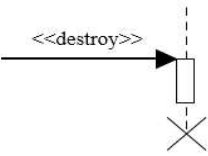
scenario yang mungkin terjadi pada *use case*. Sehingga jumlah diagram sekuen akan banyak jika jumlah diagram *use case* banyak (A.S dan Shalahuddin, 2016:165).

Tabel 2.4 Simbol pada *Sequence diagram*

Simbol	Deskripsi
<p>Aktor/actor</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>

Tabel 2.4 Lanjutan

	 <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif.</p>
<p>Pesan tipe create</p> <p>—————>><<create>>—————></p> <p>pesan tipe call</p> <p>1 : nama_metode() —————></p>	<p>Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat</p> <p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>  <p>arah panah mengarah pada objek yang memiliki operasi/metode, maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>

Tabel 2.4 Lanjutan	
Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber: A.S dan Shalahuddin (2016:165-167)

2.4 Penelitian Terdahulu

Dalam penelitian ini penulis memaparkan beberapa penelitian terdahulu yang relevan dengan permasalahan yang akan diteliti tentang analisis sistem pakar pertumbuhan anak standar who berbasis android dengan metode forward chaining.

Fidiantoro dan Setiadi (2013:367), *Model Penentuan Status Gizi Balita Di Puskesmas*, status gizi balita merupakan faktor penting yang harus diperhatikan karena masa balita merupakan periode perkembangan yang rentan dengan gizi. Upaya pemerintah dalam perbaikan gizi dengan memantau status gizi balita di

setiap wilayah kerja Puskesmas. Hasil penelitian ini berupa aplikasi model penentuan status gizi balita di Puskesmas yang hasilnya dapat membantu petugas gizi untuk menentukan status gizi balita di Puskesmas agar lebih efektif dan akurat.

Zamroni, dkk. (2013:507) pada jurnalnya yang berjudul “*Sistem Pakar Perkembangan Anak Usia 0-12 Bulan Berbasis Web Dengan Metode Forward Chaining*”, masalah yang dihadapi adalah banyaknya bayi yang lahir pada saat ini. Tetapi dengan bertambahnya tersebut masih ada saja orang tua yang belum faham atau kurang mengertinya tentang perkembangan anak secara baik dan sesuai dengan usia yang benar. Salah satu tujuan penelitian yaitu menyediakan media informasi yang mudah pada masyarakat dalam mengambil keputusan sehingga dapat segera mengetahui perkembangan anak tergolong yang seperti apa. Salah satu manfaat penelitian adalah masyarakat *non-pakar* dapat memanfaatkan untuk mengetahui perkembangan anak secara baik dan benar. Pada sistem pakar ini digunakan metode Forward Chaining, dimana proses dimulai dengan memilih fakta-fakta yang telah disediakan selanjutnya akan ditentukan seberapa jauh perkembangan anak tersebut. Saran yang di berikan yaitu “Perlu diadakan penambahan data untuk jenis perkembangan usia selanjutnya sehingga informasi yang dimiliki akan semakin luas dan banyak.”

Penelitian Ernawati, dkk. (2014:109), *Hubungan Panjang Badan Lahir Terhadap Perkembangan Anak Usia 12 Bulan*, menyatakan hasil penelitian menunjukkan sebanyak 9,5 persen bayi dengan berat badan lahir rendah (BBLR) dan 22 persen stunting. Nilai z-skor panjang badan terhadap umur pada bayi baru

lahir berkorelasi dengan perkembangan motorik dan sosial emosi sejak bayi berumur nol bulan, yaitu $\rho=0,33$; $p=0,004$ untuk motorik dan $\rho=0,244$ dengan $p=0,036$ untuk sosial emosi. Sedangkan korelasi terhadap perkembangan bahasa baru tampak pada saat bayi berumur satu bulan yaitu $\rho=0,29$ dengan $p=0,031$ dan korelasi terhadap perkembangan kognitif terjadi pada usia dua bulan $\rho=0,318$ dengan $p=0,011$. Pada anak lahir stunting median perkembangan bahasa lebih rendah dibandingkan kelompok yang normal. Pendek (*stunting*) adalah gangguan pertumbuhan pada anak balita di Indonesia yang perlu mendapat perhatian khusus. Salah satu dampaknya *stunting*, terutama pada anak usia kurang dua tahun yang mengakibatkan penurunan tingkat kecerdasan.

Penelitian Shaid, dkk. (2015:37), *Sistem Pakar Pertumbuhan Balita Berbasis Web Dengan Metode Case Based Reasoning*, menyatakan pertumbuhan balita bisa terjadi berdasarkan beberapa factor, yaitu berdasarkan kelahirannya dan pertumbuhan gizi yang dikonsumsi. Dengan memanfaatkan metode Case-base Reasoning, dapat dihasilkan suatu aplikasi untuk mengidentifikasi pertumbuhan balita. Dengan harapan sistem ini nantinya dapat digunakan sebagai sarana atau sebagai pengetahuan dalam menjaga kestabilan pertumbuhan balita dan membantu anda untuk mendapatkan hasil yang optimal dalam menjaga pertumbuhan setiap balita. Metode Case Based Reasoning (CBR) digunakan dalam aplikasi Pertumbuhan Balita dengan menggunakan Perhitungan Nearest Neighbor, Dimana data kasus baru akan dibandingkan perhitungannya dengan data kasus lama yang ada di database, dan kemudian dihitung kriteria kemiripannya berdasarkan rumus atau ketentuan yang berlaku. Dimana Shaid, dkk.

(2015: 44) menyarankan “Program ini dapat lebih disempurnakan, khususnya pada bagian proses pencarian data kasus. Hal ini akan memberikan hasil konsultasi yang diberikan perangkat lunak lebih baik.”

Nugraha, dkk. (2015:11), *Analisis Sistem Pakar Cara Diet Berdasarkan Golongan Darah*, saat ini perkembangan teknologi begitu pesat, begitu pesatnya hingga muncul kecerdasan buatan yang memiliki sistem pakar. Dari hal tersebut kami melakukan analisis tentang sistem pakar yang menyangkut cara diet berdasarkan golongan darah. Dari hal itu adapula penelitian yang dilakukan dari sebuah buku untuk mengambil data dan mencari sampel untuk uji coba program, maka dibuatlah suatu program yang berbasis web dari cara diet berdasarkan golongan darah.

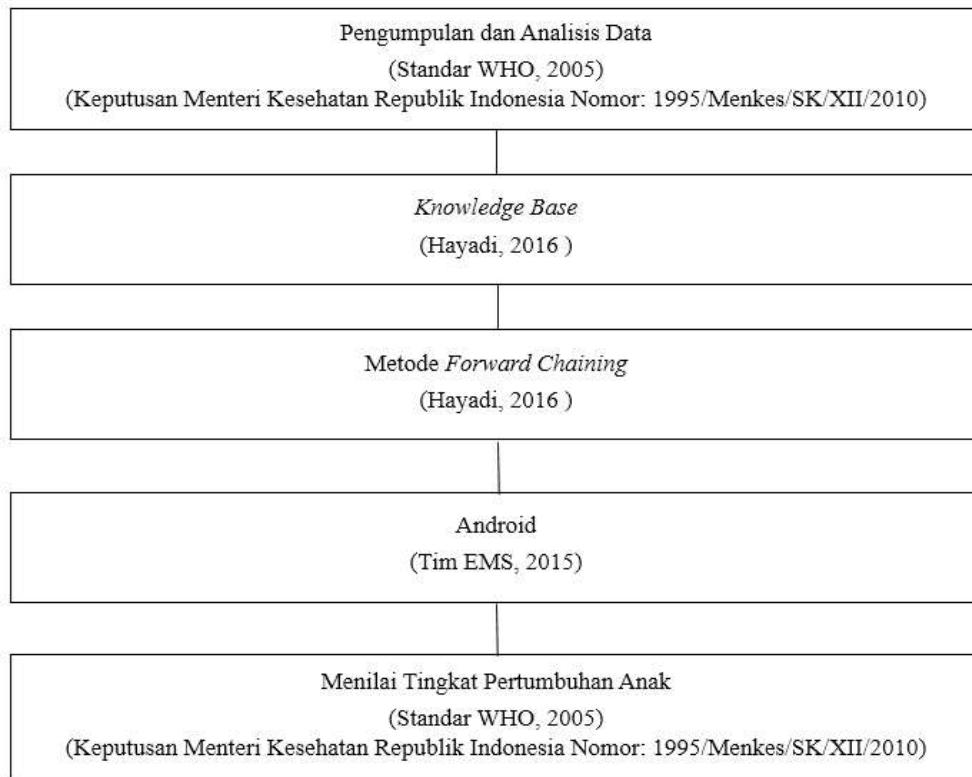
Rengganis (2016:20), *Perancangan Sistem Pakar Klasifikasi Status Gizi Balita Berdasarkan Indeks Antropometri Berat Badan Terhadap Umur (BB/U) menggunakan Metode Forward Chaining*, anak berusia dibawah lima tahun (balita) merupakan golongan usia yang sangat rentan akan kebutuhan gizi dan masalah kesehatan. Namun kurangnya kesadaran masyarakat akan gizi dan kurangnya sarana informasi penunjang kebutuhan gizi menyebabkan masalah ini kurang mendapat penanganan secara tepat dan cepat. Peneliti menyarankan bahwa penelitian lanjutan dengan menambahkan indeks antropometri lainnya.

Purwati (2016:12), *Klasifikasi Status Gizi Balita Berdasarkan Indeks Antropometri Bb/U Dan Bb/Tb Menggunakan Jaringan Saraf Tiruan*, status gizi balita dapat ditentukan berdasarkan indeks antropometri BB/U dan BB/TB dengan menggunakan standar baku WHO-NCHS. Penelitian ini bertujuan untuk

mengklasifikasikan status gizi menggunakan Jaringan Saraf Tiruan *Backpropagation* berdasarkan indeks antropometri BB/U yang akan menghasilkan status gizi kedalam gizi buruk, kurang, baik, dan lebih, serta mengklasifikasikan status gizi berdasarkan indeks antropometri BB/TB yang akan menghasilkan status sangat kurus, kurus, normal, dan gemuk. Variabel yang digunakan dalam penelitian ini yaitu jenis kelamin, umur, berat badan, tinggi badan, dan status ekonomi.

2.5 Kerangka Pemikiran

Kerangka Berpikir adalah sebuah gambaran yang menghubungkan variabel-variabel untuk mendeskripsikan dan menjelaskan masalah serta menghasilkan sebuah hipotesis atau solusi. Jadi, kerangka berpikir adalah sintesis tentang hubungan antar variabel yang disusun dari berbagai teori yang dideskripsikan (Sudaryono, 2015:21). Kerangka berfikir pada penelitian ini dapat dilihat pada gambar 2.9.



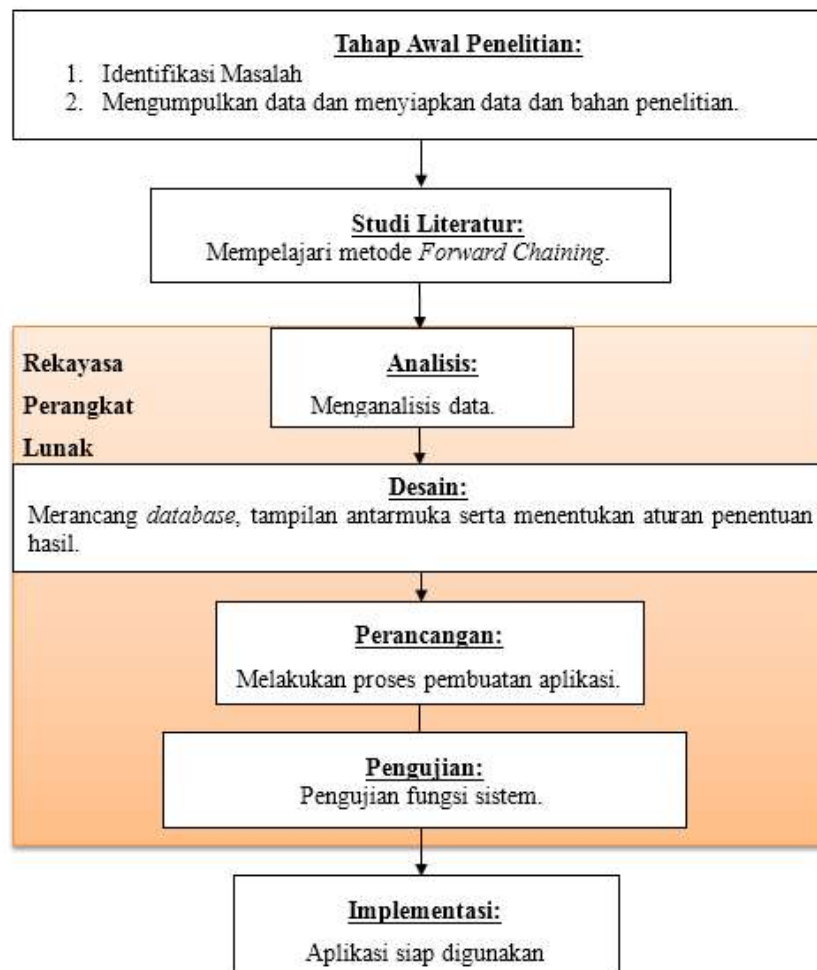
Gambar 2.9 Kerangka Pemikir
(Sumber: Data Penelitian, 2016)

Mengumpulkan data-data yang berkaitan dengan penilaian tingkat pertumbuhan balita standar WHO 2005. Melakukan analisis pada data yang didapatkan untuk menghasilkan sebuah basis pengetahuan (*knowledge base*). Sehingga bisa digunakan dalam perancangan sebuah sistem pakar dengan menggunakan metode *forward chaining*. Sistem pakar ini akan dirancang menjadi sebuah aplikasi Android. Dimaksudkan agar aplikasi android ini dapat menguraikan penilaian tingkat pertumbuhan balita dan dapat digunakan dengan mudah oleh masyarakat *non-pakar*.

BAB III METODOLOGI PENELITIAN

3.1 Desain Penelitian

Desain penelitian mencakup antara lain tahapan awal, studi literatur, analisis, desain, perancangan, pengujian, implementasi.



Gambar 3.1 Desain Penelitian
(Sumber: Data Penelitian, 2016)

Penjelasan desain penelitian:

1. Tahap awal penelitian

Penelitian dimulai dengan menentukan kebutuhan data penelitian dan pengumpulan data.

2. Studi Literatur

Studi literatur dilakukan dengan cara mempelajari aspek aspek yang berkaitan dengan penelitian ini, teori metode *forward chaining* dan perhitungan hasil. Data yang digunakan dalam studi literatur didapat dengan cara mengumpulkan jurnal, penelusuran internet, dan buku yang berkaitan dengan topik.

3. Rekayasa perangkat lunak

Setelah itu dilanjutkan membangun perangkat lunak dari penelitian ini, dimana rekayasa perangkat lunak yang digunakan terdiri dari analisis, desain, perancangan, dan tes.

4. Implementasi

Pada tahap ini sistem yang dirancang siap untuk di gunakan.

3.2 Teknik Pengumpulan Data

Pengumpulan data dalam penelitian dimaksudkan untuk memperoleh bahan, keterangan, kenyataan, dan informasi yang dapat dipercaya. Untuk memperoleh

data seperti yang dimaksudkan, dalam penelitian dapat digunakan berbagai metode, di antaranya angket, pengamatan, wawancara, tes, analisa dokumen, dan sebagainya. Peneliti dapat menggunakan salah satu atau gabungannya tergantung pada masalah yang dihadapi.

Teknik yang digunakan dalam penelitian ini adalah Studi Literatur. Studi Literatur adalah cara untuk menyelesaikan persoalan dengan menelusuri sumber-sumber tulisan yang pernah dibuat sebelumnya. Pencarian dan pengumpulan bahan yang berhubungan dengan topik melalui studi pustaka, buku-buku referensi dan sumber lain seperti jurnal, makalah, dan internet.

3.3 Operasional Variabel

Tabel dibawah ini adalah penjelasan akan variabel agar dapat diukur dengan melihat pada dimensi sebuah variabel.

Tabel 3.1 Operasional Variabel

Variabel	Definisi Konseptual	Definisi Operasional		
		Dimensi	Skor Motivasi	Skala Pengukuran
Umur	Umur balita adalah indeks dasar untuk melakukan pengukur tingkat pertumbuhan balita	1. <i>Length/height - for-age</i> 2. <i>Weight - for-age</i>	0 bulan sampai 12 bulan	nominal

Tabel 3.1 Lanjutan

<i>Length/height-for-age</i>	Tinggi badan memberikan gambaran fungsi pertumbuhan yang dilihat dari keadaan kurus kering dan kecil pendek.	1. Umur	40 cm sampai 125 cm	nominal
<i>Weight-for-age</i>	Berat badan merupakan salah satu ukuran yang memberikan gambaran massa jaringan, termasuk cairan tubuh.	1. Umur	2 kg sampai 28 kg	nominal

Sumber: Data Penelitian, 2016

Tabel 3.2 Kategori dan Ambang Batas Status Gizi Berdasarkan Variabel

Variabel	Kode	Kategori Status Gizi	Ambang Batas (SD)
Berat Badan menurut Umur (BB/U) Anak Umur 0 – 1 Bulan	G1	Gizi Kurang	<-2 SD
	G2	Gizi Baik	-2 SD sampai dengan 2 SD
	G3	Gizi Lebih	>2 SD
Panjang Badan menurut Umur (PB/U) atau Tinggi Badan menurut Umur (TB/U) Anak Umur 0 – 1 Bulan	G1	Gizi Kurang	<-2 SD
	G2	Gizi Baik	-2 SD sampai dengan 2 SD
	G3	Gizi Lebih	> 2 SD

Sumber: Data Penelitian, 2016

Tabel 3.3 Tabel Gejala / Sifat

Kode	Gejala / Sifat
K1	Jenis kelamin anak laki-laki
K2	Umur anak Anda berkisar 0 – 1 bulan
K3	Umur anak Anda berkisar 1 – 2 bulan
K4	Umur anak Anda berkisar 2 – 3 bulan
K5	Umur anak Anda berkisar 3 – 4 bulan
K6	Umur anak Anda berkisar 4 – 5 bulan
K7	Umur anak Anda berkisar 5 – 6 bulan

Tabel 3.3 Lanjutan

K8	Umur anak Anda berkisar 6 – 7 bulan
K9	Umur anak Anda berkisar 7 – 8 bulan
K10	Umur anak Anda berkisar 8 – 9 bulan
K11	Umur anak Anda berkisar 9 – 10 bulan
K12	Umur anak Anda berkisar 10 – 11 bulan
K13	Umur anak Anda berkisar 11 – 12 bulan
K14	Berat badan anak berada pada kisaran 2.5 – 4.4 Kg (Berdasarkan Standar WHO 2005, kisaran berat sesuai dengan umur dan jenis kelamin)
K15	Panjang badan anak berada pada kisaran 46.1 – 53.7 cm (Berdasarkan Standar WHO 2005, kisaran panjang sesuai dengan umur dan jenis kelamin)
K16	Berat melebihi kisaran
K17	Tinggi melebihi kisaran
K18	Diberikan ASI eksklusif sesering mungkin tanpa batas waktu menyesuaikan dengan kebutuhan bayi
K19	Diberikan MP-ASI dengan frekuensi pemberian 2/3 kali sehari yang jumlahnya disesuaikan dengan umur
K20	MP-ASI yang diberikan mengandung banyak garam, gula, MSG dan bahan penyedap
K21	Badan nampak kurus, perut cekung dan sering disertai penyakit infeksi (umumnya kronis) dan diare (Gizi Buruk Marasmus)
K22	Edema pada seluruh tubuh, terutama di punggung kaki, pandangan mata sayu, suka rewel dan kelainan kulit berupa bercak merah muda yang meluas dan berubah warna menjadi coklat (Gizi buruk Kwashiorkor)

Sumber: Data Penelitian, 2016

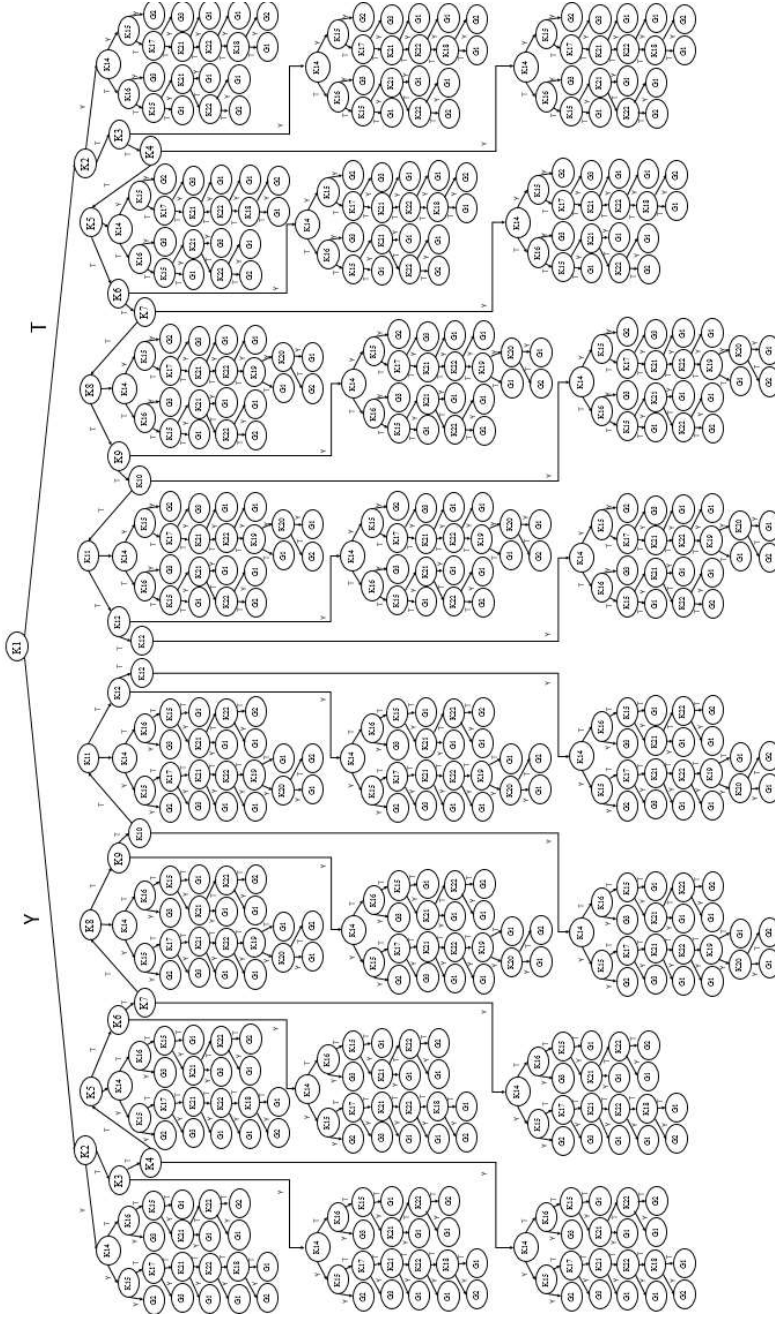
Tabel 3.4 Tabel Keputusan

Indikator	Gejala	K 14	K 15	K 16	K 17	K 18	K 19	K 20	K 21	K 22
G1	K2							✓	✓	✓
	K3							✓	✓	✓
	K4							✓	✓	✓
	K5							✓	✓	✓
	K6							✓	✓	✓
	K7							✓	✓	✓
	K8							✓	✓	✓
	K9							✓	✓	✓
	K10							✓	✓	✓
	K11							✓	✓	✓
	K12							✓	✓	✓
	K13							✓	✓	✓
	G2	K2	✓	✓			✓			

	K3	✓	✓			✓				
	K4	✓	✓			✓				
	K5	✓	✓			✓				
	K6	✓	✓			✓				
	K7	✓	✓			✓				
	K8	✓	✓				✓			
	K9	✓	✓				✓			
	K10	✓	✓				✓			
	K11	✓	✓				✓			
	K12	✓	✓				✓			
	K13	✓	✓				✓			
G3	K2			✓	✓					
	K3			✓	✓					
	K4			✓	✓					
	K5			✓	✓					
	K6			✓	✓					
	K7			✓	✓					
	K8			✓	✓					
	K9			✓	✓					
	K10			✓	✓					
	K11			✓	✓					
	K12			✓	✓					
	K13			✓	✓					

Sumber: Data Penelitian, 2016

Berikut merupakan gambar pohon keputusan dengan metode *forward chaining*.



Gambar 3.2 Pohon keputusan
(Sumber: Data penelitian, 2016)

Dari gambar pohon keputusan diatas, dibawah ini adalah penjelasan kode-kode yang digunakan.

1. Kode K1 menyatakan jenis kelamin anak, apakah jenis kelamin anak laki-laki.
2. Kode K2 menanyakan umur anak, apakah umur anak kisaran 0 sampai 1 bulan.
3. Kode K14 menanyakan apakah berat badan anak berada pada standar WHO 2005 dalam kategori gizi normal
4. Kode K15 menanyakan apakah tinggi badan anak berada pada standar WHO 2005 dalam kategori gizi normal
5. G1 menyatakan bahwa berat badan anak berada pada kategori gizi normal.

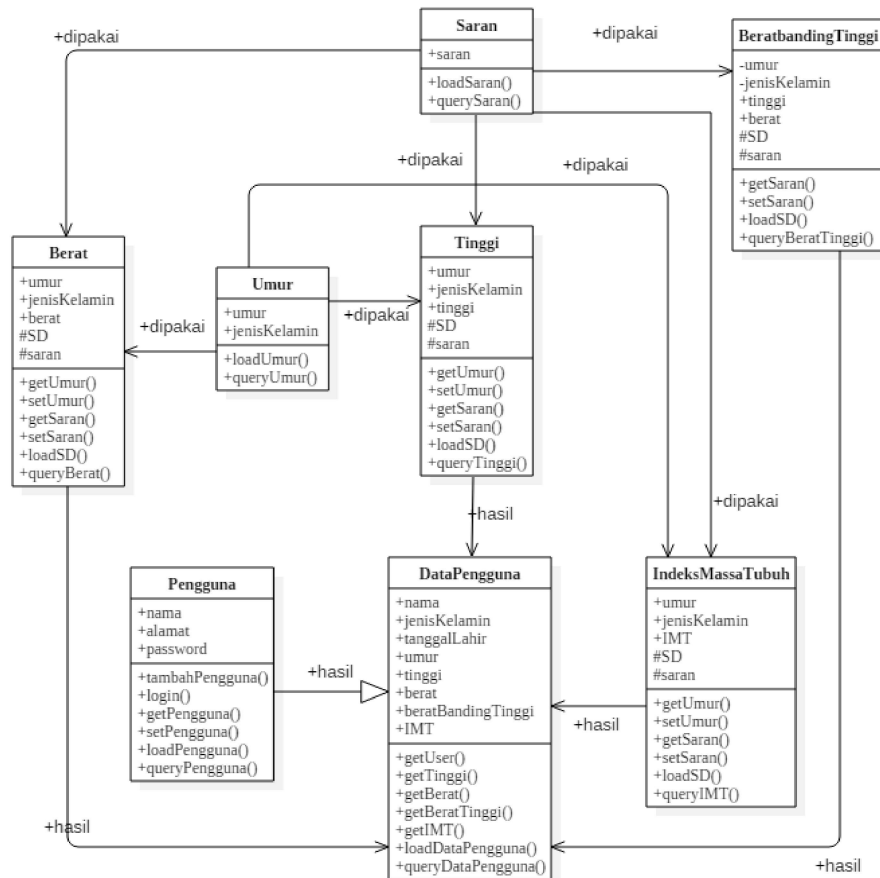
Salah satu contoh cara baca pohon keputusan tersebut. Seorang anak laki-laki yang berumur 1 bulan, memiliki berat badan 3.3 kg dan panjang badan 52 cm. Maka sesuai dengan Standar WHO 2005, anak akan di nilai berat badannya berada pada ambang batas 0 SD dan panjang badan berada pada ambang batas 1 SD. Sehingga untuk penilaian berdasarkan standar WHO 2005 menyatakan anak bergizi normal.

3.4 Perancangan Sistem

3.4.1 UML (Unified Modelling Language)

3.4.1.1 Class Diagram

Diagram di bawah ini menggambarkan struktur dan deksripsi *class*, operasi dan atribut yang akan dirancang pada aplikasi Android.

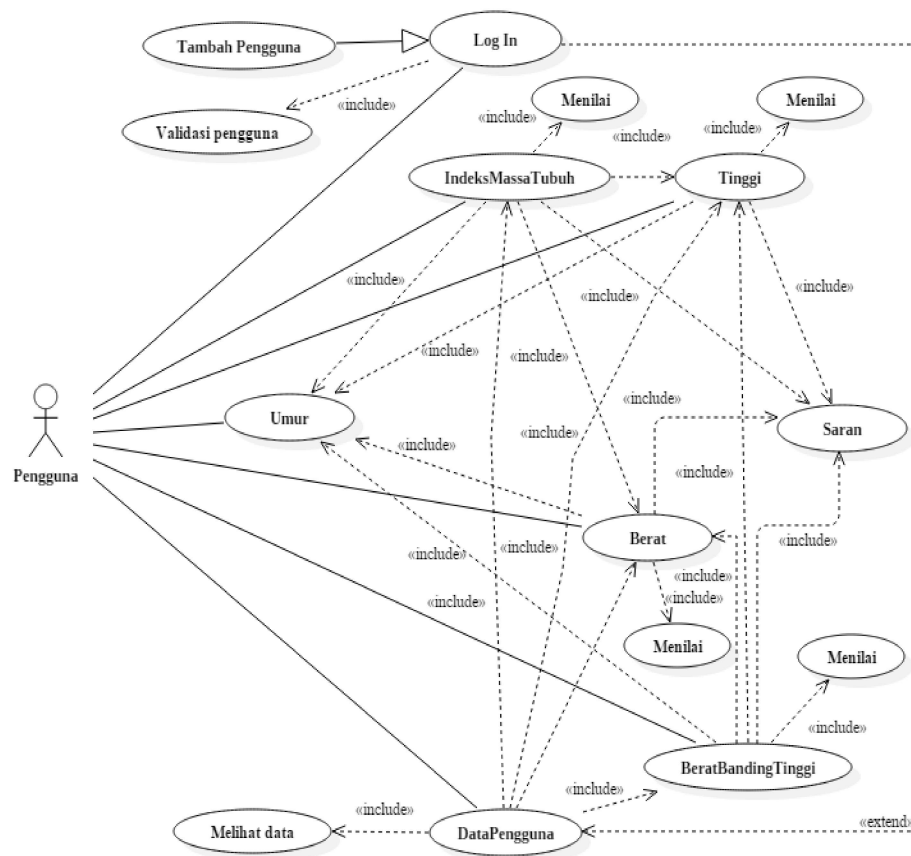


Gambar 3.3 Class diagram
(Sumber: Data penelitian, 2016)

3.4.1.2 Use Case Diagram

1. Use case Diagram Pengguna biasa

Diagram dibawah ini menjelaskan aktivitas atau interaksi seorang aktor atau pengguna biasa dengan sistem.



Gambar 3.4 Use case diagram pengguna biasa
(Sumber: Data penelitian, 2016)

Aktor di sini adalah seorang pengguna biasa, yang akan menggunakan aplikasi sistem pakar ini. Berikut adalah deskripsi *Use Case* pada aplikasi sistem pakar.

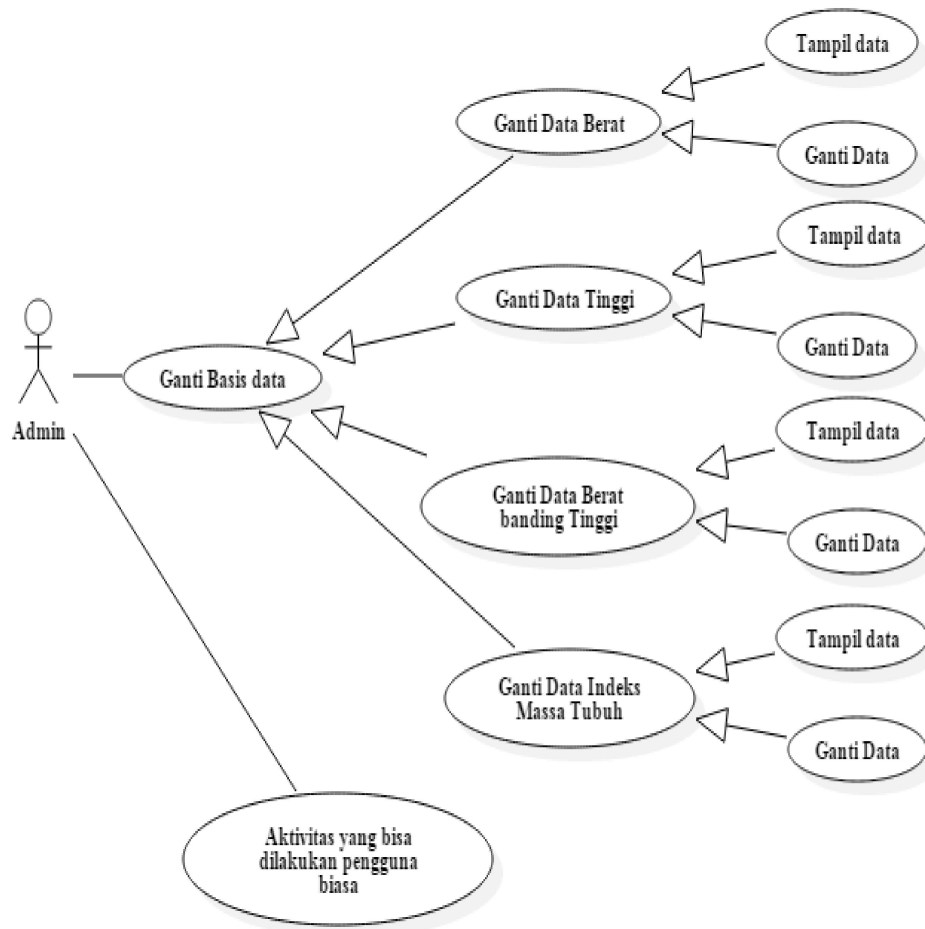
Tabel 3.5 Deskripsi *Use Case* Pengguna

No	Use Case	Deskripsi
1	<i>Log In</i>	Sebuah unit yang berfungsi untuk aktor melakukan <i>login</i> ke sistem
2	Validasi Pengguna	Sebuah unit yang harus dijalankan terlebih dahulu sebelum menjalankan <i>use case login</i> .
3	Tambah Pengguna	Sebuah unit khusus dari <i>use case login</i> , yang berguna untuk menambah pengguna baru
4	Umur	Sebuah unit yang akan digunakan untuk aktor menginput nilai umur.
5	Berat	Sebuah unit yang akan digunakan untuk aktor menginput nilai berat.
6	Tinggi	Sebuah unit yang akan digunakan untuk aktor menginput nilai tinggi,
7	BeratBandingTinggi	Sebuah unit yang akan digunakan untuk menghitung nilai berat banding tinggi berdasarkan nilai yang didapatkan dari <i>use case berat dan tinggi</i>
8	IndeksMassaTubuh	Sebuah unit yang akan digunakan untuk menghitung nilai indeks massa tubuh berdasarkan nilai yang didapatkan dari <i>use case berat dan tinggi</i>
9	Saran	Sebuah unit yang menyediakan pesan saran yang dibutuhkan oleh <i>use case menilai</i> untuk menghasilkan sebuah hasil akhir
10	Menilai	Sebuah unit yang bertukar pesan antar <i>use case umur, berat, tinggi, berat banding tinggi, indeks massa tubuh dan saran</i> untuk menghasilkan hasil akhir
11	Data Pengguna	Sebuah unit yang berfungsi menyimpan data-data dari <i>use case umur, berat, tinggi, berat banding tinggi, indeks massa tubuh dan saran</i>
12	Melihat data	Sebuah unit khusus dari <i>use case login</i> , yang berguna untuk melihat data pengguna

Sumber: Data Penelitian, 2016

2. Use case Diagram Admin

Diagram dibawah ini menjelaskan aktivitas atau interaksi seorang aktor atau admin dengan sistem.



Gambar 3.5 Use case diagram admin
(Sumber: Data penelitian, 2016)

Aktor di sini adalah seorang admin, yang akan menggunakan aplikasi sistem pakar ini. Berikut adalah deskripsi *Use Case* pada aplikasi sistem pakar.

Tabel 3.6 Deskripsi *Use Case* Admin

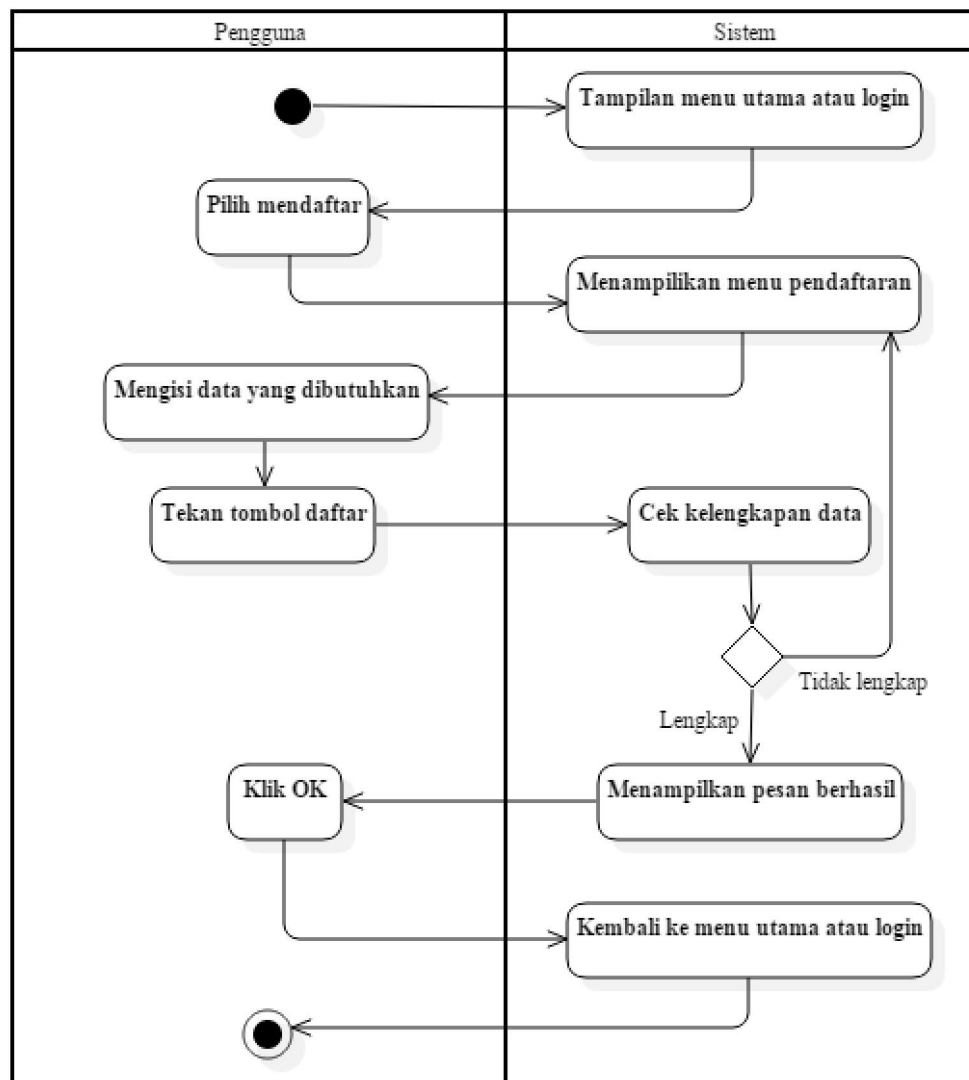
No	Use Case	Deskripsi
1	Ganti Basis Data	Sebuah unit yang berfungsi untuk aktor memilih basis data yang akan di ganti.
2	Ganti Data Berat	Sebuah unit yang berfungsi untuk aktor melihat atau merubah data berat.
3	Ganti Data Tinggi	Sebuah unit yang berfungsi untuk aktor melihat atau merubah data tinggi.
4	Ganti Data Berat banding Tinggi	Sebuah unit yang berfungsi untuk aktor melihat atau merubah data berat banding tinggi.
5	Ganti Data Indeks Massa Tubuh	Sebuah unit yang berfungsi untuk aktor melihat atau merubah data indeks massa tubuh.
6	Ganti Data	Sebuah unit khusus dari <i>use case</i> Ganti data berat, tinggi, berat banding tinggi dan indeks massa tubuh, yang berguna untuk mengganti data
7	Tampil Data	Sebuah unit khusus dari <i>use case</i> Ganti data berat, tinggi, berat banding tinggi dan indeks massa tubuh, yang berguna untuk melihat data
8	Aktivitas yang bisa dilakukan pengguna biasa	Unit ini menjelaskan bahwa hal-hal (<i>use case</i>) yang bisa dilakukan aktor pengguna biasa juga bisa dilakukan oleh aktor admin.

Sumber: Data Penelitian, 2016

3.4.1.3 Activity Diagram

1. Activity Diagram Daftar

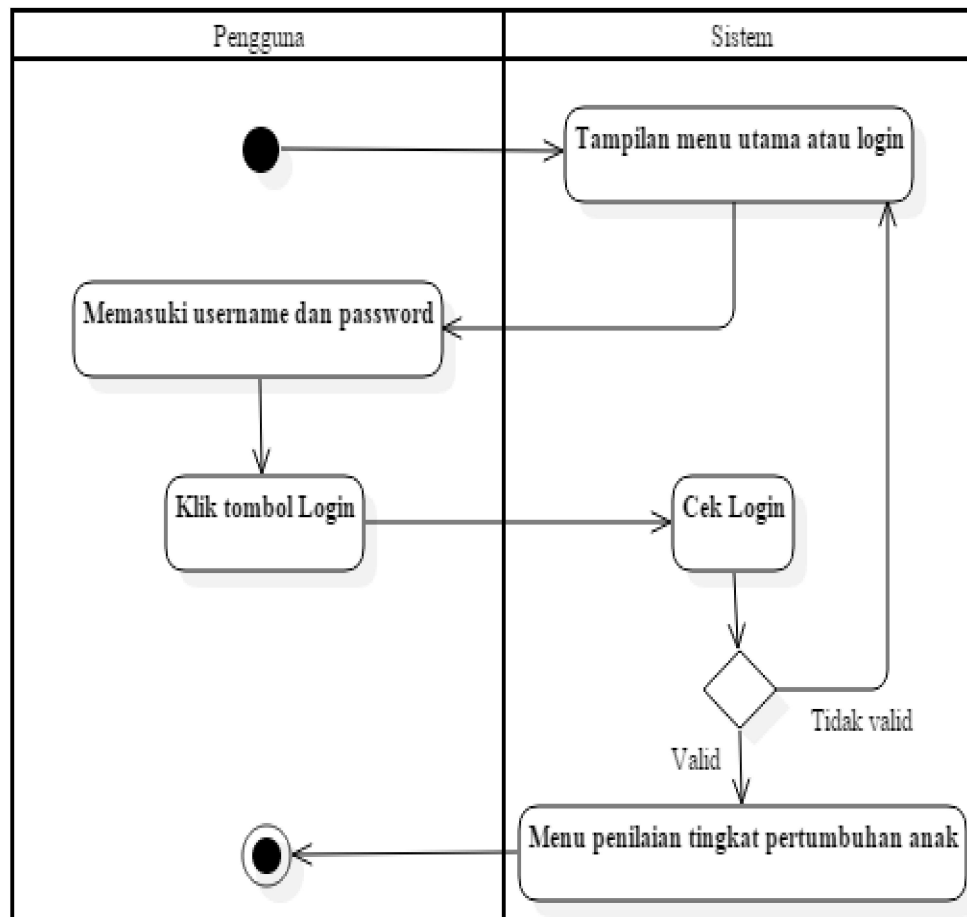
Diagram ini menjelaskan proses yang terjadi ketika pengguna melakukan pendaftaran pada aplikasi.



Gambar 3.6 Activity diagram daftar
(Sumber: Data penelitian, 2016)

2. Activity Diagram Login

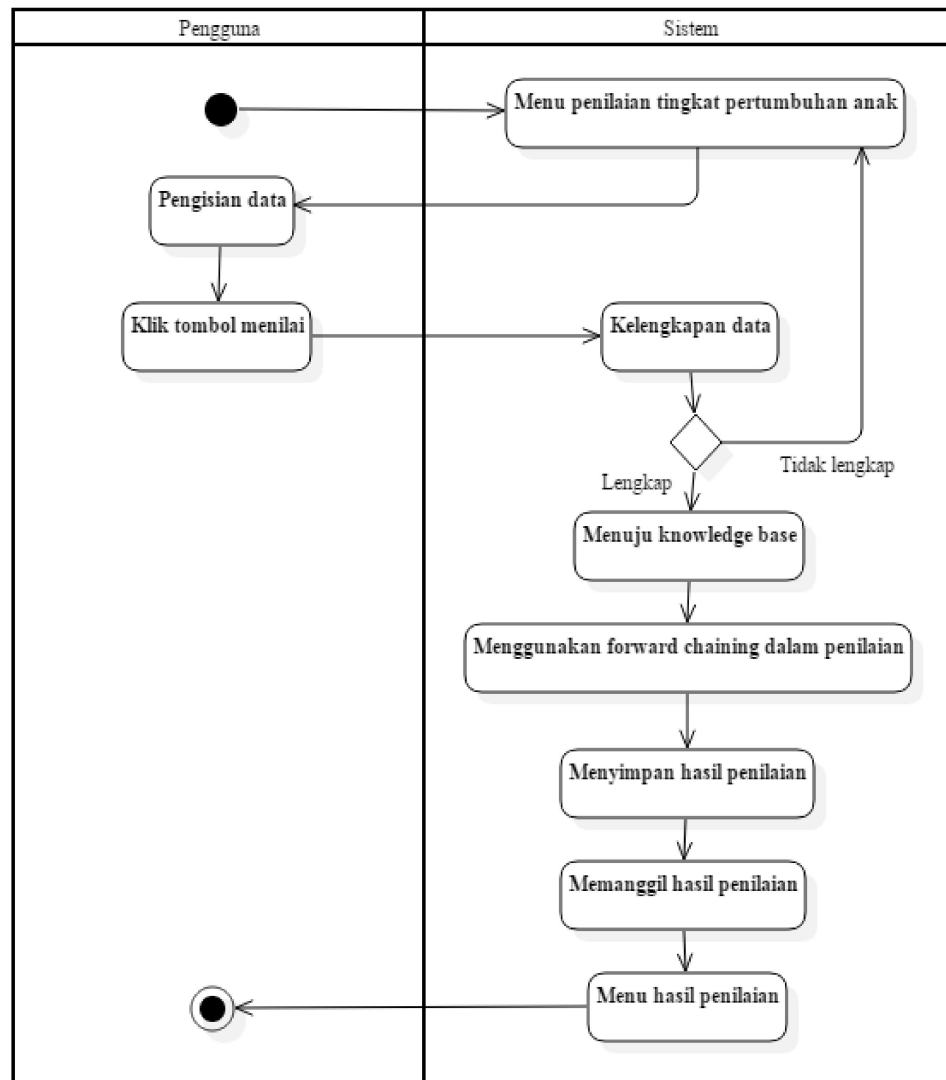
Diagram ini menjelaskan proses yang terjadi ketika pengguna melakukan *login* pada aplikasi. Sebelumnya, pengguna harus telah melakukan pendaftar, jika tidak pengguna tidak akan bisa berhasil *login* ke aplikasi.



Gambar 3.7 Activity diagram login
(Sumber: Data penelitian, 2016)

3. Activity Diagram penilaian tingkat pertumbuhan anak

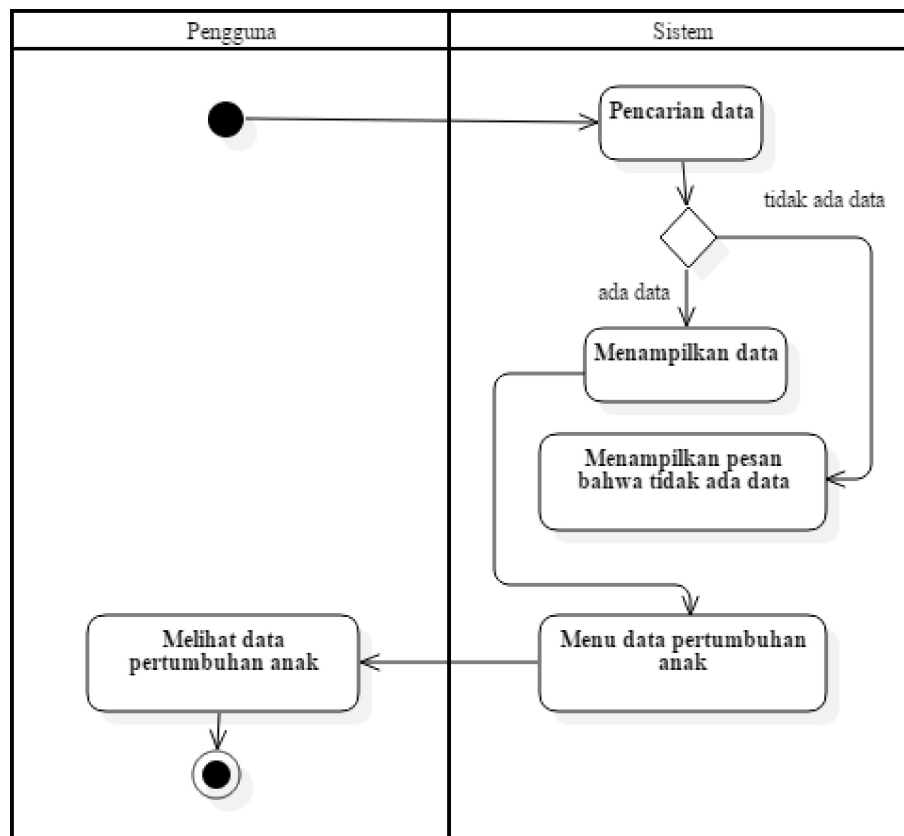
Diagram ini menjelaskan proses yang terjadi ketika pengguna melakukan aktivitas untuk mengetahui tingkat pertumbuhan anak pada aplikasi. Pengguna harus telah berhasil *login* untuk melakukan aktivitas ini.



Gambar 3.8 Activity diagram penilaian tingkat pertumbuhan anak
(Sumber: Data penelitian, 2016)

4. Activity Diagram data pertumbuhan anak

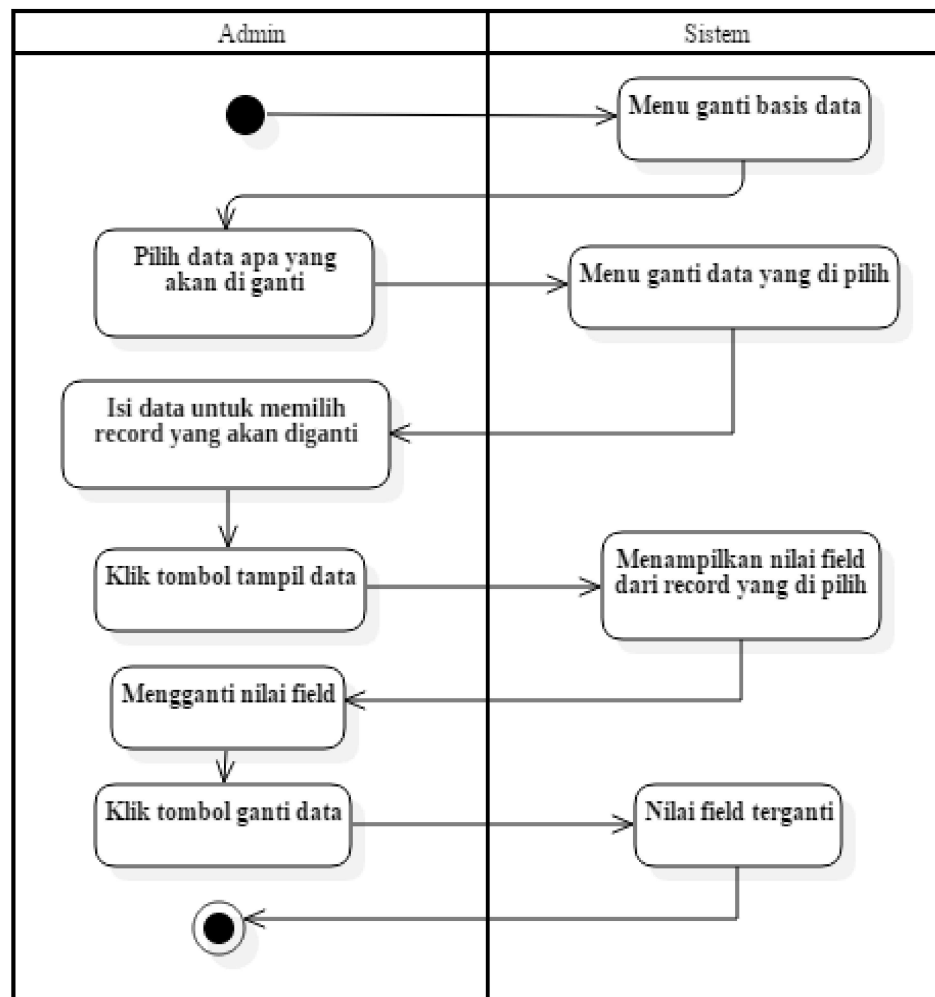
Diagram ini menjelaskan proses yang terjadi ketika pengguna melihat hasil penilaian pertumbuhan anak secara berkala. Pengguna harus telah berhasil *login* untuk melakukan aktivitas ini.



Gambar 3.9 Activity diagram data pertumbuhan anak
(Sumber: Data penelitian, 2016)

5. Activity Diagram ganti basis data

Diagram ini menjelaskan proses yang terjadi ketika pengguna mengganti basis data secara berkala. Pengguna harus telah berhasil *login* untuk melakukan aktivitas ini.

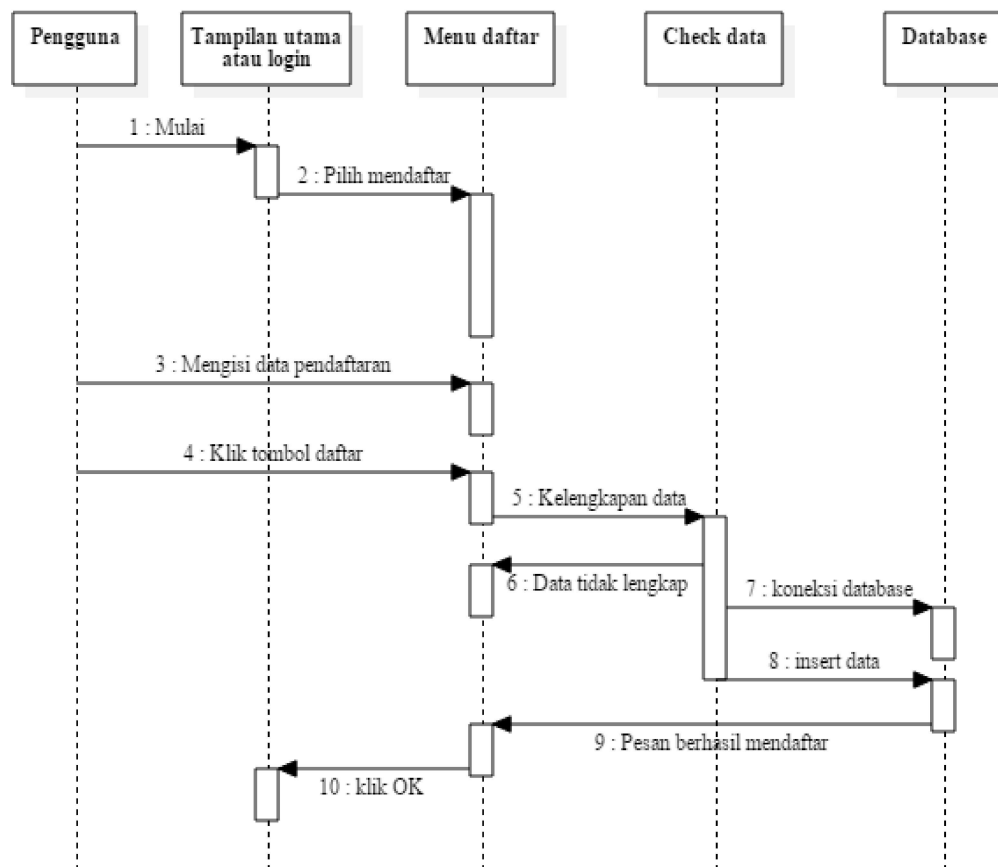


Gambar 3.10 Activity diagram ganti basis data
(Sumber: Data penelitian, 2017)

3.4.1.4 Sequence Diagram

1. Sequence Diagram Daftar

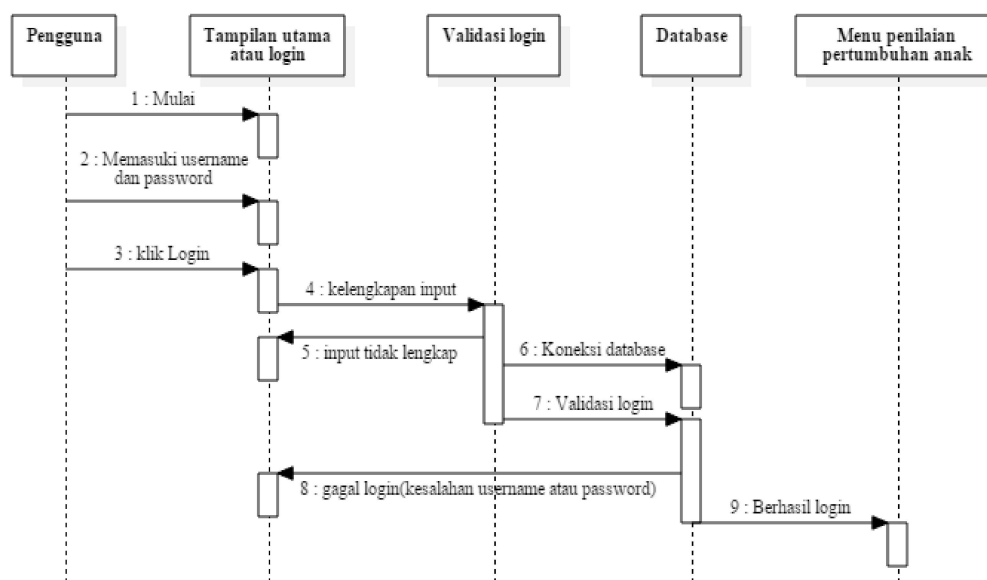
Diagram ini menjelaskan waktu hidup objek dan *message* yang dikirim dan diterima antar objek ketika pengguna melakukan pendaftaran pada aplikasi.



Gambar 3.11 Sequence diagram daftar
(Sumber: Data penelitian, 2016)

2. Sequence Diagram Login

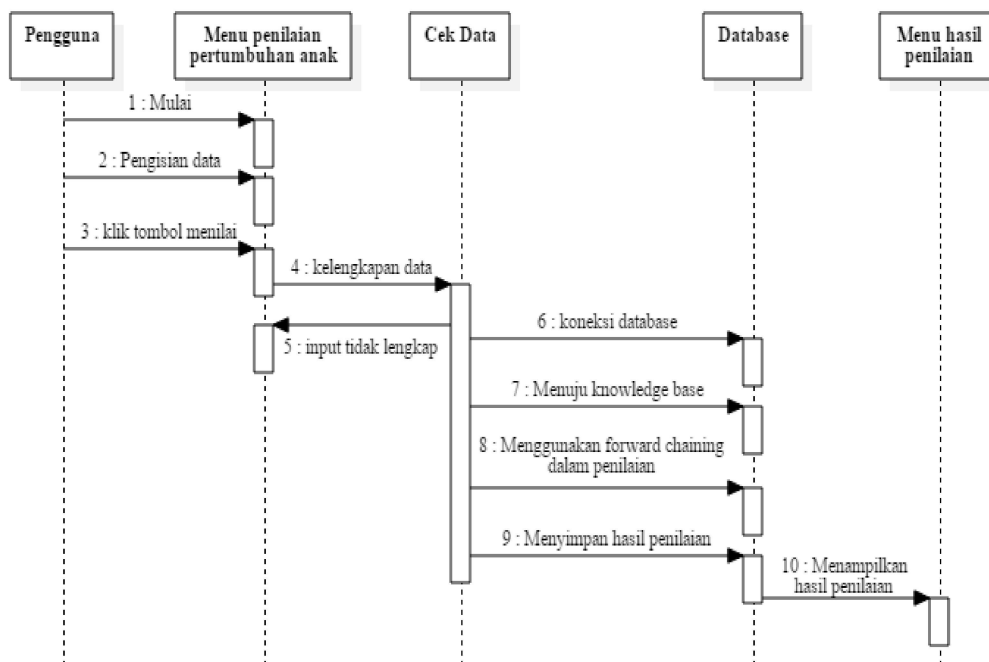
Diagram ini menjelaskan waktu hidup objek dan *message* yang dikirim dan diterima antar objek ketika pengguna melakukan *login* pada aplikasi. Sebelumnya, pengguna harus telah melakukan pendaftar, jika tidak pengguna tidak akan bisa berhasil *login* ke aplikasi.



Gambar 3.12 Sequence diagram login
(Sumber: Data penelitian, 2016)

3. *Sequence* Diagram penilaian tingkat pertumbuhan anak

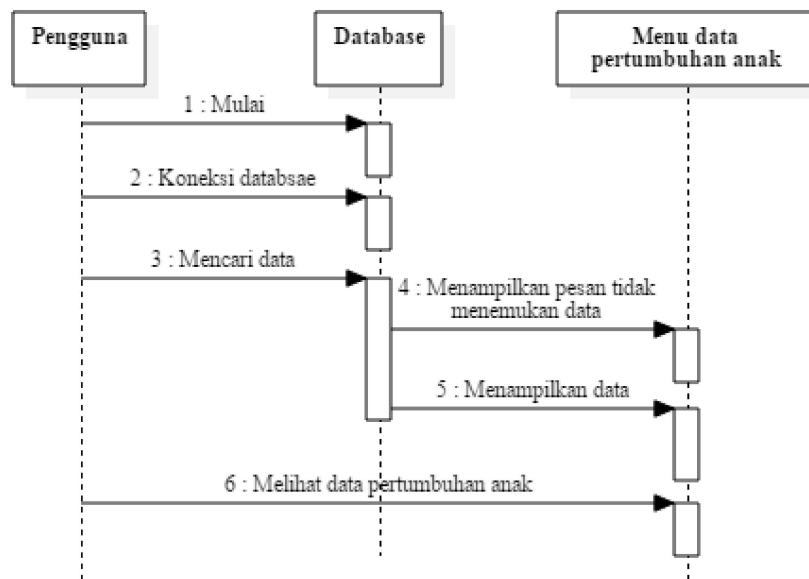
Diagram ini menjelaskan waktu hidup objek dan *message* yang dikirim dan diterima antar objek ketika pengguna melakukan aktivitas untuk mengetahui tingkat pertumbuhan anak pada aplikasi. Pengguna harus telah berhasil *login* untuk melakukan aktivitas ini.



Gambar 3.13 *Sequence* diagram penilaian tingkat pertumbuhan anak
(Sumber: Data penelitian, 2016)

4. *Sequence* Diagram data pertumbuhan anak

Diagram ini menjelaskan waktu hidup objek dan *message* yang dikirim dan diterima antar objek ketika pengguna melihat hasil penilaian pertumbuhan anak secara berkala. Pengguna harus telah berhasil *login* untuk melakukan aktivitas ini.

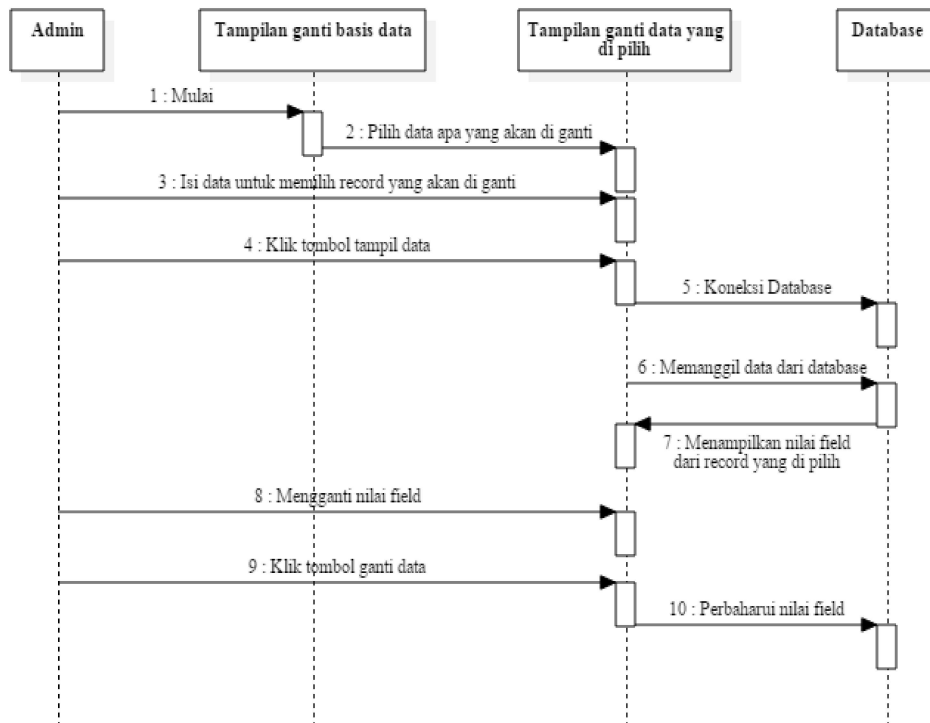


Gambar 3.14 *Sequence* diagram data pertumbuhan anak
(Sumber: Data penelitian, 2016)

5. Sequence Diagram ganti basis data

Diagram ini menjelaskan waktu hidup objek dan *message* yang dikirim dan diterima antar objek ketika pengguna mengganti basis data secara berkala.

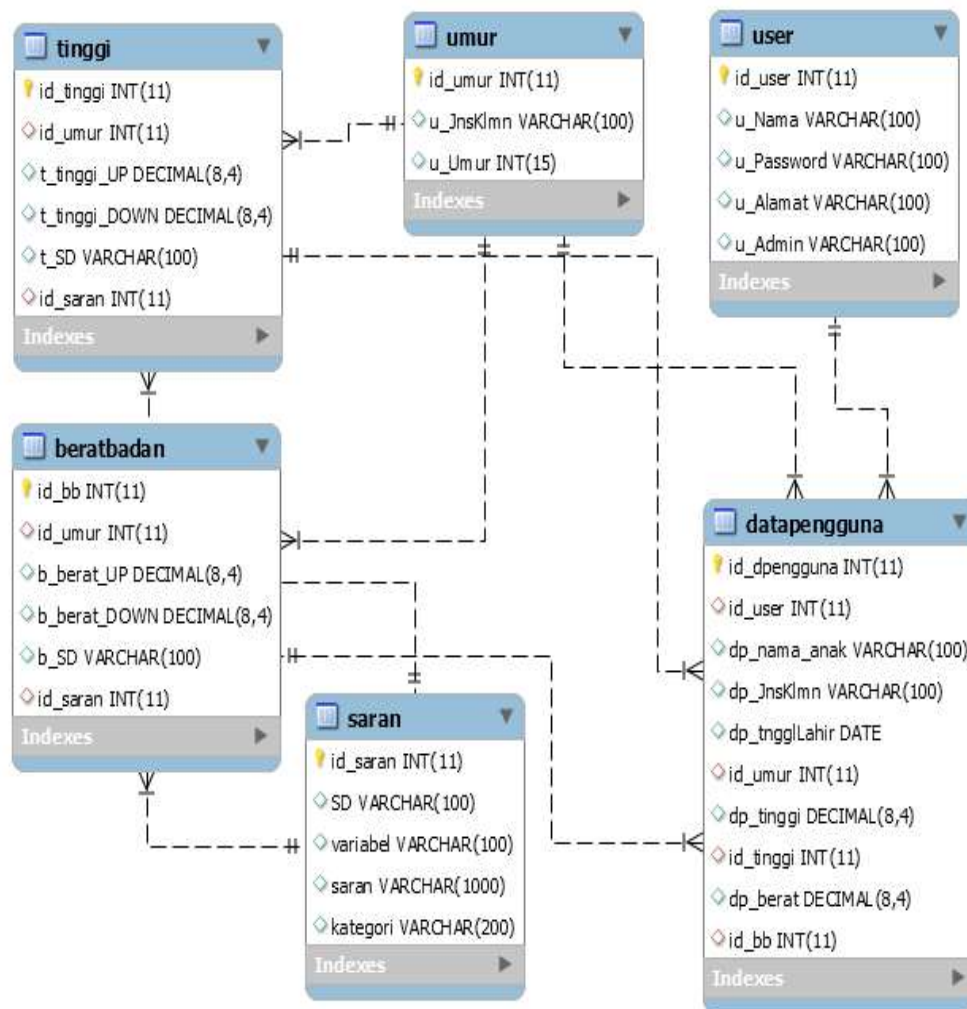
Pengguna harus telah berhasil *login* untuk melakukan aktivitas ini.



Gambar 3.15 Sequence diagram ganti basis data
(Sumber: Data penelitian, 2017)

3.4.2 Desain Basis Data

Desain basis data yang akan dirancang dalam penelitian ini bisa di lihat pada gambar dibawah. Gambar di bawah menjelaskan hubungan-hubungan antar tabel yang di telah dirancang atau yang dinamakan *Entity Relationship Diagram* (ERD).



Gambar 3.16 Rancangan *database*
(Sumber: Data Penelitian, 2016)

Berikut rincian tabel-tabel yang ada pada gambar 3.16 Rancangan *database*, sebagai berikut :

Tabel 3.7 Rincian tabel *user*

Nama Kolom	Tipe Data	Kunci	Keterangan
id_user	INT(11), <i>Auto Increment</i>	Primary	Id rekor user
u_Nama	VARCHAR(100), <i>Nullable</i>		Menyimpan nama
u_Password	VARCHAR(100), <i>Nullable</i>		Menyimpan <i>password</i>
u_Alamat	VARCHAR(100), <i>Nullable</i>		Menyimpan alamat
u_Admin	VARCHAR(100), <i>Nullable</i>		Menyimpan akses admin

Sumber: Data Penelitian, 2017

Tabel 3.8 Rincian tabel umur

Nama Kolom	Tipe Data	Kunci	Keterangan
id_umur	INT(11), <i>Auto Increment</i>	Primary	Id rekor umur
u_JnsKlmm	VARCHAR(100), <i>Nullable</i>		Menyimpan jenis kelamin
u_Umur	INT(15), <i>Nullable</i>		Menyimpan umur

Sumber: Data Penelitian, 2017

Tabel 3.9 Rincian tabel beratbadan

Nama Kolom	Tipe Data	Kunci	Keterangan
id_bb	INT(11), <i>Auto Increment</i>	Primary	Id rekor berat badan
id_umur	INT(11), <i>Nullable</i>	Foreign	Id rekor umur dari tabel umur
b_berat_UP	DECIMAL(8,4), <i>Nullable</i>		Menyimpan batas atas berat badan
b_berat_DOWN	DECIMAL(8,4), <i>Nullable</i>		Menyimpan batas bawah berat badan
b_SD	VARCHAR(100), <i>Nullable</i>		Menyimpan nilai SD berat badan

Tabel 3.9 Lanjutan

id_saran	INT(11), <i>Nullable</i>	Foreign	Id rekor saran dari tabel saran
----------	--------------------------	---------	---------------------------------

Sumber: Data Penelitian, 2017

Tabel 3.10 Rincian tabel tinggi

Nama Kolom	Tipe Data	Kunci	Keterangan
id_tinggi	INT(11), <i>Auto Increment</i>	<i>Primary</i>	Id rekor tinggi
id_umur	INT(11), <i>Nullable</i>	<i>Foreign</i>	Id rekor umur dari tabel umur
t_tinggi_UP	DECIMAL(8,4), <i>Nullable</i>		Menyimpan batas atas tinggi badan
t_tinggi_DOWN	DECIMAL(8,4), <i>Nullable</i>		Menyimpan batas bawah tinggi badan
t_SD	VARCHAR(100), <i>Nullable</i>		Menyimpan nilai SD berat badan
id_saran	INT(11), <i>Nullable</i>	<i>Foreign</i>	Id rekor saran dari tabel saran

Sumber: Data Penelitian, 2017

Tabel 3.11 Rincian tabel saran

Nama Kolom	Tipe Data	Kunci	Keterangan
id_saran	INT(11), <i>Auto Increment</i>	<i>Primary</i>	Id rekor saran
saran	VARCHAR(1000), <i>Nullable</i>		Menyimpan saran
variabel	VARCHAR(100), <i>Nullable</i>		Menyimpan berat, tinggi, berat banding tinggi atau imt
kategori	VARCHAR(200), <i>Nullable</i>		Menyimpan nilai SD

Sumber: Data Penelitian, 2017

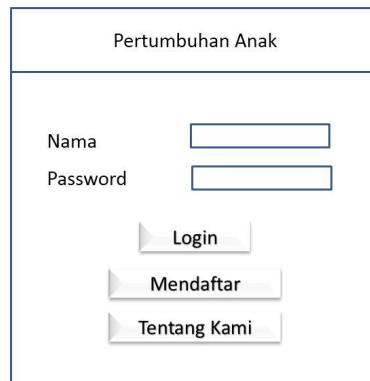
Tabel 3.12 Rincian tabel datapengguna

Nama Kolom	Tipe Data	Kunci	Keterangan
id_dpengguna	INT(11), <i>Auto Increment</i>	<i>Primary</i>	Id rekor data pengguna
id_user	INT(11), <i>Nullable</i>	<i>Foreign</i>	Id rekor user dari tabel user
dp_nama_anak	VARCHAR(100), <i>Nullable</i>		Menyimpan nama anak
dp_JnsKlmm	VARCHAR(100), <i>Nullable</i>		Menyimpan jenis kelamin
dp_tggLahir	DATE, <i>Nullable</i>		Menyimpan tanggal lahir
id_umur	INT(11), <i>Nullable</i>	<i>Foreign</i>	Id rekor umur dari tabel umur
dp_tinggi	DECIMAL(8,4), <i>Nullable</i>		Menyimpan tinggi badan
id_tinggi	INT(11), <i>Nullable</i>	<i>Foreign</i>	Id tinggi dari tabel tinggi
dp_berat	DECIMAL(8,4), <i>Nullable</i>		Menyimpan berat badan
id_bb	INT(11), <i>Nullable</i>	<i>Foreign</i>	Id berat dari tabel beratbadan

Sumber: Data Penelitian, 2017

3.4.3 Prototype

Prototype memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan. Dibawah ini adalah *prototype-prototype* dari aplikasi yang akan dirancang.



Pertumbuhan Anak

Nama

Password

Login

Mendaftar

Tentang Kami

Gambar 3.17 Tampilan halaman utama atau *login*
(Sumber: Data Penelitian, 2016)



Pertumbuhan Anak

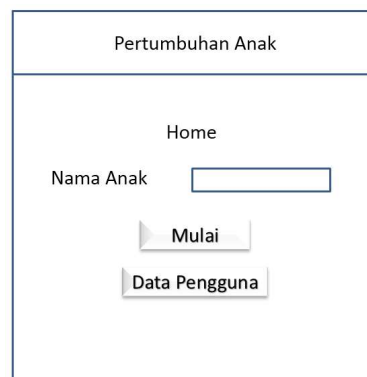
Nama

Alamat

Password

Daftar

Gambar 3.18 Halaman pendaftaran
(Sumber: Data Penelitian, 2016)



Pertumbuhan Anak

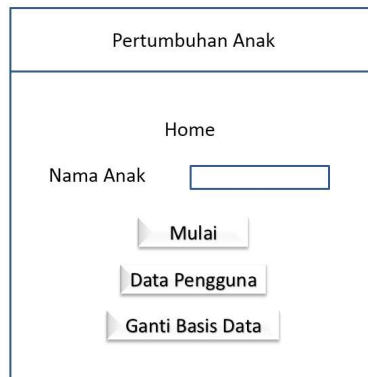
Home

Nama Anak

Mulai

Data Pengguna

Gambar 3.19 Halaman *home* (pengguna)
(Sumber: Data Penelitian, 2016)



Pertumbuhan Anak

Home

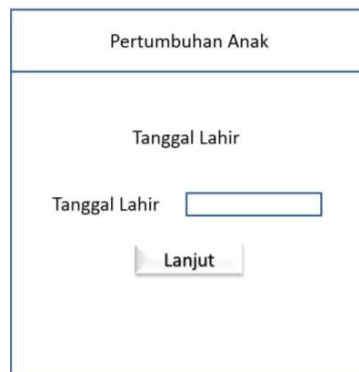
Nama Anak

Mulai

Data Pengguna

Ganti Basis Data

Gambar 3.20 Halaman *home* (admin)
(Sumber: Data Penelitian, 2016)



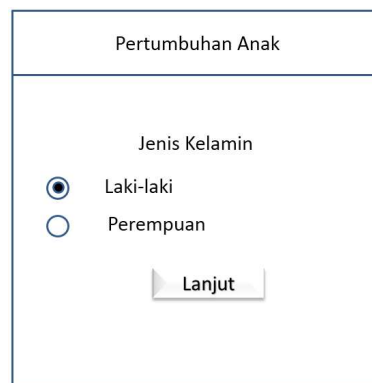
Pertumbuhan Anak

Tanggal Lahir

Tanggal Lahir

Lanjut

Gambar 3.21 Halaman tanggal lahir
(Sumber: Data Penelitian, 2016)



Pertumbuhan Anak

Jenis Kelamin

Laki-laki

Perempuan

Lanjut

Gambar 3.22 Halaman jenis kelamin
(Sumber: Data Penelitian, 2016)

Pertumbuhan Anak
<p>Berat Badan (kg)</p> <p>Berat Badan <input type="text"/></p> <p><input type="button" value="Lanjut"/></p>

Gambar 3.23 Halaman berat badan
(Sumber: Data Penelitian, 2016)

Pertumbuhan Anak
<p>Tinggi atau Panjang (cm)</p> <p>Panjang atau Tinggi <input type="text"/></p> <p><input type="button" value="Lanjut"/></p>

Gambar 3.24 Halaman tinggi atau panjang
(Sumber: Data Penelitian, 2016)

Pertumbuhan Anak
<p>Data Pengguna</p> <p>Nama Anak = Kelvin Jenis Kelamin = Laki-laki Tgl Lahir = 2015-06-18 Tinggi (cm) = 70.2 Berat (kg) = 8.6</p> <p><input type="button" value="Kembali"/></p>

Gambar 3.25 Halaman data pengguna
(Sumber: Data Penelitian, 2016)



Pertumbuhan Anak

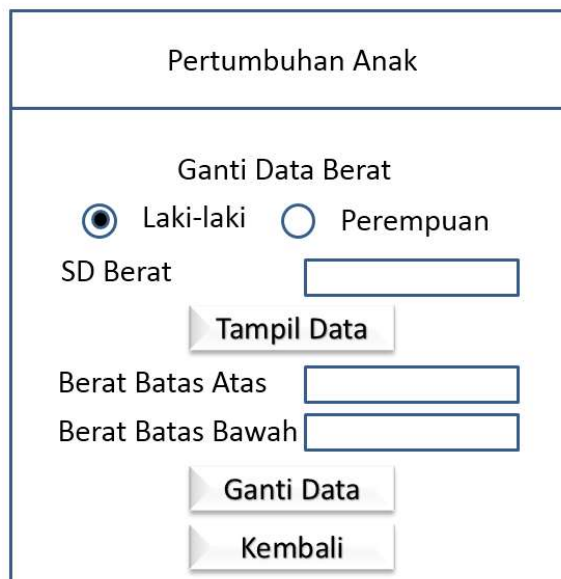
Basis Data

Ganti Data Berat

Ganti Data Tinggi

Kembali

Gambar 3.26 Halaman ganti basis data
(Sumber: Data Penelitian, 2016)



Pertumbuhan Anak

Ganti Data Berat

Laki-laki Perempuan

SD Berat

Tampil Data

Berat Batas Atas

Berat Batas Bawah

Ganti Data

Kembali

Gambar 3.27 Halaman ganti data berat
(Sumber: Data Penelitian, 2016)

3.5 Lokasi dan Jadwal Penelitian

3.5.1 Lokasi Penelitian

Lokasi penelitian dilakukan di Kota Batam.

3.5.2 Jadwal Penelitian

Kegiatan yang dilakukan dan jadwal pelaksanaan penelitian dibuat dalam bentuk matriks adalah sebagai berikut:

Tabel 3.13 Jadwal Penelitian

Kegiatan	Tahun 2016												Tahun 2017						
	September			Oktober			November			Desember			Januari			Februari			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
Pengajuan																			
Judul																			
Bab I																			
Bab II																			
Bab III																			
Bab IV																			
Bab V																			
Penyerahan Softcover, dll.																			