

# **SKRIPSI**

## **APLIKASI SISTEM PAKAR BERBASIS WEB DIAGNOSA PENYAKIT *STUNTING* PADA ANAK DENGAN METODE *FORWARD CHAINING***



**NAMA : BARY DEWANDA PUTRA**

**NIM : 1755201006**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PAHLAWAN TUANKU TAMBUSAI  
RIAU  
2021**

# **SKRIPSI**

## **APLIKASI SISTEM PAKAR BERBASIS WEB DIAGNOSA PENYAKIT *STUNTING* PADA ANAK DENGAN METODE *FORWARD CHAINING***



**NAMA : BARY DEWANDA PUTRA**  
**NIM : 1755201006**

**Diajukan Sebagai Persyaratan untuk Memperoleh Gelar Sarjana S1 Teknik Informatika**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PAHLAWAN TUANKU TAMBUSAI  
RIAU  
2021**

**LEMBAR PERSETUJUAN DEWAN PENGUJI  
UJIAN SKRIPSI S1 TEKNIK INFORMATIKA**

<b>NO</b>	<b>NAMA</b>	<b>TANDA TANGAN</b>
<b>1</b>	<b><u>Prof. Dr. Amir Luthfi</u> Ketua</b>	<b>(.....)</b>
<b>2</b>	<b><u>Novi Yona Sidratul Munti, M.Kom</u> Sekretaris</b>	<b>(.....)</b>
<b>3</b>	<b><u>Emon Azriadi, S.T., M.Sc.E</u> Anggota I</b>	<b>(.....)</b>
<b>4</b>	<b><u>Hanantatur Adeswastoto, S.T., M.T</u> Anggota II</b>	<b>(.....)</b>

**Mahasiswa :**

**Nama : BARY DEWANDA PUTRA**

**NIM : 1755201006**

**Tanggal Ujian : 15 November 2021**

**HALAMAN LEMBAR PERSETUJUAN PEMBIMBING**

**Skripsi yang Berjudul :**

**APLIKASI SISTEM PAKAR BERBASIS WEB  
DIAGNOSA PENYAKIT *STUNTING* PADA ANAK DENGAN METODE  
*FORWARD CHAINING***

Disusun Oleh:

**Nama : Bary Dewanda Putra**  
**NIM : 1755201006**  
**Program Studi : S1 Teknik Informatika**

Bangkinang, 15 November 2021

Disetujui oleh:

**Pembimbing I**

**Pembimbing II**

**Prof. Dr. Amir Luthfi**

**Novi Yona Sidratul Munti, S.Kom., M.Kom**  
**NIP TT. 096 542 170**

Mengetahui,

**Fakultas Teknik**  
**Dekan**

**Program Studi S1 Teknik Informatika**  
**Ketua**

**Emon Azriadi, S.T., M.Sc.E**  
**NIP TT. 096 542 190**

**Deddy Gusman, S.Kom., M.T.I**  
**NIP TT. 096 542 160**

## SURAT PERTANYATAAN

Dengan ini saya menyatakan bahwa :

1. Skripsi saya dengan judul, “**APLIKASI SISTEM PAKAR BERBASIS WEB DIAGNOSA PENYAKIT *STUNTING* PADA ANAK DENGAN METODE *FORWARD CHAINING***” adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik baik di Universitas Pahlawan Tuanku Tambusai maupun di perguruan tinggi lainnya.
2. Skripsi ini murni gagasan, penilaian, dan rumusan saya sendiri, tanpa bantuan tidak sah dari pihak lain, kecuali arahan dari pembimbing.
3. Di dalam Skripsi ini tidak terdapat hasil karya atau pendapat yang telah di tulis atau dipublikasikan orang lain, kecuali dikutip secara tertulis dengan jelas dan dicantumkan sebagai acuan di dalam naskah saya dengan disebutkan nama pengarangnya dan dicantumkan pada daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya, dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran pernyataan ini, saya bersedia menerima sanksi berupa pencabutan gelar yang saya peroleh karena Skripsi ini, serta sanksi lainnya sesuai dengan norma dan ketentuan hukum yang berlaku.

Bangkinang, 15 November 2021  
Saya yang menyatakan

BARY DEWANDA PUTRA  
NIM 1755201006

## PERSEMBAHAN

Alhamdulillah sujud syukur kepada Allah SWT, yang mana atas segala rahmat dan karunia-Nya saya dapat menyelesaikan penulisan skripsi ini. Skripsi ini saya persembahkan untuk ayahanda saya tercinta yaitu Yuswar dan ibunda saya tercinta Hidayati yang telah memberikan kebahagiaan dan cinta didalam kehidupan saya. Saya juga mengucapkan terima kasih kepada adik-adik saya tercinta Rian Marwanda, Melati Eriawanda, Arif Yuwananda yang selalu memberikan semangat kepada saya dalam penulisan skripsi ini.

Saya juga mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Prof. DR. Amir Luthfi dan juga kepada Ibu Novi Yona Sidratul Munti, M.Kom selaku dosen pembimbing saya, dengan penuh kesabaran bapak dan ibu selalu membimbing saya hingga penulisan skripsi saya selesai dengan baik, dan juga untuk semua dosen yang ada di Universitas Pahlawan Tuanku Tambusai saya ucapkan terima kasih banyak karena telah mendidik dan mengajarkan saya selama proses perkuliahan berlangsung.

Tak lupa pula saya ucapkan terima kasih banyak kepada seluruh rekan seperjuangan saya di prodi S1 Teknik Informatika yang sudah memotivasi dan memberikan masukan serta saran kepada saya sehingga saya bisa menyelesaikan skripsi ini hingga selesai dengan baik.

## MOTTO

*“Tanpa membaca dan belajar kita akan menjadi orang kelas teri, perundung dan pemaki, mudah diprovokasi tanpa keluasan hati dan imajinasi”*

Najwa Shihab

## **APPLICATION OF EXPERT SYSTEMS WEB-BASED DIAGNOSIS OF STUNTING IN CHILDHOOD WITH FORWARD CHAINING METHOD**

### **ABSTRACT**

*Expert system is a system that seeks to adopt human knowledge to computers so that computers can solve problems as experts usually do. The Expert System asks for facts that will show symptoms of a particular disease and can provide an explanation of the results of the consultations that have been carried out. In diagnosis, an expert faces a problem including the answers found in the form of uncertain answers. The limited presence of an expert makes it difficult for some people to take steps or actions when exposed to disease due to lack of knowledge. The presence of the Expert System is made to replace the existence of an expert who is indirectly in the community.*

*This system is based on website technology and is built using the PHP Native programming language and uses MySQL database storage. With the waterfall methodology used in system development, the stages are more structured and clear. The expert system created can produce a process of diagnosing stunting and other malnutrition in humans by using the Forward Chaining method. So that users are able to obtain information about stunting and other malnutrition.*

*It is necessary to add data for other types of malnutrition and their clinical symptoms so that the results of the information they have will be more precise and accurate. The website-based Expert System created is able to help users to diagnose diseases and obtain information about stunting and other malnutrition, because this system has a reading list or article in its features.*

*Keywords: Expert System, Stunting, Malnutrition Disease.*

# **APLIKASI SISTEM PAKAR BERBASIS WEB DIAGNOSA PENYAKIT *STUNTING* PADA ANAK DENGAN METODE *FORWARD CHAINING***

## **ABSTRAK**

Sistem Pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem Pakar menanyakan fakta - fakta yang akan menunjukkan gejala penyakit tertentu dan dapat memberikan penjelasan atas hasil konsultasi yang telah dilakukan. Dalam diagnosa, seorang pakar menghadapi suatu permasalahan diantaranya jawaban yang ditemukan berupa jawaban belum pasti. Terbatasnya keberadaan seorang pakar membuat beberapa masyarakat kesulitan dalam mengambil langkah atau tindakan saat terkena penyakit karena kurangnya pengetahuan yang dimiliki. Hadirnya Sistem Pakar dibuat untuk menggantikan keberadaan seorang pakar yang secara tidak langsung berada ditengah masyarakat.

Sistem ini berbasis teknologi *website* dan dibangun dengan menggunakan bahasa pemrograman *PHP Native* dan menggunakan penyimpanan *database MySQL*. Dengan metodologi *waterfall* yang digunakan dalam pengembangan sistem, menghasilkan tahap-tahap yang lebih terstruktur dan jelas. Sistem Pakar yang dibuat dapat menghasilkan proses diagnosis penyakit *stunting* dan kurang gizi lainnya pada manusia dengan menggunakan metode *Forward Chaining*. Sehingga pengguna mampu memperoleh informasi mengenai penyakit *stunting* dan kurang gizi lainnya.

Perlu diadakan penambahan data untuk jenis penyakit kurang gizi lainnya beserta gejala-gejala klinisnya sehingga hasil informasi yang dimiliki akan semakin tepat dan akurat. Sistem Pakar berbasis *website* yang dibuat mampu membantu pengguna untuk mendiagnosis penyakit dan memperoleh informasi mengenai penyakit *stunting* dan kurang gizi lainnya, karena sistem ini memiliki daftar bacaan atau artikel didalam fiturnya.

Kata kunci: Sistem Pakar, *Stunting*, Penyakit Kurang Gizi.

## KATA PENGANTAR

Puji dan syukur peneliti ucapkan kepada Allah SWT yang telah memberikan rahmat berupa kesehatan dan kesempatan kepada peneliti sehingga mampu menyelesaikan skripsi ini, yang diajukan guna melengkapi dan memenuhi salah satu syarat dalam menyelesaikan pendidikan pada Universitas Pahlawan Tuanku Tambusai dengan judul **APLIKASI SISTEM PAKAR BERBASIS WEB DIAGNOSA PENYAKIT *STUNTING* PADA ANAK DENGAN METODE *FORWARD CHAINING***.

Dalam penulisan skripsi ini peneliti banyak menghadapi kesulitan, namun berkat bimbingan pengarahan dan bantuan dari semua pihak, skripsi ini dapat diselesaikan. Pada kesempatan ini peneliti mengucapkan terima kasih kepada yang terhormat :

1. Bapak Prof. DR. Amir Luthfi. selaku Rektor Universitas Pahlawan Tuanku Tambusai sekaligus Pembimbing I yang telah memberikan kritik dan saran dalam kesempurnaan skripsi ini.
2. Bapak Emon Azriadi, S.T., M.Sc.E. selaku Dekan Fakultas Teknik Universitas Pahlawan Tuanku Tambusai, sekaligus selaku Penguji I yang banyak memberikan bimbingan dan pengarahan serta dorongan kepada peneliti dalam menyelesaikan skripsi ini.
3. Bapak Deddy Gusman, S.Kom., M.T.I. selaku Ketua Prodi S1 Teknik Informatika Fakultas Teknik Universitas Pahlawan Tuanku Tambusai.

4. Ibu Novi Yona Sidratul Munti, S.Kom., M.Kom. selaku Dosen Pembimbing II yang telah meluangkan waktu, pikiran, serta memberikan arahan dan masukan terhadap penulisan skripsi ini.
5. Bapak Hanantatur Adeswastoto, M.T selaku Penguji II yang banyak memberikan bimbingan dan masukan serta dorongan kepada peneliti dalam menyelesaikan skripsi ini.
6. Ade Purnama Sari, S.Gz. Sebagai Narasumber I yang telah memberikan pengetahuan dan pengalaman mengenai kasus *stunting* serta memberikan kritik dan saran dalam kesempurnaan penyusunan skripsi ini.
7. Faridatul Akmalia, S.Gz. Selaku Narasumber II yang telah memberikan pengetahuan dan pengalaman mengenai kasus *stunting* serta memberikan kritik dan saran dalam penyusunan skripsi ini.
8. Dia Armyati, S.Gz. Selaku Narasumber III yang telah memberikan pengetahuan dan pengalaman mengenai kasus *stunting*.
9. Bapak dan Ibu Dosen Universitas Pahlawan Tuanku Tambusai yang telah memberikan kesempatan dan kemudahan bagi peneliti dalam penyelesaian penyusunan skripsi ini.
10. Kedua orang tua beserta seluruh keluarga peneliti yang telah memberikan dukungan dan motivasi selama penyusunan skripsi ini.
11. Kepada teman seperjuangan yang telah membantu peneliti dalam menyelesaikan skripsi ini.

Peneliti menyadari bahwa skripsi ini jauh dari kesempurnaan dengan segala kekurangannya, untuk itu peneliti mengharapkan kritik dan saran yang membangun dari semua pihak demi kesempurnaan dari skripsi ini.

Bangkinang, 15 November 2021  
Peneliti

BARY DEWANDA PUTRA  
NIM. 1755201006

## DAFTAR ISI

	<b>Halaman</b>
<b>LEMBAR PERSETUJUAN DEWAN PENGUJI .....</b>	<b>ii</b>
<b>HALAMAN LEMBAR PERSETUJUAN PEMBIMBING .....</b>	<b>iii</b>
<b>SURAT PERTANYAAN.....</b>	<b>iv</b>
<b>PERSEMBAHAN.....</b>	<b>v</b>
<b><i>ABSTRACT</i> .....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>DAFTAR GAMBAR.....</b>	<b>xv</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
A. Latar Belakang .....	1
B. Batasan Masalah.....	3
C. Perumusan Masalah.....	3
D. Tujuan Penelitian.....	3
E. Manfaat Penelitian.....	3
<b>BAB II LANDASAN TEORI .....</b>	<b>5</b>
A. Sistem Pakar .....	5
1. Konsep Umum Sistem Pakar .....	8
2. Arsitektur Sistem Pakar .....	10
3. Ciri - Ciri Sistem Pakar .....	13
4. Manfaat dan Kekurangan Sistem Pakar .....	13
5. Klasifikasi Sistem Pakar .....	15
6. Teknik Inferensi .....	16
7. Karakteristik Sistem Pakar .....	18
B. <i>Website</i> .....	19
C. Basis Data ( <i>Database</i> ).....	23
1. Pengertian Data .....	24

2.	Keuntungan dan Kerugian Penggunaan Basis Data .....	31
3.	Perbedaan TFM dengan DBMS .....	32
4.	Istilah - Istilah yang dipergunakan pada Sistem Basis Data.....	32
5.	<i>Database Management System (DBMS)</i> .....	34
6.	<i>Data Independence</i> .....	35
D.	Pemograman <i>Web</i> .....	36
1.	Aplikasi Berbasis <i>Web</i> .....	36
2.	Arsitektur Aplikasi Berbasis <i>Web</i> .....	38
3.	HTML, CSS, dan <i>Javascript</i> .....	40
4.	PHP.....	42
5.	XAMPP .....	47
6.	Rekayasa Perangkat Lunak (RPL).....	51
E.	UML ( <i>Unified Modeling Language</i> ).....	54
1.	<i>Use Case Diagram</i> .....	57
2.	<i>Class Diagram</i> .....	59
3.	<i>Activity Diagram</i> .....	60
4.	<i>Sequence Diagram</i> .....	61
F.	<i>Stunting</i> (Pendek) .....	62
1.	Faktor - Faktor yang Mempengaruhi <i>Stunting</i> .....	63
2.	Ciri - Ciri dan Gejala <i>Stunting</i> .....	64
3.	Cara Penilaian Status Gizi Balita .....	70
4.	Cara Penilaian Status Gizi anak Umur 5-18 Tahun.....	73
5.	Cara Penilaian Status Gizi Dewasa (>18 tahun) .....	74
<b>BAB III METODE PENELITIAN .....</b>		<b>76</b>
A.	Instrumen Penelitian.....	76
1.	Tempat Penelitian.....	76
2.	Alat Penelitian .....	76
B.	Teknik Pengumpulan Data.....	77
1.	Wawancara .....	77
2.	Dokumen atau Studi Pustaka.....	78
C.	Jenis Data .....	78

1. Data Primer.....	78
2. Data Sekunder .....	79
D. Teknik Pengembangan Sistem .....	79
E. Model Penelitian .....	81
F. Data Gejala <i>Stunting</i> .....	82
<b>BAB IV PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>83</b>
A. Analisis Sistem.....	83
B. Pengumpulan Data .....	83
1. Data Gejala .....	83
2. Data penyakit.....	84
C. Desain Sistem.....	86
1. Perancangan UML ( <i>Unified Modelling Language</i> ).....	86
2. Perancangan <i>Interface Web</i> .....	94
3. Perancangan <i>Database</i> .....	101
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>105</b>
A. Kesimpulan.....	105
B. Saran .....	105
<b>DAFTAR PUSTAKA .....</b>	<b>107</b>

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 2. 1 Hubungan antara pengguna dan fungsi sistem pakar.....	13
Tabel 2. 2 Simbol pada <i>Use Case Diagram</i> .....	58
Tabel 2. 3 Simbol - Simbol yang ada pada <i>Class Diagram</i> .....	59
Tabel 2. 4 Simbol - Simbol <i>Activity Diagram</i> .....	60
Tabel 2. 5 Simbol - Simbol dalam <i>Sequence Diagram</i> .....	61
Tabel 3. 1 Data Gejala <i>Stunting</i> .....	82
Tabel 4. 1 Kode data dan data gejala penyakit <i>stunting</i> .....	84
Tabel 4. 2 Data penyakit <i>stunting</i> .....	85
Tabel 4. 3 Tabel basis pengetahuan penyakit <i>stunting</i> pada anak.....	85
Tabel 4. 4 Penjelasan perancangan <i>Use Case Diagram Admin</i> .....	87
Tabel 4. 5 Penjelasan perancangan <i>Use Case Diagram User</i> .....	88
Tabel 4. 6 Tabel <i>database Admin</i> .....	102
Tabel 4. 7 Tabel <i>database</i> analisa hasil .....	102
Tabel 4. 8 Tabel <i>database</i> data artikel .....	102
Tabel 4. 9 Tabel <i>database</i> buku tamu .....	103
Tabel 4. 10 Tabel <i>database</i> gejala.....	103
Tabel 4. 11 Tabel <i>database rules</i> .....	104
Tabel 4. 12 Tabel <i>database</i> solusi.....	104

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar 2. 1 Area dari <i>Artificial Intelligence (AI)</i> .....	6
Gambar 2. 2 Konsep dasar fungsi Sistem Pakar Berbasis Pengetahuan .....	7
Gambar 2. 3 Hubungan <i>Problem Domain</i> dan <i>Knowledge Domain</i> .....	8
Gambar 2. 4 Pengembangan Sistem Pakar .....	9
Gambar 2. 5 Arsitektur Sistem Pakar .....	11
Gambar 2. 6 Proses Inferensi <i>Forward Chaining</i> .....	17
Gambar 2. 7 10 Negara Tertinggi Pengguna Internet di Asia .....	20
Gambar 2. 8 Penyusun Sistem Basis Data .....	24
Gambar 2. 9 Data yang didapat Dari Fakta Pada Transaksi .....	26
Gambar 2. 10 Hubungan Data dengan Basis Data .....	26
Gambar 2. 11 Cara Kerja Model <i>Client-Server</i> .....	39
Gambar 2. 12 Arsitektur Aplikasi Berbasis <i>Web</i> Dinamis .....	40
Gambar 2. 13 Tren Penggunaan PHP oleh <i>Netcraft</i> .....	44
Gambar 2. 14 Halaman <i>website Apache Friends</i> .....	48
Gambar 2. 15 Halaman <i>setup</i> pertama .....	48
Gambar 2. 16 Halaman <i>select components</i> dari XAMPP .....	49
Gambar 2. 17 Halaman <i>installation folder</i> XAMPP .....	49
Gambar 2. 18 Halaman <i>Learn more about Bitnami for XAMPP</i> .....	50
Gambar 2. 19 Halaman proses instalasi XAMPP .....	50
Gambar 2. 20 Halaman notifikasi <i>completing XAMPP setup</i> .....	51
Gambar 2. 21 Tampilan XAMPP <i>Control Panel v3.2.3</i> .....	51
Gambar 2. 22 Hubungan Unsur RPL dengan RPL .....	53
Gambar 2. 23 Diagram UML .....	55
Gambar 2. 24 Anak kelas 4 SD dengan tinggi yang berbeda .....	62
Gambar 2. 25 Peta <i>Stunting</i> dunia, tahun 2007 – 2011 .....	67
Gambar 2. 26 Perbandingan prevalensi <i>stunting</i> antar beberapa negara ASEAN .....	68
Gambar 2. 27 Kerangka logis masalah gizi .....	68
Gambar 2. 28 Kerangka pembahasan <i>stunting</i> di Indonesia .....	69
Gambar 3. 1 <i>Waterfall</i> Model .....	80

Gambar 4. 1 Perancangan <i>Use Case Diagram</i> .....	87
Gambar 4. 2 Perancangan <i>Class Diagram</i> .....	90
Gambar 4. 3 Perancangan <i>Activity Diagram Admin</i> .....	91
Gambar 4. 4 Perancangan <i>Activity Diagram User</i> .....	91
Gambar 4. 5 Perancangan <i>Sequence Diagram Login Admin</i> .....	92
Gambar 4. 6 Perancangan <i>Sequence Diagram Input Data Admin</i> .....	93
Gambar 4. 7 Perancangan <i>Sequence Diagram Edit Data Admin</i> .....	93
Gambar 4. 8 Perancangan <i>Sequence Diagram Konsultasi User</i> .....	93
Gambar 4. 9 Halaman utama <i>Admin</i> .....	94
Gambar 4. 10 Halaman artikel kesehatan .....	94
Gambar 4. 11 Halaman <i>form input</i> artikel kesehatan.....	95
Gambar 4. 12 Halaman data gejala .....	95
Gambar 4. 13 Halaman <i>form input</i> data gejala .....	95
Gambar 4. 14 Halaman data solusi .....	96
Gambar 4. 15 Halaman <i>form input</i> data solusi.....	96
Gambar 4. 16 Halaman data <i>rules</i> .....	96
Gambar 4. 17 Halaman untuk menentukan aturan <i>rules</i> .....	97
Gambar 4. 18 Halaman data buku tamu.....	97
Gambar 4. 19 Halaman data <i>Admin</i> .....	97
Gambar 4. 20 Halaman laporan hasil diagnosa.....	98
Gambar 4. 21 Halaman <i>Login Admin</i> .....	98
Gambar 4. 22 Halaman utama <i>User</i> .....	98
Gambar 4. 23 Halaman profil sistem pada <i>User</i> .....	99
Gambar 4. 24 Halaman <i>form input</i> konsultasi <i>User</i> .....	99
Gambar 4. 25 Halaman artikel kesehatan pada <i>User</i> .....	99
Gambar 4. 26 Halaman buku tamu <i>User</i> .....	100
Gambar 4. 27 Halaman informasi kontak <i>Admin</i> .....	100
Gambar 4. 28 Halaman proses diagnosa penyakit .....	100
Gambar 4. 29 Halaman hasil diagnosa penyakit.....	101
Gambar 4. 30 Halaman cetak laporan diagnosa pasien.....	101

## **DAFTAR LAMPIRAN**

- Lampiran 1 : Biodata Diri
- Lampiran 2 : Surat Permohonan
- Lampiran 3 : Lembar Persetujuan Narasumber 1
- Lampiran 4 : Lembar Persetujuan Narasumber 2
- Lampiran 5 : Lembar Persetujuan Narasumber 3
- Lampiran 6 : Lembar Konsultasi
- Lampiran 7 : Basis Data Sistem
- Lampiran 8 : Kodingan
- Lampiran 9 : Lembar Wawancara
- Lampiran 10 : Foto Penelitian
- Lampiran 11 : Lembar Angket

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Kecerdasan buatan (*Artificial Intelligence*) merupakan bagian dari ilmu komputer sehingga komputer dapat melakukan pekerjaan seperti yang dilakukan manusia. Sistem Pakar merupakan salah satu dari bagian kecerdasan buatan yang membuat agar mesin (komputer) dapat melakukan pekerjaannya seperti dan sebaik yang dilakukan oleh manusia. Sistem Pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem Pakar menanyakan fakta - fakta yang akan menunjukkan gejala penyakit tertentu dan dapat memberikan penjelasan atas hasil konsultasi yang telah dilakukan. Dalam diagnosa, seorang pakar menghadapi suatu permasalahan diantaranya jawaban yang ditemukan berupa jawaban belum pasti.

Terbatasnya keberadaan seorang pakar membuat beberapa masyarakat kesulitan dalam mengambil langkah atau tindakan saat terkena penyakit karena kurangnya pengetahuan yang dimiliki. Hadirnya Sistem Pakar dibuat untuk menggantikan keberadaan seorang pakar yang secara tidak langsung berada ditengah masyarakat. Hal ini membuat masyarakat bisa mengetahui dan mengambil tindakan pertolongan pertama jika terkena suatu penyakit. Masyarakat harus memperhatikan gejala - gejala penyakit yang muncul agar sistem bisa membaca kondisi seorang pasien dengan benar.

Pada proses tumbuh kembang seorang anak, banyak masyarakat yang menganggap bahwa beberapa masalah yang terjadi merupakan hal yang lumrah atau hal yang biasa. Padahal banyak hal yang berpotensi berakibat fatal apabila ada hal yang memperlambat pertumbuhan seorang anak dan dibiarkan begitu saja. Sistem ini dibuat agar masyarakat lebih mengetahui dan mewaspadai masalah - masalah yang mempengaruhi proses pertumbuhan anak.

Salah satunya adalah penyakit *Stunting* yang bisa menyerang anak dan bisa berakibat permanen jika tidak ditangani lebih lanjut. *Stunting* adalah kondisi ketika anak lebih pendek dibandingkan anak - anak seusianya, atau dengan kata lain, tinggi badan anak berada dibawah standar. Standar yang dipakai sebagai acuan adalah kurva pertumbuhan yang dibuat oleh Badan Kesehatan Dunia (*WHO*).

Pada penelitian ini lebih mengarah kedalam penggunaan Pemrograman *Website* sebagai media untuk mengimplementasikan aplikasi Sistem Pakar. Saat ini masyarakat sangat mudah untuk melakukan berbagai macam aktivitas ataupun untuk mendapatkan suatu informasi. Salah satu contoh yang akan penulis rancang adalah cara mengetahui dan mendeteksi seorang anak mengalami penyakit *stunting* dengan menggunakan Sistem Pakar. Dengan adanya sistem ini, maka seseorang bisa mengetahui langsung informasi awal terkait penyakit tersebut tanpa harus menemui seorang ahli gizi.

## **B. Batasan Masalah**

Dalam penelitian ini, pembahasan dibatasi pada perancangan dan simulasi penerapan implementasi aplikasi Sistem Pakar diagnosa penyakit *stunting* pada anak dan beberapa penyakit anak lainnya.

## **C. Perumusan Masalah**

1. Bagaimana peranan dan efektivitas Sistem Pakar penyakit *stunting* pada anak terhadap masyarakat yang menggunakan sistem ini?
2. Apakah aplikasi Sistem Pakar ini meningkatkan kesadaran masyarakat tentang penyakit *stunting* bahwa penyakit ini merupakan salah satu penyakit yang bisa berdampak permanen atau selamanya?

## **D. Tujuan Penelitian**

1. Mendiagnosa gejala - gejala awal suatu penyakit dengan menggunakan Sistem Pakar sebagai media untuk mengambil tindakan pertama dalam mencegah ataupun mengobati penyakit tersebut.
2. Meningkatkan kesadaran masyarakat tentang penyakit *stunting* bahwa penyakit ini berdampak hingga anak tumbuh dewasa dengan cara memberikan edukasi melalui teknologi informatika.
3. Mengimplementasikan aplikasi Sistem Pakar tentang diagnosa penyakit *stunting*.

## **E. Manfaat Penelitian**

1. Menerapkan implementasi ilmu pengetahuan yang telah di dapatkan selama perkuliahan untuk menunjukkan kemampuan dalam melakukan perancangan aplikasi sistem berbasis *website*.

2. Mendiagnosa penyakit *stunting* yang dialami masyarakat melalui aplikasi Sistem Pakar.
3. Memberikan edukasi mengenai penyakit *stunting* dan penyakit kekurangan gizi pada anak lainnya melalui aplikasi Sistem Pakar.

## **BAB II**

### **LANDASAN TEORI**

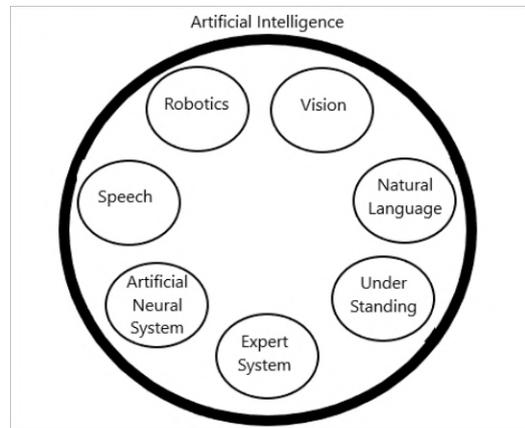
#### **A. Sistem Pakar**

Sistem Pakar merupakan sistem yang berada dalam pemrograman komputer yang menggunakan basis pengetahuan dan prosedur inferensi sebagai untuk menyelesaikan masalah yang membutuhkan ahli atau pakar untuk menyelesaikannya. Menurut Gede & Divayana (dalam Ramadhan et al., 2018) mengatakan bahwa Sistem Pakar termasuk kedalam kelompok kecerdasan buatan yang mempunyai kemampuan untuk menyelesaikan kondisi permasalahan yang ada. Al Ajlan (dalam Ramadhan et al., 2018) menyampaikan bahwa dalam penelitian selanjutnya dikemukakan bahwa Sistem Pakar ialah pengembangan dari sistem cerdas berbasis aplikasi.

Pendapat lainnya yang mengemukakan, Hossain et al., (dalam Ramadhan et al., 2018) Sistem Pakar ialah hasil dari pengetahuan pakar dan teknik pencarian. Sistem Pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem Pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah. Bidang Sistem Pakar merupakan penyelesaian pendekatan yang sangat bagus dan berhasil untuk permasalahan AI (*Artificial Intelligence*) klasik dari pemograman cerdas (Rosnelly, 2012).

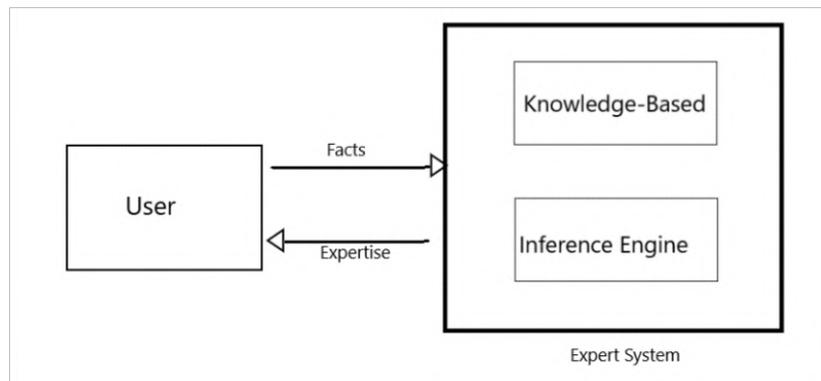
Profesor Edward Feigenbaum dari *Stanford University* yang merupakan penggagas dalam teknologi Sistem Pakar mendefenisikan Sistem Pakar sebagai sebuah program pintar dalam komputer yang memanfaatkan pengetahuan dan

prosedur inferensi untuk memecahkan masalah yang cukup sulit sehingga membutuhkan keahlian khusus dari manusia atau ahli.



**Gambar 2. 1** Area dari Artificial Intelligence (AI)

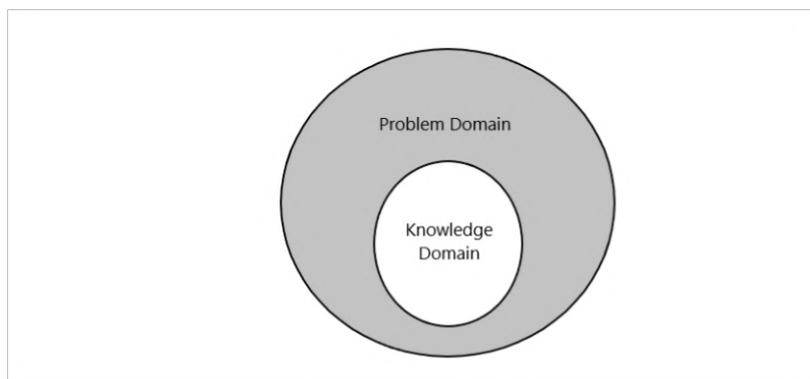
Pakar atau ahli merupakan orang yang memiliki pengetahuan dan keahlian yang spesifik dan tidak dimiliki oleh kebanyakan orang. Seorang pakar mampu memecahkan masalah yang tidak dapat dipecahkan oleh kebanyakan orang. Dengan kata lain, dapat memecahkan suatu masalah dengan lebih efisien dan lebih efektif. Pengetahuan yang dimuat ke dalam Sistem Pakar dapat berasal dari seorang pakar atau pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasikan lainnya, serta orang yang memiliki pengetahuan meskipun bukan ahli (Rosnelly, 2012). Istilah Sistem Pakar sering juga disebut sebagai sistem berbasis pengetahuan (*knowledge-based system*) atau Sistem Pakar berbasis pengetahuan (*knowledge based expert system*).



**Gambar 2. 2 Konsep dasar fungsi Sistem Pakar Berbasis Pengetahuan**

Pada Gambar 2.2 mengilustrasikan konsep dasar Sistem Pakar berbasis pengetahuan (*knowledge based expert system*). *User* memberikan informasi atau data kepada sistem dan menerima respon berupa saran ahli (*expertise*). Sistem terdiri dari dua komponen utama yaitu basis pengetahuan yang berisi tentang pengetahuan yang digunakan oleh komponen lain yaitu mesin inferensi untuk menghasilkan kesimpulan atau hasil sebagai respon terhadap masukan yang di *input* oleh *user*.

Pengetahuan yang dimiliki pakar bersifat spesifik dalam satu area masalah (*problem domain*). Area masalah tersebut merupakan satu wilayah yang spesifik seperti pengobatan, keuangan, rekayasa dan lainnya. Pengetahuan untuk memecahkan suatu masalah dikenal dengan area pengetahuan (*knowledge domain*) (Rosnelly, 2012). Gambar 2.3 berikut mengilustrasikan hubungan antara area *problem* dengan area *knowledge*. Area *knowledge* berada dalam area *problem*. Bagian yang berada diluar area pengetahuan menyatakan masalah tidak dimiliki oleh sistem. Sebuah Sistem Pakar tidak memiliki pengetahuan lain diluar area pengetahuannya kecuali jika diprogram dan dimuat kedalam sistem tersebut.



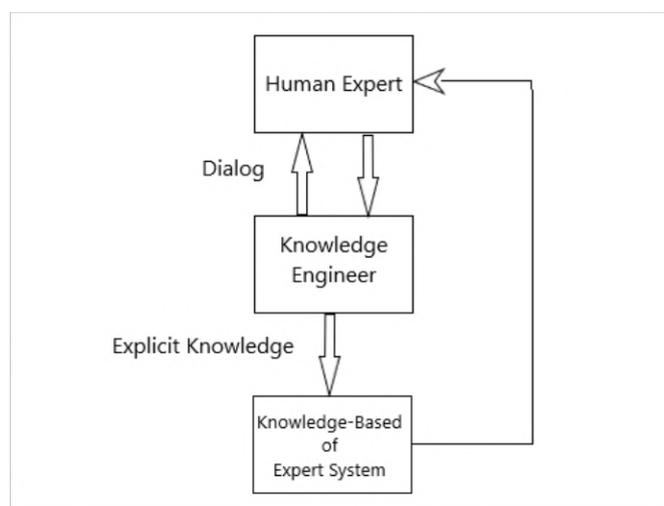
Gambar 2. 3 Hubungan Problem Domain dan Knowledge Domain

## 1. Konsep Umum Sistem Pakar

Dalam Sistem Pakar terdapat 2 jenis penalaran, diantaranya *Rule Base Reasoning* dan *Case Base Reasoning*. Menurut Ramadhan & Pane (2018) *Rule Base Reasoning* merupakan bentuk penalaran yang menggunakan konsep aturan - aturan, metode yang termasuk diantaranya: *Certainty Factor*, *Teorema Bayes*, *Dempster Shafer*, *Euclidean Probability*. Sementara itu untuk *Case Base Reasoning*, penalaran ini merupakan bentuk yang menggunakan teknik kemiripan antara kasus baru dengan kasus sebelumnya, metode yang termasuk didalamnya adalah *K-Nearest Neighbor*, *Manhatan Distance* dan *Minkowski Distance*. Dalam pengembangannya Sistem Pakar juga mengadopsi logika *fuzzy* yang dikenal sebagai metode yang digunakan dalam masalah prediksi atau ketidakpastian.

Pengetahuan Sistem Pakar direpresentasikan dalam beberapa cara. Metode *IF THEN rules* merupakan cara yang paling umum digunakan. Rosnelly (2012) berpendapat bahwa pengetahuan tidak tertulis yang dimiliki oleh seorang pakar harus di ekstraksi melalui wawancara secara

ekstensif oleh *knowledge engineer*. Proses pengembangan Sistem Pakar dengan perolehan pengetahuan dari pakar maupun sumber lain dan kodingnya disebut sebagai *knowledge engineering* yang dilaksanakan oleh *knowledge engineer*. Gambar 2.4 dibawah ini mengilustrasikan tahapan pengembangan Sistem Pakar secara umum.



**Gambar 2. 4 Pengembangan Sistem Pakar**

Proses awal, *knowledge engineer* melakukan diskusi dengan pakar untuk mengumpulkan pengetahuan dari pakar tersebut. Proses ini serupa dengan proses diskusi persyaratan atau kebutuhan yang dilakukan *system engineer* pada sistem konvensional dengan kliennya. Setelah itu *knowledge engineer* membangun pengetahuan secara eksplisit kedalam *knowledge base*. Pakar kemudian mengevaluasi sistem dan memberikan kritik. Proses ini berlangsung secara iteratif hingga dinilai sesuai oleh pakar (Rosnelly, 2012).

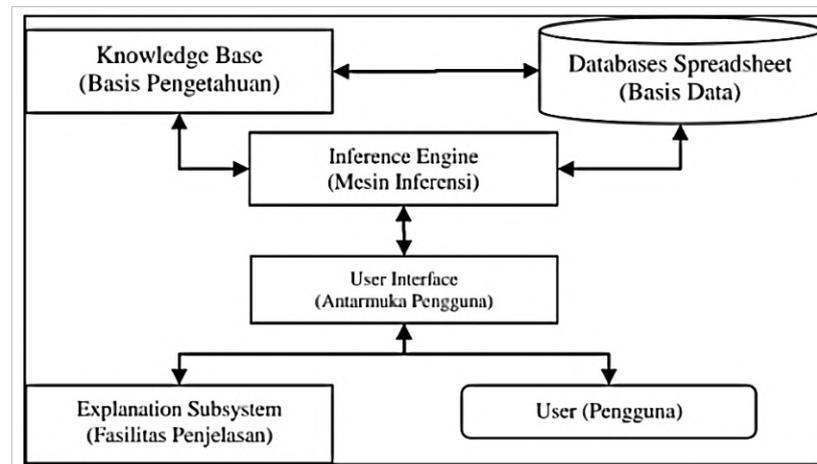
Pengetahuan yang belum pernah tersusun secara sistematis akan sulit untuk diformalisasikan. Sering terjadi masalah dalam penerapannya seperti

inkonsistensi, ambiguitas, duplikasi. Seorang pakar mengetahui batas pengetahuan yang dimiliki dan membatasi saran yang diberikan.

Menurut Rosnelly (2012) berpendapat bahwa keterbatasan Sistem Pakar dalam prakteknya saat ini adalah kurangnya pengetahuan kausal (*causal knowledge*) yang berarti sistem tidak memiliki pemahaman yang melandasi sebab dan efek dalam sistem. Membangun Sistem Pakar dengan pengetahuan bersifat dangkal (*shallow knowledge*) yang didasarkan atas pengetahuan empirik dan heuristik lebih mudah dibandingkan dengan memprogram sistem dengan pengetahuan mendalam (*deep knowledge*) yang berdasarkan pada struktur, fungsi dan perilaku sebuah objek. Masalah lain dalam Sistem Pakar adalah keterbatasan pengetahuan. Sebuah Sistem Pakar tidak dapat melakukan analogi terhadap sebuah masalah baru seperti yang dilakukan manusia meskipun dapat diterapkan aturan induksi, namun hanya berlaku pada beberapa tipe pengetahuan saja.

## **2. Arsitektur Sistem Pakar**

Menurut Ramadhan & Pane (2018) Sistem Pakar disusun oleh 6 bagian utama, yaitu: basis pengetahuan (*knowledge base*), motor inferensi (*inference engine*), basis data (*database spreadsheet*), antarmuka pengguna (*user interface*), fasilitas penjelasan (*explanation subsystem*), dan pengguna (*user*).



Gambar 2. 5 Arsitektur Sistem Pakar

Berikut ini penjelasan tentang komponen - komponen arsitektur sistem pakar menurut Ramadhan,dkk (2018) yaitu:

**a. Basis Pengetahuan (*Knowledge Base*)**

Basis Pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:

- 1) Fakta, misalnya situasi, kondisi, atau permasalahan yang ada,
- 2) Aturan (*rule*), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

**b. Basis Data (*Databases Spreadsheet*)**

Digunakan sebagai media yang berfungsi untuk menampung fakta - fakta, kondisi yang diperoleh dari basis pengetahuan untuk disimpan dan diproses oleh komputer.

**c. Mesin Inferensi (*Inference Engine*)**

Mesin Inferensi 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 pengendalian 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.

**d. Antarmuka Pengguna (*User Interface*)**

Digunakan sebagai media komunikasi antara pengguna dan Sistem Pakar. Komunikasi ini paling bagus jika dalam bahasa alami dan dilengkapi dengan *graphic*, *menu*, dan formulir elektronik. Pada bagian ini akan terjadi dialog antara Sistem Pakar dan pengguna.

**e. Fasilitas Penjelasan (*Explanation Subsystem*)**

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar dan pemecahan masalah.

**f. Pengguna (*User*)**

Pada umumnya pengguna Sistem Pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi atau saran dari berbagai permasalahan yang ada.

Tabel 2. 1 Hubungan antara pengguna dan fungsi Sistem Pakar

Pengguna	Kepentingan	Fungsi Sistem Pakar
Klien bukan pakar	Mencari saran/nasehat	Konsultan atau penasihat
Mahasiswa	Belajar	Instruktur
Pembangun Sistem	Memperbaiki/menambah Basis pengetahuan	Rekan ( <i>partner</i> )
Pakar	Membantu analisis rutin atau proses komputasi, mencari (mengklasifikasi) informasi, alat bantu diagnosa	Rekan kerja atau asisten

Sumber : Rika Rosnelly, 2012

### 3. Ciri - Ciri Sistem Pakar

Ada beberapa ciri - ciri dari Sistem Pakar (Hayadi, 2018), diantaranya:

- a. Terbatas pada *domain* keahlian tertentu,
- b. Dapat memberikan penalaran untuk data yang tidak pasti,
- c. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami,
- d. Berdasarkan pada kaidah atau *rule* tertentu,
- e. Dirancang untuk dapat dikembangkan secara bertahap,
- f. Pengetahuan dan mekanisme inferensi jelas terpisah,
- g. Keluarannya bersifat anjuran,
- h. Sistem dapat mengaktifkan kaidah secara searah yang sesuai yang dituntun oleh dialog dengan pemakai.

### 4. Manfaat dan Kekurangan Sistem Pakar

#### a. Manfaat Sistem Pakar

Menurut T. Sutojo (dalam Hayadi, 2018) Sistem Pakar menjadi populer karena banyak manfaat dan kemampuannya, diantaranya:

- 1) Meningkatkan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat daripada manusia,
- 2) Membuat seseorang yang tidak biasa bekerja seperti layaknya seorang pakar,
- 3) Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan,
- 4) Mampu menangkap pengetahuan dan kepakaran seseorang,
- 5) Memudahkan akses pengetahuan seorang pakar,
- 6) Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan Sistem Pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru,
- 7) Meningkatkan kemampuan untuk menyelesaikan masalah karena Sistem Pakar mengambil sumber pengetahuan dari banyak pakar.

**b. Kekurangan Sistem Pakar**

Sistem Pakar juga memiliki kekurangan dalam penggunaannya, diantaranya:

- 1) Membutuhkan biaya yang relatif mahal untuk membuat dan memeliharanya,
- 2) Keterbatasan seorang pakar membuat sistem sulit dikembangkan,
- 3) Akurasi Sistem Pakar tidak bisa mencapai angka 100% benar.

## 5. Klasifikasi Sistem Pakar

Siswanto (dalam Hayadi, 2018) berpendapat bahwa Sistem Pakar memiliki beberapa klasifikasi, yaitu:

### a. **Diagnosis:**

- 1) Digunakan untuk merekomendasikan: Obat untuk orang sakit, kerusakan mesin, kerusakan rangkaian elektronik,
- 2) Menemukan masalah atau kerusakan yang terjadi,
- 3) Menggunakan pohon keputusan (*decision tree*) sebagai representasi pengetahuannya.

### b. **Pengajaran:**

- 1) Digunakan untuk pengajaran, mulai dari SD sampai dengan PT,
- 2) Membuat diagnosa penyebab kekurangannya dari siswa, kemudian memberikan cara untuk memperbaikinya.

### c. **Interpretasi:**

Untuk menganalisa data yang tidak lengkap, tidak teratur, dan data yang kontradiktif. Misalnya: untuk interpretasi citra.

### d. **Prediksi:**

- 1) Contoh: bagaimana seorang pakar meteorologi memprediksi cuaca besok berdasarkan data - data sebelumnya,
- 2) Untuk peramalan cuaca,
- 3) Penentuan masa tanam.

**e. Perencanaan:**

- 1) Mulai dari perencanaan mesin - mesin sampai dengan manajemen bisnis,
- 2) Untuk menghemat biaya, waktu dan material, sebab pembuatan model,
- 3) Sudah tidak diperlukan,
- 4) Contoh: Sistem konfigurasi komputer.

**f. Kontrol:**

- 1) Digunakan untuk mengontrol kegiatan yang membutuhkan presisi waktu tinggi,
- 2) Misal: Pengontrolan pada industri - industri berteknologi tinggi.

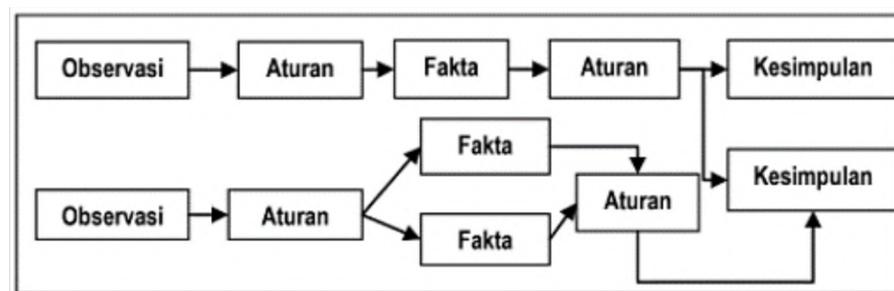
**6. Teknik Inferensi**

Nurajizah & Saputra (dalam Ramadhan et al., 2018) berpendapat bahwa Inferensi merupakan sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada mesin inferensi yang mencakup prosedur - prosedur mengenai pemecahan masalah. Mesin inferensi memiliki tugas yaitu menarik kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

Terdapat dua metode umum dalam melakukan inferensi yaitu, *forward chaining* dan *backward chaining*. Menurut Munti & Effindri (2017), *Forward chaining* adalah suatu metode pengambilan keputusan yang umum digunakan dalam Sistem Pakar. Proses pencarian pada metode ini bergerak dari kiri kekanan, yaitu dari premis menuju kepada kesimpulan akhir. Menurut

Rosnelly (2012) *Backward chaining* melibatkan proses penalaran terbalik dari hipotesis yang merupakan kesimpulan yang akan dibuktikan, ke fakta yang mendukung hipotesis.

Menurut (Ramadhan & Pane, 2018) *Forward chaining* adalah teknik pencarian yang mencocokkan fakta - fakta yang ada dengan bagian *IF* dari *rule 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 kedalam *database*. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali. Proses pencocokan berhenti apabila tidak ada lagi *rule* yang bisa dieksekusi.



Gambar 2. 6 Proses Inferensi *Forward Chaining*

Ramadhan & Pane (2018) berpendapat bahwa *Backward chaining* adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari hasil (*Goal*) yaitu: Bagian yang berada di *THEN* dari *rule IF-THEN*, kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta - fakta yang ada cocok dengan premis - premis di bagian *IF*. Proses ini akan berakhir jika fakta bisa membuktikan kebenaran dari hasil (*Goal*) yang ada.

Ada tiga macam penelusuran yang mempengaruhi kedua metode tersebut, yaitu *Depth First Search*, *Breadth First Search* dan *Best First Search* (Rosnelly, 2012).

- a. *Depth-first search*, melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan.
- b. *Breadth first search*, bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya.
- c. *Best first search*, bekerja berdasarkan kombinasi kedua metode sebelumnya.

## 7. Karakteristik Sistem Pakar

Rosnelly (2012) mengatakan bahwa Sistem Pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini:

- a. Kinerja sangat baik (*high performance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.
- b. Waktu respon yang baik (*adequate respon time*). Sistem juga harus mampu bekerja dalam waktu yang sama baiknya (*reasonable*) atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan. Hal ini sangat penting terutama pada sistem waktu nyata (*real-time*).
- c. Dapat diandalkan (*good reliability*). Sistem harus dapat diandalkan dan tidak mudah rusak atau *crash*.

d. Dapat dipahami (*understandable*). Sistem harus mampu menjelaskan langkah - langkah penalaran yang dilakukannya seperti seorang pakar.

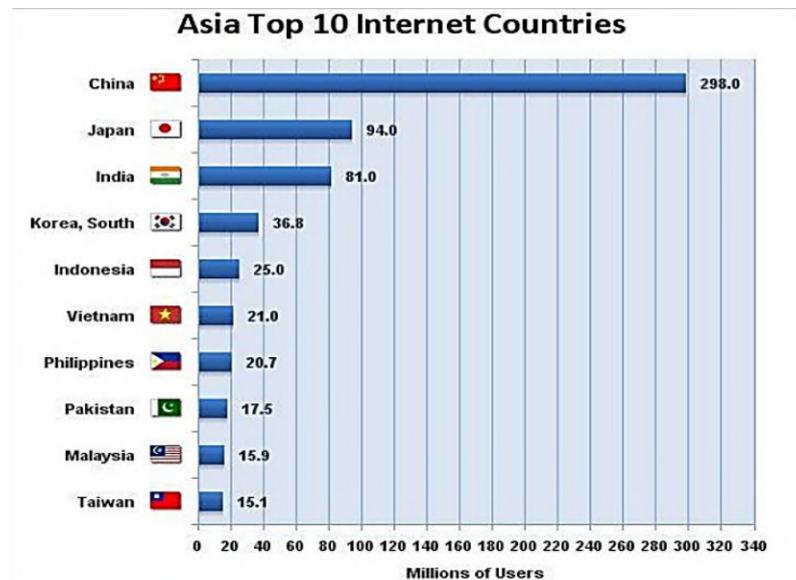
Hal ini penting untuk beberapa alasan, yaitu:

- 1) Dimungkinkan bahwa Sistem Pakar berkaitan dengan nyawa manusia atau properti lainnya sehingga harus dapat menjelaskan mengapa dihasilkan suatu kesimpulan tertentu,
- 2) Untuk mengkonfirmasi bahwa pengetahuan pakar telah dikumpulkan dengan benar dan digunakan oleh sistem dengan benar pula. Hal ini penting dalam proses *debugging* pengetahuan yang mungkin salah karena pengetikan atau pemahaman yang salah dari *knowledge engineer*.

e. Fleksibel (*flexibility*). Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan.

## **B. Website**

Pada saat ini, dunia teknologi telekomunikasi dan informasi (TIK) berkembang dengan sangat pesat. Salah satu pemicu dalam perkembangannya adalah perkembangan dalam dunia internet yang banyak digunakan oleh penduduk diseluruh dunia. Prasetiadi (2011) mengatakan bahwa pengguna internet di Indonesia mencapai angka 25 juta di tahun 2009 dan menempati peringkat ke-5 di Asia.



**Gambar 2. 7 10 Negara Tertinggi Pengguna Internet di Asia**

Dengan banyaknya pengguna internet di dunia, internet akan semakin berpengaruh dalam kehidupan sehari - hari. Dengan hanya menggunakan *search engine* dalam sebuah *browser*, dapat memperoleh berbagai macam informasi. Selain itu, kegiatan ekonomi akan semakin mudah dengan adanya teknologi internet.

Salah satu yang menarik dari internet adalah *web*. Melalui *web*, seseorang dapat mengakses informasi tidak hanya berupa teks, melainkan seperti gambar, suara, video, dan animasi. *Web* merupakan kumpulan dokumen yang sangat banyak yang berada pada komputer *server (web server)*. Server – server ini tersebar di lima benua, termasuk Indonesia dan terhubung menjadi satu melalui jaringan internet.

Menurut Prasetiadi (2011), kebutuhan dan kebiasaan pengguna internet (*web*) selalu berubah dan bertambah. Dahulu hanya mencari informasi tertentu, namun saat ini interaksi sosial di dunia maya sudah menjadi suatu kebutuhan.

Hal ini dapat dilihat dari maraknya penggunaan situs – situs jejaring sosial, seperti *Facebook*, *Twitter*, *Friendster*, dan sebagainya. Generasi *web* yang seperti ini disebut dengan Era *Web 2.0*.

Dimasa yang akan datang, tren penggunaan internet akan berubah lagi sehingga dibutuhkan teknologi *web* yang lebih modern. Teknologi ini adalah teknologi *Web 3.0*, yaitu *web browser* seolah - olah berperan seperti asisten pribadi seperti manusia dan bisa berkonsultasi atau berkomunikasi dengan teknologi tersebut.

## **1. Perkembangan Teknologi *Website***

### **a. Era *Web 1.0***

Menurut Prasetiadi (2011), istilah Era *Web 1.0* merupakan retronim, yaitu istilah tersebut tidak ada sebelum *booming*-nya Era *Web 2.0* pada tahun 2004. Ciri - ciri Era *Web 1.0* adalah sebagai berikut.

- 1) *Web pages* bersifat statik, *website* hanya bersifat “*read-only*” sehingga terjadi komunikasi satu arah. Pembaca hanya dapat menerima informasi, tanpa berpartisipasi dalam memberikan informasi,
- 2) Penggunaan *frameset*, yaitu tampilan beberapa buah *web page* dalam satu *window*,
- 3) Penggunaan *tag HTML* seperti `<blink>` dan `<marquee>`,
- 4) Adanya fitur buku tamu *online*,
- 5) Penggunaan *button* dengan format *GIF*, biasanya 88 x 31 untuk promosi produk.

**b. Era *Web 2.0***

Prasetiadi (2011) menjelaskan bahwa istilah Era *Web 2.0* diperkenalkan oleh *Darcy DiNucci* dalam artikelnya yang berjudul "*Fragmented Future*". Meskipun demikian, istilah Era *Web 2.0* mulai muncul ke permukaan ketika *Tim O'Reilly* dari *O'Reilly Media* mengadakan konferensi *Web 2.0* pertama kalinya di tahun 2004. Berikut ini adalah beberapa karakteristik *Web 2.0*.

- 1) Pengguna *web* dapat mengubah isi *web page*, seperti pada *Wikipedia*,
- 2) Penggunaan *web page* untuk berhubungan dengan *user* lain, seperti pada situs *Facebook* atau *MySpace* yang *booming* karena pengguna mudah untuk menjalin silaturahmi dengan rekannya, bahkan dengan teman - teman lama yang sudah lama tidak dijumpai,
- 3) *Share content* yang cepat dan efisien, seperti *Youtube*, pengguna dapat membuat video dan mengunggahnya ke dalam *channel* miliknya agar orang lain bisa dapat melihat video tersebut,
- 4) Cara baru untuk mendapatkan informasi, yaitu *Really Simple Syndication* (RSS), *user* dapat menerima notifikasi apabila suatu *web page* ter-update,
- 5) Koneksi internet tidak terbatas melalui komputer, tetapi juga dapat melalui perangkat lain, seperti telepon seluler.

**c. Era *Web 3.0***

Konsep Era *Web 3.0* pertama kali diperkenalkan pada tahun 2001,

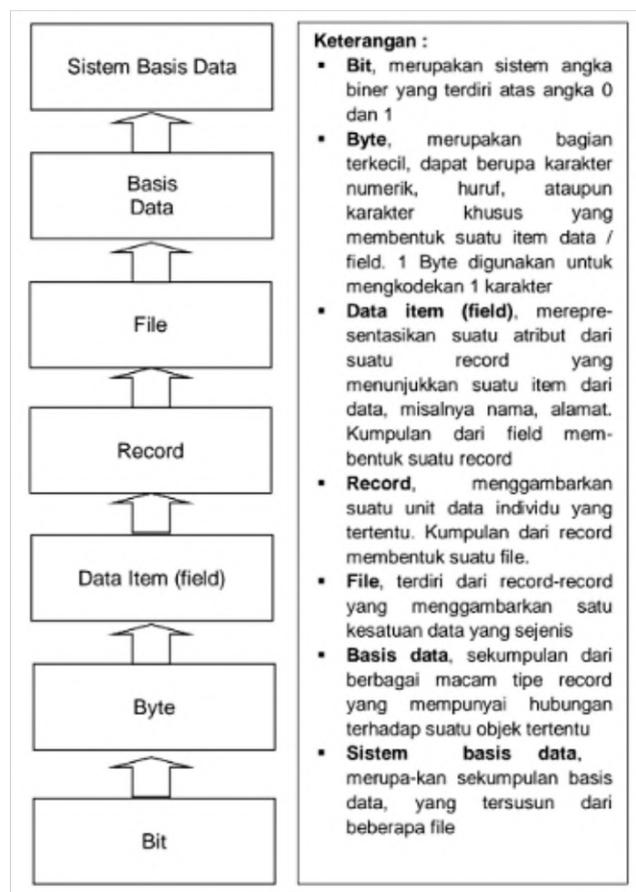
yaitu saat Tim *Berners-Lee*, penemu *World Wide Web*, menulis sebuah artikel ilmiah yang menggambarkan Era *Web 3.0* sebagai sebuah sarana bagi mesin untuk membaca halaman - halaman *web* (Prasetiadi, 2011). Berikut ini adalah gambaran tentang Era *Web 3.0*.

- 1) Teknologi *search engine* dapat memperoleh informasi, namun ini membutuhkan waktu yang lama jika melakukannya secara manual. Dengan teknologi *Web 3.0*, pencarian ini dilakukan oleh internet itu sendiri. *Browser* akan menganalisa dan menampilkan data yang berkaitan ke *user*. Hal ini tentu akan lebih praktis.
- 2) *Search engine* pada teknologi *Web 3.0* membuat *keyword* yang dimasukkan akan menampilkan hasil pencarian yang lebih relevan dan tepat sasaran.
- 3) Setiap *user* mempunyai profil internet yang unik berdasarkan catatan kebiasaan *user* ketika mengakses internet. *Web 3.0* memanfaatkan *history* ini ketika melakukan *searching*. Artinya, ketika dua *user* yang berbeda melakukan *searching* dengan *keyword* yang sama, hasil yang ditampilkan kemungkinan dapat berbeda, sesuai dengan rekaman *browsing* masing - masing pengguna internet.

### C. Basis Data (*Database*)

Sistem informasi berperan sebagai sistem karena mempunyai ruang lingkup yang relatif lebih luas dan lebih kompleks. Sedangkan sistem basis data merupakan subsistem karena menjadi bagian dan berada dalam sistem informasi.

Sistem basis data adalah sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lain dan membuatnya tersedia untuk beberapa aplikasi yang bermacam - macam didalam suatu organisasi. Keberadaan sistem basis data didalam sistem informasi adalah mutlak. Sistem informasi tidak akan terwujud tanpa melibatkan basis data.



Gambar 2. 8 Penyusun Sistem Basis Data

Sumber : Pamungkas, 2017

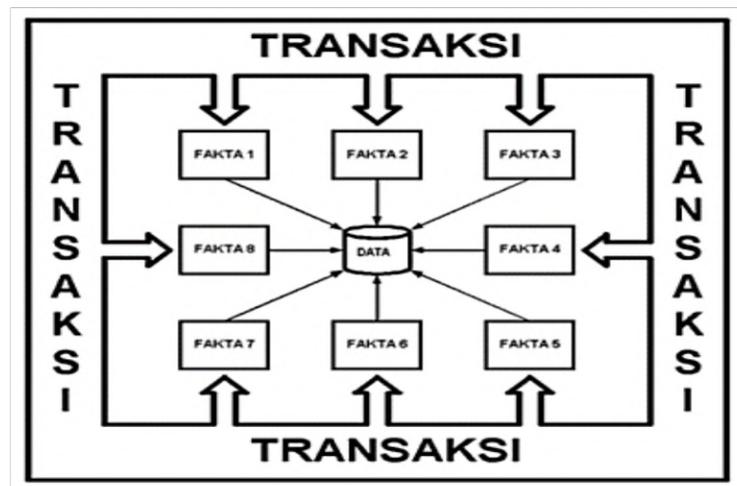
## 1. Pengertian Data

Data adalah fakta - fakta yang diperoleh dan menggambarkan suatu kejadian yang sebenarnya pada waktu tertentu. Jadi data didapat dari suatu kejadian yang benar terjadi, misalnya data penjualan didapat dari data hasil penjualan, data pembelian didapat dari kejadian pembelian, dan sebagainya.

Data dalam suatu perusahaan yang menjual produk identik dengan bukti transaksi, misalnya kuitansi pembayaran dan sebagainya. Sedangkan data dalam suatu perusahaan, data dapat diidentikkan dengan laporan tertulis baik antar departemen maupun dari luar perusahaan, misalnya bukti pajak, bukti bank, dan sebagainya. Menurut Lubis (2016) penggunaan data pada basis data mempunyai peran yang kuat pada masing - masing bagian, yang dikelompokkan dalam tiga jenis data pada sistem basis data, yaitu:

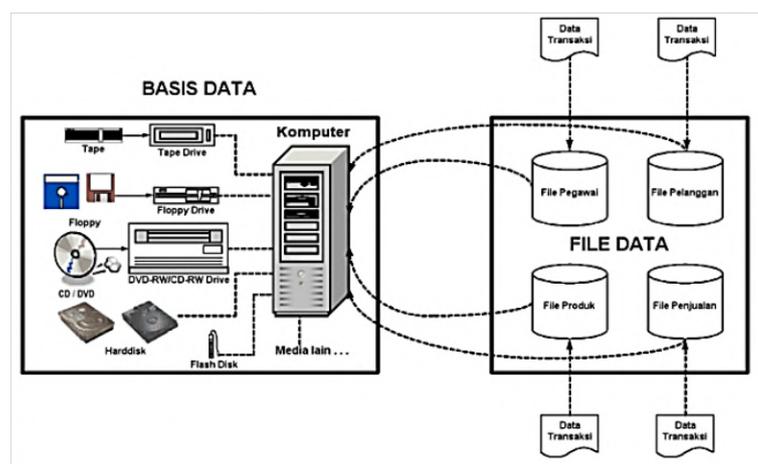
- a. Data operasional dari suatu organisasi, berupa data yang disimpan didalam basis data.
- b. Data masukan (*input data*), data dari luar sistem yang dimasukkan melalui peralatan *input (keyboard)* yang dapat mengubah data operasional.
- c. Data keluaran (*output data*), berupa laporan melalui peralatan *output (screen, printer)* sebagai hasil dari dalam sistem yang mengakses data operasional.

Yang terpenting adalah bahwa data tersebut berasal, sebagai bukti adanya suatu transaksi, dan seterusnya. Lubis (2016) mengemukakan lebih jelas pada gambar berikut.



Gambar 2. 9 Data yang didapat Dari Fakta Pada Transaksi

Setelah semua fakta diterima, fakta - fakta tersebut dikelompokkan kedalam data. Kemudian data tersebut dimasukkan kedalam *file - file* yang disebut dengan *file data*. Dari *file data* tersebut akan dimasukkan dan dihubungkan antara *file data* yang satu dengan *file data* lainnya dalam sebuah wadah penampungan data yang dikenal dengan basis data (*database*).



Gambar 2. 10 Hubungan Data dengan Basis Data

Basis data dalam dunia komputer, terutama oleh *programmer* sudah tidak asing lagi karena seringkali disinggung dan berhubungan langsung. Basis data merupakan kumpulan *file - file* data yang dibentuk dengan suatu

relasi yang membuatnya saling terhubung dan dapat diungkapkan dengan catatan serta bersifat independen (Lubis, 2016).

Basis data menggunakan media penyimpanan yaitu berkaitan dengan setiap alat yang dapat menerima data yang dapat disimpan, dan dapat dipanggil kembali data itu pada waktu berikutnya atau setiap alat yang dapat digunakan untuk menyimpan data. Media penyimpanan yang digunakan seperti *diskette* atau *floppy disk*, *tape*, *compact disk (CD)* atau *DVD*, *harddisk* dan juga *flashdisk*. Menurut Lubis (2016) terdapat empat komponen dasar sistem basis data, yaitu:

**a. Data**

Data yang digunakan dalam sebuah basis data harus memiliki ciri - ciri sebagai berikut:

- 1) Data disimpan secara terintegrasi, yaitu *database* merupakan kumpulan dari berbagai macam *file* dari aplikasi - aplikasi yang berbeda yang disusun dengan cara menghilangkan bagian yang rangkap atau redundansi.
- 2) Data dapat dipakai secara bersama-sama (*shared*), yaitu masing - masing bagian dari *database* dapat diakses oleh pemakai dalam waktu yang bersamaan, untuk aplikasi yang berbeda.

**b. Hardware**

Terdiri dari semua peralatan perangkat keras komputer yang digunakan untuk pengelolaan sistem *database*, seperti:

- 1) Peralatan untuk penyimpanan, disk, drum dan sebagainya,
- 2) Peralatan *input* dan *output*,
- 3) Peralatan komunikasi data dan sebagainya.

**c. *Software***

Berfungsi sebagai *interface* antara *user* dengan data fisik pada *database*, dapat berupa:

- 1) *Database Management System (DBMS)*,
- 2) Program - program aplikasi dan prosedur - prosedur yang lain, seperti *Oracle*, *SQL Server*, *MySQL*, dan sebagainya.

**d. *User***

Terbagi menjadi 3 klasifikasi:

- 1) *Database Administrator (DBA)*, yaitu orang atau tim yang bertugas mengelola sistem *database* secara keseluruhan.
- 2) *Programmer*, yaitu orang atau tim yang membuat program aplikasi yang mengakses *database* dengan menggunakan bahasa pemrograman.
- 3) *End User*, orang yang mengakses *database* melalui terminal dengan menggunakan *query language* atau program aplikasi yang dibuat oleh *programmer*.

Penggunaan basis data pada komputer memang tidak terlepas dari hubungan atau relasi antar data dalam bentuk *file*. Adapun *file* menurut Noor (dalam Lubis, 2016) adalah suatu pengumpulan yang terorganisir dari catatan yang saling berhubungan. Misalnya satu garis (*line*) dari faktur dapat

berbentuk *item*, suatu faktur dapat membentuk catatan, suatu serangkaian catatan yang sedemikian ini dapat membentuk suatu *file*, pengumpulan dari *file* kontrol keuangan dapat berbentuk perpustakaan (*library*), dan keseluruhan *library* yang digunakan oleh suatu organisasi dalam bentuk bank data.

*File* data identik dengan tabel - tabel yang terdiri dari *field* dan *record*. Pada beberapa tipe konvensional yang termasuk *file* dan tabel menurut Jeffrey L. Whitten et. al. (dalam Lubis, 2016) antara lain:

**a. *File Master***

*File Master* merupakan tabel yang berisi *record* yang bersifat tetap. Sekali *record* ditambahkan kedalam *file master*, akan tetap dan dapat digunakan selama dioperasikan pada sebuah sistem. Sedangkan perubahan untuk *record* biasanya dilakukan untuk waktu yang tidak terbatas, akan tetapi tiap *record* akan dapat bertahan terhadap sekumpulan *record* dalam batas waktu yang tidak terbatas. Biasanya *file master* ini digunakan sebagai pedoman dalam penggabungan beberapa *file* yang ada. Contoh *file master* adalah: *file* karyawan, *file* pelanggan, *file* produk, *file* pemasok, dan sebagainya.

**b. *File Transaksi***

*File* transaksi adalah tabel yang berisi gambaran tiap peristiwa perusahaan. Penggambaran yang dimaksud adalah penggambaran dalam skala yang normal atau tidak dibuat-buat dan dalam waktu yang

terbatas. Pembuatan tabel transaksi tidak dapat diwakili atau kolektif, melainkan harus dapat menggambarkan semua transaksi yang terjadi. Yang termasuk *file* transaksi adalah *file* pendaftaran atau registrasi, *file* pembelian, *file* pengiriman, dan sebagainya.

**c. *File Dokumen***

Merupakan tabel yang berisi penyimpanan salinan *record* yang berisi data yang bersejarah untuk memudahkan pencarian kembali dan dapat dilihat dengan menggunakan *overhead*.

**d. *File Arsip***

Merupakan *file* atau tabel yang berisi *file master* atau transaksi yang telah dihapus dari penyimpanan secara langsung. Penghapusan bertujuan untuk memindahkan data dari penyimpanan data yang *online* ke penyimpanan data yang *offline*. Pembuatan *file* arsip ini digunakan untuk kebutuhan dalam kebijakan dan kebutuhan saat audit atau analisis.

**e. *File Table Look-Up***

Dalam pembuatan sistem informasi, relasi dalam tabel banyak digunakan. Oleh karena itu yang dimaksud dengan *file table look-up* adalah tabel yang berisi hubungan statis data yang digunakan bersama-sama oleh aplikasi untuk perawatan secara tetap dan dapat memperbaiki kinerja dari sistem, contohnya tabel pajak penjualan, tabel pendapatan pajak dan sebagainya.

**f. *File Audit***

*Update file audit* untuk *file - file* yang lain, khususnya *file master* dan transaksi. Semuanya digunakan untuk bersama dengan *file arsip* untuk menelusuri data yang hilang (*lost*). Penelusuran jejak dengan audit dapat digunakan membangun teknologi basis data yang lebih baik.

## 2. Keuntungan dan Kerugian Penggunaan Basis Data

Menurut Lubis (2016) penggunaan basis data pada mempunyai keuntungan, antara lain:

- a. Terkontrolnya kerangkapan data dan inkonsistensi,
- b. Terpeliharanya keselarasan data,
- c. Data dapat dipakai secara bersama-sama,
- d. Memudahkan penerapan standarisasi,
- e. Memudahkan penerapan batasan - batasan pengamanan,
- f. Terpeliharanya integritas data,
- g. Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi,
- h. Program atau data independen.

Adapun kerugian penggunaan basis data menurut Lubis (2016), yaitu:

- a. Mahal dalam implementasinya,
- b. Rumit atau kompleks,
- c. Penanganan proses *recovery* dan *backup* sulit,
- d. Kerusakan pada sistem basis data dapat mempengaruhi departemen yang terkait,
- e. Dan sebagainya.

### 3. Perbedaan TFM dengan DBMS

Pada awal penggunaan data, perusahaan masih menggunakan data yang terpisah - pisah penggunaannya (*traditional file management* atau TFM), tergantung pemrogram yang membuatnya. Dengan *Database Management System* (DBMS), perbedaan dan kelemahan tersebut dapat diatasi. Adapun perbedaan antara *traditional file management* (TFM) dengan *database management system* (DBMS) menurut Lubis (2016), yaitu:

#### a. *Traditional File Management* (TFM)

- 1) Bersifat program *oriented*,
- 2) Bersifat kaku,
- 3) Terjadi kerangkapan data dan tidak terjaminnya keselarasan data (data inkonsistensi).

#### b. *Database Management System* (DBMS)

- 1) Bersifat data *oriented*,
- 2) Bersifat luwes atau fleksibel,
- 3) Kerangkapan data serta keselarasan data dapat terkontrol.

### 4. Istilah - Istilah yang dipergunakan pada Sistem Basis Data

Menurut Lubis (2016) untuk mempermudah pembuatan dan penggunaan basis data, digunakan beberapa istilah, antara lain:

- a. *Enterprise*, suatu bentuk organisasi.

Contoh:

Data sekolah → data mhs

Rumah sakit → pasien

- b. *Entity* (entitas), suatu objek yang dapat dibedakan dengan objek lainnya.

Contoh:

Bidang administrasi → entitas mahasiswa, buku, pembayaran.

Bidang kesehatan → entitas pasien, dokter, obat.

- c. *Attribute* atau *Field*, setiap entitas mempunyai atribut atau suatu sebutan untuk mewakili suatu entitas.

Contoh:

*Entity* siswa → *field* NIM, nama\_siswa, alamat

*Entity* nasabah → *field* kode\_nasabah, nama\_nsh

- d. *Data Value* (nilai atau isi data), data aktual atau informasi yang disimpan pada tiap data elemen atau *attribute*. Isi dari atribut disebut nilai data.

Contoh:

Atribut nama karyawan → Sutrisno, Budiman

- e. *Record* atau *Tuple*, kumpulan elemen - elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap.

Contoh:

Satu *record* mewakili satu informasi → nim, nm\_mhs, alamat

- f. *File*, kumpulan *record* sejenis yang mempunyai panjang elemen sama, *attribute* sama, namun berbeda-beda *data value*-nya.

- g. Kunci elemen data, tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas.

- h. *Database Management System (DBMS)*, kumpulan *file* yang saling berkaitan bersama dengan program pengelolaannya.

#### 5. *Database Management System (DBMS)*

Menurut Lubis (2016), DBMS adalah perangkat lunak yang menangani semua pengaksesan *database*. Dengan DBMS, diharapkan basis data dapat dikelola dengan baik dan mudah dalam penggunaannya. DBMS mempunyai fungsi antara lain:

- a. *Data Definition*, DBMS harus dapat mengolah dan pendefinisian data.
- b. *Data Manipulation*, DBMS harus dapat menangani permintaan dari pemakai untuk mengakses data.
- c. *Data Security and Integrity*, DBMS harus dapat memeriksa keamanan dan integritas data yang didefinisikan oleh DBA. Untuk melaksanakannya, dapat dilakukan sebagai berikut:
  - 1) *Data Recovery and Concurrency*, DBMS harus dapat menangani kegagalan - kegagalan pengaksesan *database* yang dapat disebabkan oleh kesalahan sistem, kerusakan *disk*, dan sebagainya.
  - 2) *Data Dictionary*, DBMS harus menyediakan kamus data yang berfungsi untuk identifikasi isi data pada sebuah *database*. Penambahan kamus data ini bertujuan untuk memudahkan pembacaan oleh sistem analis lain atau *user* terhadap data apa saja yang terdapat pada suatu sistem, baik dengan menggunakan DFD maupun ER-D. Untuk pembuatan kamus data, perlu diperhatikan notasi apa saja yang digunakan.

Menurut Lubis (2016) dibutuhkan komponen - komponen dalam penggunaan DBMS, antara lain:

- a. *Query Processor*, komponen yang mengubah bentuk *query* dalam bentuk instruksi kedalam *database manager*.
- b. *Database Manager*, menerima *query*, menguji eksternal dan konseptual untuk menentukan apakah *record - record* tersebut dibutuhkan untuk memenuhi permintaan kemudian hari dari *database manager* dengan memanggil *file manager* untuk menyelesaikan permintaan.
- c. *File Manager*, memanipulasi penyimpanan *file* dan mengatur alokasi ruang penyimpanan *disk*.
- d. *DML Processor*, modul yang mengubah perintah DML yang ditempelkan kedalam program aplikasi dalam bentuk fungsi - fungsi.
- e. *DDL Compiler*, mengubah *statement* DDL menjadi kumpulan tabel atau *file* yang berisi *data dictionary* atau meta data.
- f. *Dictionary Manager*, mengatur akses dan memelihara *data dictionary*.

## 6. *Data Independence*

Merupakan salah satu kelebihan dari sistem DBMS yaitu DBA dapat bekerja mengubah struktur *storage* atau penyimpanan yang digunakan dan strategi akses dalam pengembangan sistem *database* tanpa mengganggu program - program aplikasi yang sudah tersedia. Menurut Lubis (2016) *data independence* terbagi menjadi dua tingkatan, yaitu:

- a. *Physical Data Independence*, yaitu perubahan internal *schema* dapat dilakukan tanpa mengganggu *conceptual schema*.

- b. *Logical Data Independence*, yaitu *conceptual schema* dapat dirubah tanpa mempengaruhi eksternal *schema*.

Pembuatan *data independence* diperlukan untuk diterapkan pada pengelolaan sistem basis data, disebabkan antara lain:

- a. *Database Administrator* dapat mengubah isi, lokasi, dan organisasi *database* tanpa mengganggu program aplikasi yang ada.
- b. *Vendor Hardware* dan *Software* pengelolaan data bisa memperkenalkan produk - produk baru tanpa mengganggu program - program aplikasi yang tersedia.
- c. Memudahkan pengembangan program aplikasi.
- d. Memberikan fasilitas pengontrolan terpusat oleh DBA demi keamanan dan integritas data dengan memperhatikan perubahan berdasarkan kebutuhan *user*.

## **D. Pemograman *Web***

### **1. Aplikasi Berbasis *Web***

Aplikasi atau perangkat lunak (*software*) merupakan bagian penting dari suatu sistem komputer, disamping keberadaan pengguna (*brainware*), perangkat keras (*hardware*), dan jaringan (*networking*). Aplikasi dapat dibagi menjadi aplikasi berbasis *desktop*, aplikasi berbasis *web*, dan aplikasi berbasis *mobile*. Aplikasi berbasis *desktop* merupakan aplikasi yang memerlukan proses instalasi di setiap komputer yang akan menggunakannya. Contoh aplikasi berbasis *desktop* seperti *Microsoft Office*, *Mozilla Firefox*, *Adobe Photoshop* dan sebagainya.

Sementara itu aplikasi berbasis *web* tidak memerlukan instalasi disetiap komputer yang akan menggunakannya karena aplikasi berada didalam server. Untuk membuka aplikasi cukup menggunakan *browser* yang terhubung melalui jaringan ke server. Situs *web* merupakan salah satu contoh jenis aplikasi berbasis *web*.

Aplikasi berbasis *mobile* merupakan aplikasi yang hanya dapat dijalankan pada perangkat bergerak (*mobile*) seperti *handphone*, *smartphone*, dan PDA. Contoh dari jenis aplikasi ini seperti *OperaMini*, *Blackberry Messenger* (BBM), *Whatsapp Messenger*, *Polaris Office* dan sebagainya.

Perkembangan aplikasi berbasis *web* sangat pesat karena memang memiliki beberapa kelebihan dibanding dengan aplikasi berbasis *desktop*. Solichin (2016) berpendapat bahwa ada beberapa kelebihan yang dimiliki oleh jenis aplikasi berbasis *web* seperti:

- a. Pada sisi *client* (pengguna), tidak memerlukan proses instalasi. Jika terjadi perubahan aplikasi, *client* juga tidak perlu melakukan proses *update* karena cukup dilakukan disisi server.
- b. Dapat diakses darimana saja melalui jaringan. Jika server berada dalam jaringan intranet (LAN), aplikasi dapat diakses dari seluruh komputer didalam jaringan intranet tersebut. Dan jika server aplikasi berada di jaringan internet (memiliki *IP Public* atau diletakkan di *web hosting*), maka aplikasi dapat diakses dari internet.

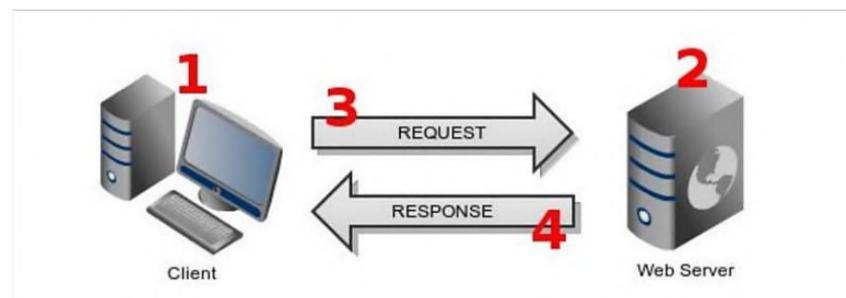
- c. Data disimpan disisi server, sehingga akses terhadap data dari sisi *client* (pengguna) dapat diatur sesuai kebutuhan.
- d. *Cross-Platform*, artinya aplikasi dapat diakses melalui komputer dengan berbagai sistem operasi (*Windows, Linux, atau Mac*) dengan syarat memiliki *browser*.
- e. Dari sisi *client*, tidak memerlukan spesifikasi komputer yang besar karena hampir seluruh proses aplikasi dilakukan disisi server.
- f. *Client* (pengguna) lebih aman dari virus atau gangguan keamanan lainnya karena aplikasi berjalan diatas *browser*.

Aplikasi berbasis *web* juga memiliki kekurangan terutama dari sisi performa aplikasi yang sangat bergantung pada kondisi jaringan yang digunakan. Pada jaringan yang lambat, performa aplikasi juga menurun sehingga akses menjadi lama. Selain itu, menurut Solichin (2016) aplikasi berbasis *web* yang bisa diakses dimana saja, membuatnya rentan terkena serangan dari pihak - pihak luar yang tidak bertanggung jawab. Oleh karena itu perlindungan dan pengamanan terhadap aplikasi mutlak dilakukan oleh pengembang dengan sebaik-baiknya.

## **2. Arsitektur Aplikasi Berbasis *Web***

Menurut Solichin (2016) Aplikasi berbasis *web*, baik yang sederhana maupun yang kompleks, berjalan atas dasar mekanisme kerja *client-server*. Sisi *client* diwakili oleh suatu perangkat elektronik (komputer, *handphone, smartphone, PDA, dan sebagainya*) yang tertanam perangkat lunak *web browser*. Sedangkan sisi server merupakan suatu perangkat komputer yang

terinstall *web server* didalamnya. *Client* dan *server* berkomunikasi melalui suatu jaringan dalam lingkup yang kecil (*localhost*, LAN) maupun besar (*internet*).



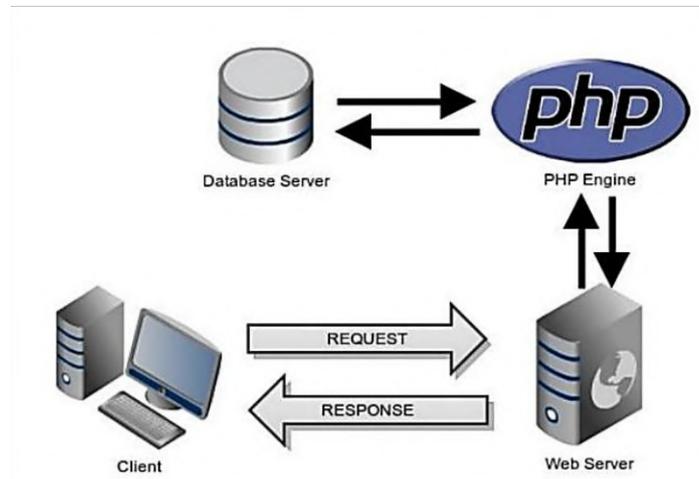
**Gambar 2. 11 Cara Kerja Model Client-Server**

Penjelasan gambar cara kerja model *Client-Server* menurut Solichin (2016):

- a. *Client* dapat berupa komputer *desktop*, laptop, *handphone* atau *smartphone* dengan minimal memiliki *browser* dan terhubung ke *web server*.
- b. Komputer yang berfungsi sebagai *server*, didalamnya terdapat perangkat lunak *web server* dan terhubung ke dalam jaringan dan memiliki alamat atau *IP*.
- c. *Client* meminta suatu halaman ke *server* untuk ditampilkan dikomputer *client*. Melalui jaringan dan protokol *HTTP*, permintaan akan diproses. Proses ini disebut *request*.
- d. *Server* menerima *request* dari *client* dan *server* akan mencari halaman yang diminta oleh *client* agar bisa dikirimkan ke komputer *client* untuk ditampilkan. *Request client* tidak selalu bisa dipenuhi karena tidak tersedianya halaman yang diminta oleh *client* di dalam *server*. Maka

*server* akan mengirimkan pesan bahwa permintaan tidak ditemukan.

Proses ini disebut *response*.



**Gambar 2. 12** Arsitektur Aplikasi Berbasis Web Dinamis

Gambar diatas merupakan arsitektur aplikasi berbasis *web* dinamis. Arsitektur ini digunakan apabila *server* memiliki *client* dalam jumlah yang banyak. Dengan bantuan *script* PHP atau yang biasa disebut PHP *Engine*, *web server* tidak langsung mengirim *response* kepada *client* tetapi *web server* akan meminta bantuan dari PHP *Engine* untuk menangani *request* dari *client* dan PHP *Engine* akan mengambil data dari *Database Server* dan menerjemahkannya. Selanjutnya akan dikembalikan lagi kepada *Web Server* dan mengirimkan *response* kepada *Client* untuk ditampilkan.

### 3. HTML, CSS, dan Javascript

HTML merupakan singkatan dari *Hypertext Markup Language*. HTML dikembangkan pertama kali oleh *Tim Berners-Lee* bersamaan dengan protokol HTTP (*Hypertext Transfer Protocol*) pada tahun 1989. Tujuan utama pengembangan HTML adalah untuk menghubungkan satu halaman

*web* dengan halaman *web* lainnya. Tentunya pada awal perkembangannya halaman *web* hanya berupa teks, tidak seperti saat ini (Solichin, 2016).

HTML berupa sekumpulan simbol - simbol atau tag - tag yang dituliskan dalam sebuah file yang dimaksudkan untuk menampilkan halaman pada web browser. Tag - tag tadi memberitahu browser bagaimana menampilkan halaman web dengan lengkap kepada pengguna (Haryono et al., 2016).

Pada dasarnya, setiap halaman *web* dibuat dalam bentuk HTML. HTML merupakan bahasa alami atau bahasa *default* dari sebuah *browser*. HTML menggunakan bahasa sederhana sehingga memudahkan seorang *programmer* untuk mengimplementasikannya. Saat ini, HTML dapat menampilkan objek-objek seperti teks, gambar, tautan, tabel, audio, dan video.

Menurut Solichin (2016) HTML berkembang dari waktu ke waktu. Hal ini membuat terbentuknya standar perintah HTML yang diprakarsai oleh *Tim Berners-Lee* dengan nama W3C (*World Wide Web Consortium*) dan membuat teknologi seperti XML, XHTML, dan SOAP. Beberapa standar yang dikeluarkan oleh W3C antara lain HTML 2.0 pada tahun 1995, HTML 3.2 pada tahun 1997, HTML 4.0 pada Desember 1997, dan direvisi menjadi HTML 4.01 pada bulan Desember 1999. Pada bulan Mei 2000, W3C mengumumkan standar baru HTML dengan nama XHTML yang sebenarnya pengembangan dari HTML 4.01. Selanjutnya standar baru yang

dirilis W3C yaitu HTML 5. Banyak fitur baru yang dikenal pada standar HTML 5 seperti penanganan audio, video, dan penyimpanan lokal.

HTML merupakan bahasa dasar *web* yang berfungsi untuk menampilkan berbagai komponen *web*. Sementara itu, untuk mempercantik tampilan *web*, dikembangkanlah CSS atau *Cascading Style Sheet*. CSS pertama kali diusulkan oleh *Hakon Wium Lie* pada tahun 1994 dan selanjutnya distandarisasi oleh W3C. CSS memberikan cara yang mudah dan efisien bagi *programmer* untuk menentukan tata letak halaman *web* dan mempercantik halaman dengan elemen desain seperti warna, sudut bulat, gradien, dan animasi.

Selain HTML dan CSS, sebuah aplikasi berbasis *web* tidak dapat lepas dari teknologi *Javascript*. Solichin (2016) berpendapat bahwa *Javascript* dikembangkan oleh *Netscape* dengan nama awal *LiveScript*. Fungsi utama dari *Javascript* adalah untuk menambah fungsionalitas dan kenyamanan halaman *web*. *Javascript* lebih fokus pada proses pengolahan data di sisi *client* serta menyajikan komponen *web* yang lebih interaktif. *Javascript* semakin populer sejak kemunculan konsep AJAX (*Asynchronous Javascript and XML*) yang memungkinkan interaksi antara *client* dan *server* lebih elegan dan fleksibel.

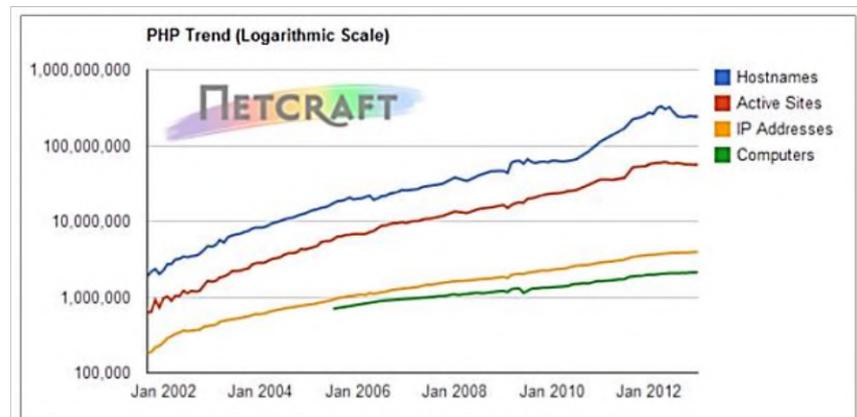
#### 4. PHP

Solichin (2016) berpendapat bahwa PHP pertama kali dikembangkan oleh *Rasmus Lerdorf*, seorang pengembang *software* dan anggota tim *Apache*, dan dirilis pada akhir tahun 1994. PHP dikembangkan dengan

tujuan awal hanya untuk mencatat pengunjung pada *website* pribadi *Rasmus Lerdorf*. Pada rilis keduanya, ditambahkan *Form Interpreter*, sebuah *tools* untuk melakukan penerjemahan perintah SQL. Rilis kedua disebut dengan PHP/FI. Sejak itu, PHP mulai diterima sebagai sebuah bahasa pemrograman baru yang sangat diminati. Terbukti pada pertengahan tahun 1997, tercatat sekitar 50.000 situs diseluruh dunia telah menggunakan PHP.

Mustakim (2018) berpendapat bahwa Sebagian besar dari PHP mirip dengan bahasa *C*, *Java* dan *Perl*, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang *web* menulis halaman *web* dinamik dengan cepat.

Solichin (2016) mengatakan bahwa pada tahun 1998 terjadi peningkatan penggunaan PHP yang sangat besar, bersamaan dengan naiknya popularitas penggunaan teknologi *Open Source*. Pada bulan Oktober 1998, PHP sudah digunakan oleh sekitar 100.000 *domain* di internet. Dan setahun setelahnya, sudah mencapai 1 juta *domain*. Perkembangan selanjutnya pada tahun 2004 tercatat sekitar 15 juta *server web* terpasang PHP didalamnya. Berdasarkan survey yang dilakukan oleh *Netcraft* pada bulan Januari 2013 situs yang menggunakan PHP sudah mencapai 244 juta situs.



Gambar 2. 13 Tren Penggunaan PHP oleh Netcraft

Haryono et al., (2016) berpendapat tentang beberapa kelebihan bahasa pemrograman PHP sebagai berikut:

a. Keamanan

Keamanan sebuah program selain Sistem Operasi menjadi sangat penting. PHP menyediakan tiga jenis autentikasi *user*, yaitu http autentikasi, penggunaan *cookies*, dan penggunaan *session*. Selain itu ada beberapa fungsi disediakan seperti *crc32*, *crypt*, *md5*, *base64-decode*, *base64-encode*, dan sebagainya.

b. Integritas dengan *database*

PHP mendukung integritas, kecepatan dan efisiensi akses ke *database* yang kebanyakan menggunakan *database* berjenis *relational* seperti *MySQL*, *PostgreSQL*, *Oracle*, *SQLite* dan sebagainya.

c. *Cross-Platform*

PHP mendukung berbagai jenis sistem operasi seperti semua varian *Linux*, *Windows*, dan *Mac*. Hal ini membuat bahasa pemrograman PHP lebih fleksibel serta kompatibel dengan banyak sistem operasi.

d. Reliabilitas

PHP merupakan salah satu bahasa pemrograman yang berbasis *web*. Alasan utama adalah dukungan dokumentasi yang lengkap, aman dan banyak komunitas *helpdesk* untuk membantu para pengembang *web* sistem yang menggunakan PHP.

e. Harga

PHP berada dalam lisensi GPL (*GNU Public License*). Hal ini berarti bahwa PHP bebas digunakan dan didistribusikan serta gratis. Saat ini juga banyak *hosting* gratis dan *unlimited support* PHP.

f. Kemudahan bermigrasi

Tujuannya adalah memperbaiki kinerja dan menambah fitur-fitur baru. Kelebihan ini karena banyaknya dukungan terhadap PHP sehingga berdampak PHP terus menerus dikembangkan.

Mustakim (2018) berpendapat bahwa ada beberapa kelebihan dari program PHP yaitu, mendukung komunikasi dengan layanan seperti *protocol IIMAP, SNMP, NNTP, POP3* dan bahkan *HTTP*. PHP dapat diinstall sebagai bagian atau modul dari *apache web server* atau sebagai *CGI Script* yang mandiri. Adapun keuntungan yang diperoleh jika menggunakan PHP sebagai modul dari *Apache* menurut Mustakim (2018), diantaranya:

a. Tingkat keamanan yang cukup tinggi,

- b. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman *web* lainnya yang berorientasi pada *server-side scripting*,
- c. Akses ke sistem *database* yang lebih fleksibel, seperti *MySQL*.

Ada beberapa hal yang dapat dilakukan dengan PHP seperti:

- a. Mengurangi waktu untuk membuat *web* berskala besar,
- b. Mampu menciptakan *web* interaktif dan dinamis,
- c. Menciptakan berbagai *tools* untuk keperluan online,
- d. Mendukung *e-commerce*, *e-learning*, *e-library*, dan CMS (*Content Management System*).

Sebagai catatan hal utama yang harus diperhatikan sebelum melakukan pemrograman PHP adalah mampu dan memahami dasar bahasa pemrograman C/C++ serta *tag* pada HTML (Mustakim, 2018).

Pada PHP, pada dasarnya tipe data variabel tidak didefinisikan oleh *programmer*, akan tetapi secara otomatis ditentukan oleh *interpreter* PHP. Namun untuk beberapa kebutuhan, *programmer* dapat mendefinisikan tipe data variabel. Solichin (2016) berpendapat bahwa PHP mendukung delapan buah tipe data primitif, yaitu:

- a. *Boolean*,
- b. *Integer*,
- c. *Float*,
- d. *String*,
- e. *Array*,
- f. *Object*,

g. *Resource*,

h. *NULL*.

## 5. XAMPP

XAMPP adalah suatu *bundle web server* yang populer digunakan untuk coba - coba di *windows* karena kemudahan instalasinya. XAMPP merupakan perangkat lunak bebas yang mendukung banyak Sistem Operasi, merupakan kompilasi dari beberapa program (Informasi et al., 2018).

XAMPP berperan sebagai server *web* pada komputer lokal. XAMPP juga dapat disebut sebuah server virtual, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan internet.

XAMPP berfungsi sebagai server yang berdiri sendiri (*localhost*) yang terdiri atas *Apache HTTP Server*, *MySQL database* dan penerjemahan bahasa yang ditulis dengan bahasa pemrograman PHP. Nama XAMPP merupakan singkatan dari empat Sistem Operasi yaitu, *Apache*, *MySQL*, *PHP*, dan *Perl*. Program ini tersedia dalam *GNU* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis (Informasi et al., 2018).

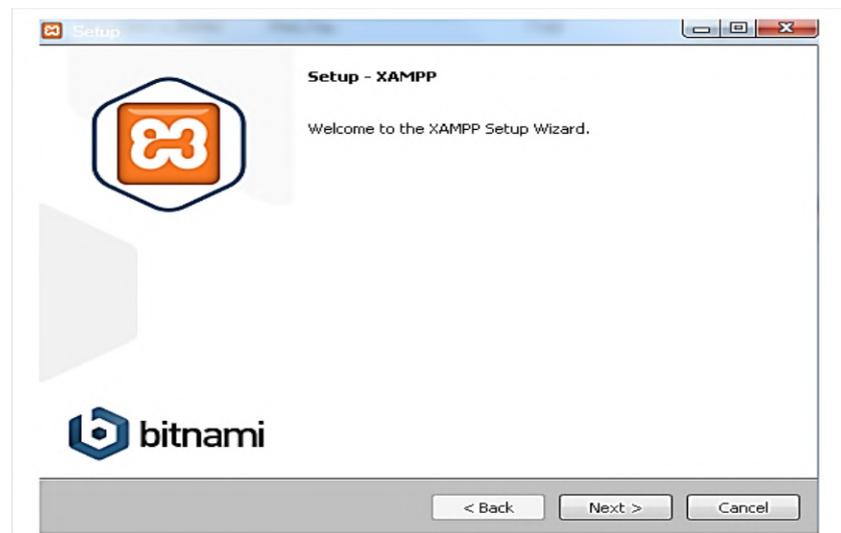
Cara penginstallan XAMPP dapat dilakukan melalui beberapa langkah. Berikut adalah langkah – langkah yang dilakukan dalam penginstallan XAMPP.

a. Unduh XAMPP melalui *website Apache Friends*,



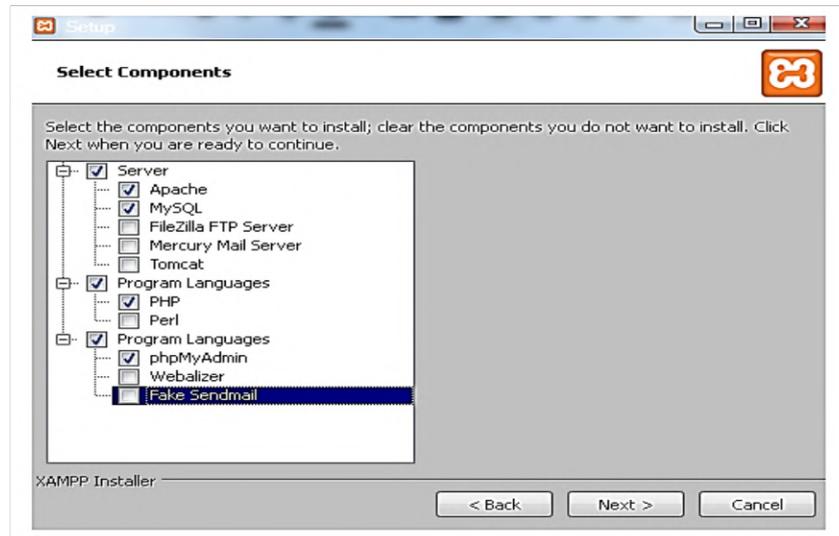
Gambar 2. 14 Halaman website Apache Friends

- b. Melakukan instalasi setelah mengunduh *file* aplikasi XAMPP *Installer*, tekan *Yes* dan klik tombol *Next*,



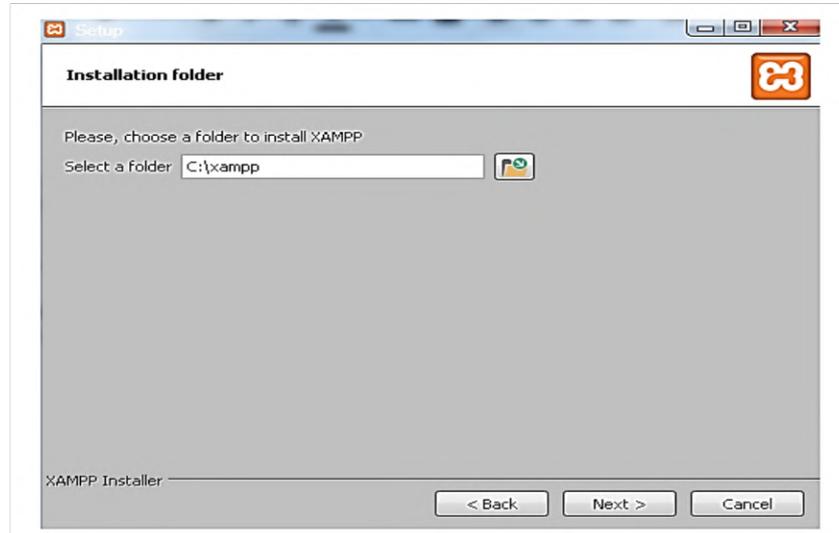
Gambar 2. 15 Halaman setup pertama

- c. Pada tampilan selanjutnya akan muncul pilihan mengenai komponen dari XAMPP yang ingin diinstal ataupun tidak diinstal. Centang pada bagian *MySQL* dan *phpMyAdmin*, untuk pilihan lainnya tetap dalam keadaan *default*.



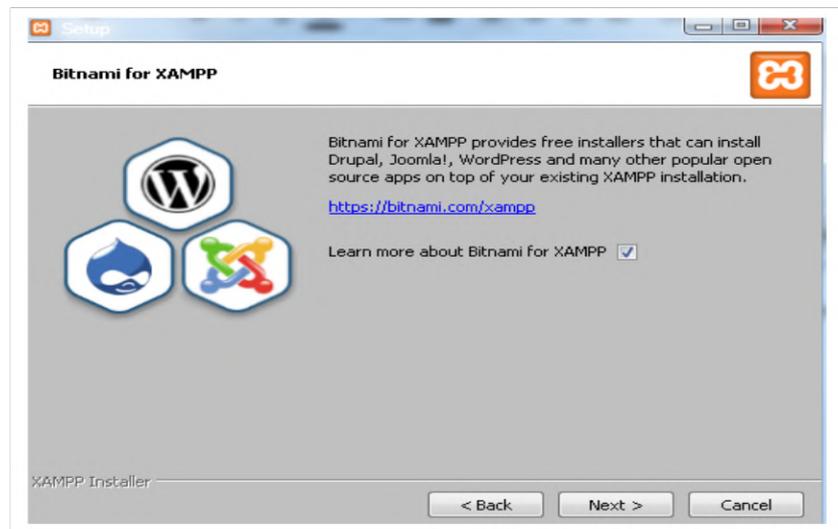
Gambar 2. 16 Halaman select components dari XAMPP

- d. Berikutnya pilih *folder* tujuan tempat dimana XAMPP akan diinstal, atau tetap pada pilihan *default* dari XAMPP,



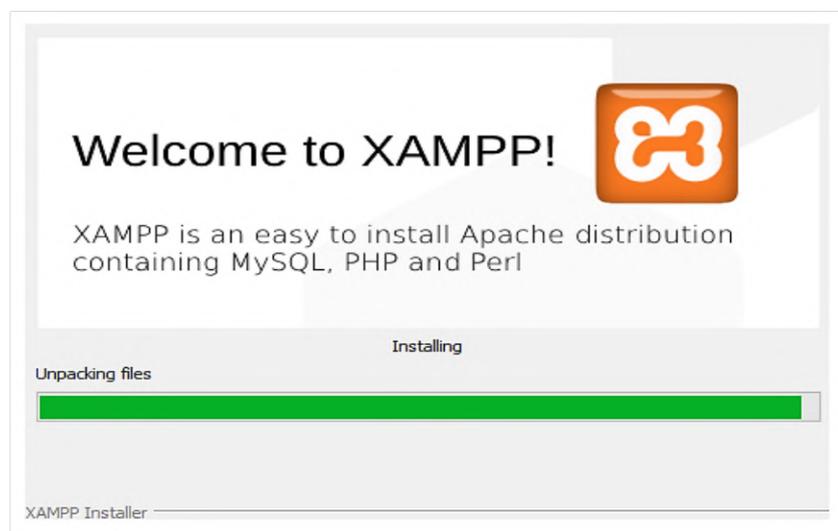
Gambar 2. 17 Halaman installation folder XAMPP

- e. Pada halaman selanjutnya akan muncul pilihan untuk mempelajari tentang *Bitnami for XAMPP* lebih lanjut, pada pilihan ini tetap pada *setting default* dari XAMPP,



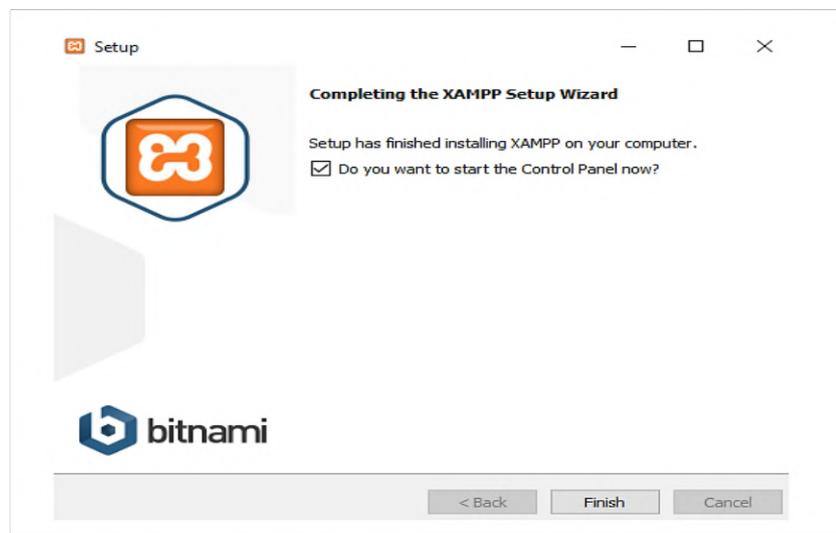
**Gambar 2. 18 Halaman Learn more about Bitnami for XAMPP**

- f. Pada langkah ini proses instalasi XAMPP akan dimulai. Tekan tombol *Next*,



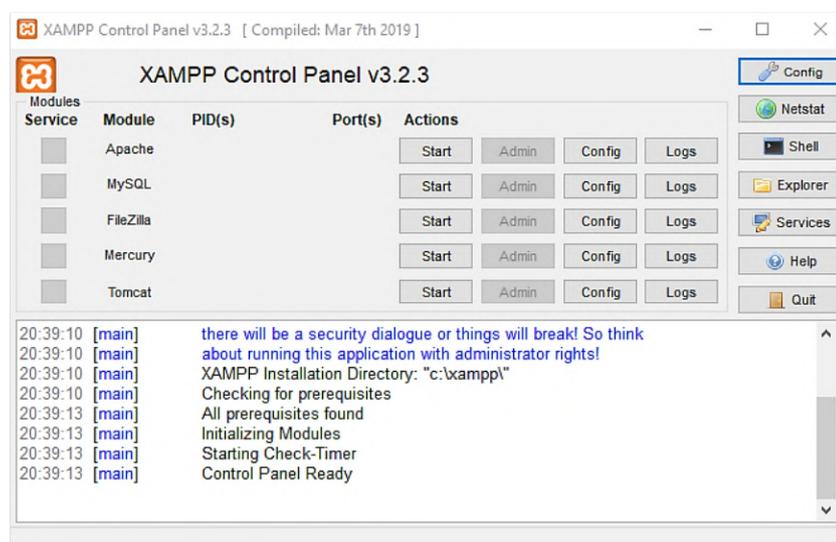
**Gambar 2. 19 Halaman proses instalasi XAMPP**

- g. Setelah berhasil diinstal, akan muncul halaman notifikasi untuk langsung menjalankan *control panel* XAMPP. Klik *Finish*,



Gambar 2. 20 Halaman notifikasi completing XAMPP setup

h. Tampilan XAMPP Control Panel v3.2.3 yang telah berhasil diinstal.



Gambar 2. 21 Tampilan XAMPP Control Panel v3.2.3

## 6. Rekayasa Perangkat Lunak (RPL)

Menurut IEEE (dalam Wicaksono, 2011) menyebutkan bahwa RPL selain sistematis juga merupakan pendekatan yang seharusnya mampu

untuk dikuantifikasikan alias diukur keberadaannya dengan angka-angka atau ukuran tertentu dalam sebuah proses pengembangan perangkat lunak.

RPL meliputi beberapa pokok bahasan penting antara lain:

a. *Domain Engineering*

Mampu memahami permasalahan yang muncul dan akan dijadikan sebagai proyek perangkat lunak,

b. *Requirement Engineering*

Mampu memahami kebutuhan pengguna sekaligus melakukan pemecahan permasalahan,

c. *Software Design*

Mampu memahami serta mengimplementasikan perancangan perangkat lunak termasuk didalamnya aspek HCI (*Human Computer Interaction*),

d. *Development*

Dalam proses pengembangan sebuah perangkat lunak akan melibatkan pembelajaran mengenai algoritma, bahasa pemrograman serta teknik yang berkaitan didalamnya seperti basis data dan sistem informasi.

e. *Operations*

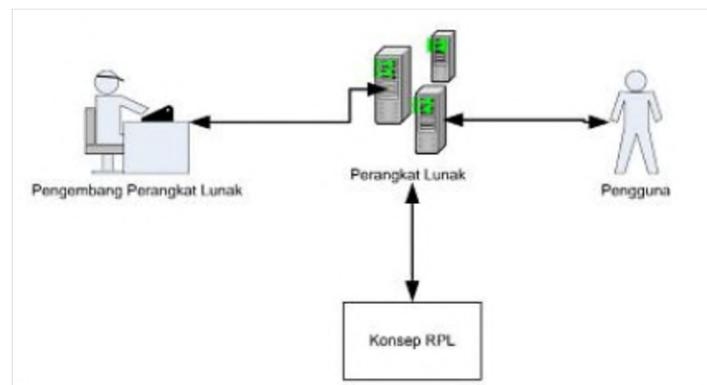
Operasional perangkat lunak dapat dipisahkan menjadi dua bagian yakni pada saat proses *testing* dan pada saat proses implementasi yang didalamnya terdapat langkah - langkah awal seperti pelatihan dan perbaikan pasca produksi.

f. *Maintenance*

Melakukan perawatan dan pemeliharaan terhadap aspek perangkat lunak seperti basis data, instalasi juga didalamnya terdapat proses dokumentasi dari pengembang perangkat lunak.

Menurut Wicaksono (2011) jika dilihat dari proses sebuah RPL sendiri melibatkan beberapa unsur antara lain:

- a. *Software Engineers* atau pengembang perangkat lunak,
- b. *Software* atau perangkat lunak,
- c. *User* atau pengguna.



Gambar 2. 22 Hubungan Unsur RPL dengan RPL

Menurut Wicaksono (2011) ketiga unsur tersebut secara langsung terlibat dalam RPL dan interaksi antara ketiga unsur tersebut yang nantinya menjadi masalah baru, sehingga diperlukan pengetahuan serta pemahaman tentang konsep dasar RPL. RPL merupakan jembatan penghubung utama antar ketiga unsur tersebut dan harus dipahami oleh seorang pengembang perangkat lunak dan juga pengguna. Pengguna dalam konteks yang telah disebutkan diatas bukan hanya direct user atau pengguna yang berhubungan langsung dengan perangkat lunak, tetapi juga didalamnya termasuk indirect

user atau pengguna perangkat lunak yang secara tidak langsung berkaitan dengan perangkat lunak tersebut.

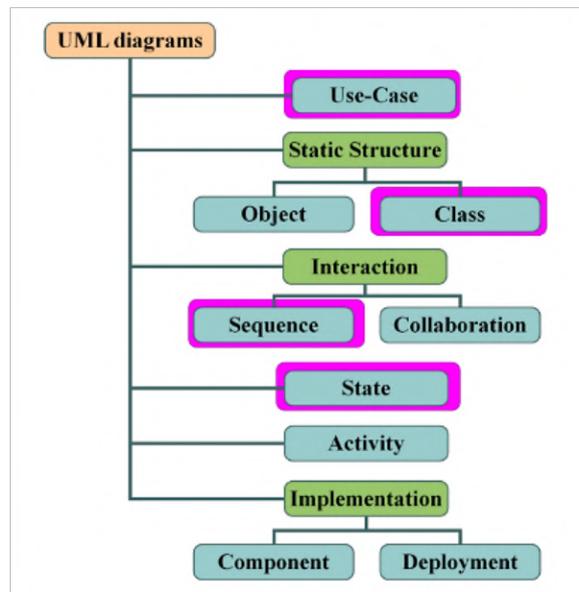
#### **E. UML (*Unified Modeling Language*)**

*Unified Modeling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri visualisasi, merancang dan mendeskripsikan sistem. UML menawarkan sebuah standar untuk merancang model dari suatu sistem. Dengan menggunakan model, pengembangan *software* dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat.

Mamed Rofendy Manalu (dalam Heriyanto, 2018) mengatakan *Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk membuat visualisasi dan spesifikasi dari sebuah sistem pengembangan *software* berbasis *object oriented*.

Sebagian besar perancang sistem informasi saat ini dalam menggambarkan informasi memanfaatkan pemodelan dengan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan validasi desain arsitektur perangkat lunak atau pembuat program (Haviluddin, 2011).

Saat ini konsep UML menggunakan konsep pemodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol - simbol yang cukup spesifik. Berikut gambar diagram UML menurut Haviluddin (2011).



Gambar 2. 23 Diagram UML

Sugrue J. (dalam Havaluddin, 2011) berpendapat bahwa ada beberapa tujuan utama dalam desain UML, seperti:

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna,
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti,
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu,
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan,
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek,
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem,

## 7. Memiliki integrasi praktik terbaik.

Ronald J. N. (dalam Havaluddin, 2011) berpendapat bahwa *Object Oriented Program* (OOP) merupakan paradigma baru dalam rekayasa *software* yang didasarkan pada objek dan kelas. Diakui para ahli bahwa *object-oriented* merupakan metodologi terbaik yang ada saat ini dalam rekayasa *software*. *Object-oriented* memandang setiap bagian *software* dan menggambarkan bagian tersebut kedalam suatu objek.

Havaluddin (2011) mengatakan bahwa ada empat prinsip dasar dari pemrograman berorientasi objek yang menjadi dasar kemunculan UML, yaitu:

### 1. Abstraksi

Memfokuskan perhatian pada karakteristik objek yang paling penting dan paling dominan yang bisa digunakan untuk membedakan objek tersebut dari objek lainnya.

### 2. Enkapsulasi

Menyembunyikan banyak hal yang terdapat dalam objek yang tidak perlu diketahui oleh objek lain. Dalam praktik pemrograman, enkapsulasi diwujudkan dengan membuat suatu kelas *interface* yang akan dipanggil oleh objek lain, sementara didalam objek yang dipanggil terdapat kelas lain yang mengimplementasikan apa yang terdapat dalam kelas *interface*.

### 3. Modularitas

Membagi sistem yang rumit menjadi bagian - bagian yang lebih kecil yang bisa mempermudah *developer* memahami dan mengelola objek tersebut.

#### 4. Hirarki

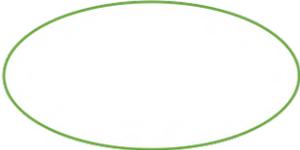
Berhubungan dengan abstraksi dan modularitas, yaitu pembagian berdasarkan urutan dan pengelompokan tertentu. Misalnya untuk menentukan objek mana yang berada pada kelompok yang sama, objek mana yang merupakan komponen dari objek yang memiliki hirarki lebih tinggi. Semakin rendah hirarki objek berarti semakin jauh abstraksi dilakukan terhadap suatu objek.

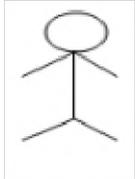
Sugrue J. (dalam Haviluddin, 2011) mengatakan bahwa sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik. Komponen atau notasi UML juga merupakan turunan dari notasi yang telah ada sebelumnya, yakni *Grady Booch*, OOD (*Object-Oriented Design*), *Jim Rumbaugh*, OMT (*Object Modeling Technique*), dan *Ivar Jacobson* OOSE (*Object-Oriented Software Engineering*).

##### 1. *Use Case Diagram*

Menurut Shalahuddin (dalam Heriyanto, 2018) mengungkapkan bahwa *Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi - fungsi tersebut. Berikut adalah simbol - simbol yang ada pada *Use Case Diagram* menurut Heriyanto (2018):

Tabel 2. 2 Simbol pada Use Case Diagram

SIMBOL	DESKRIPSI
Use Case 	Fungsionalitas yang disediakan sistem unit - unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .

SIMBOL	DESKRIPSI
Actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan manusia atau orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi <<extend>> 	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
Generalisasi 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

<p>Include</p> <p>&lt;&lt;include&gt;&gt;</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>
---	---

## 2. Class Diagram

Menurut Rossa (dalam Heriyanto, 2018) *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas - kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel - variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi - fungsi yang dimiliki oleh suatu kelas. Berikut merupakan simbol - simbol yang ada pada *Class Diagram* menurut Sari & David (dalam Heriyanto, 2018):

Tabel 2. 3 Simbol - Simbol yang ada pada Class Diagram

SIMBOL	DESKRIPSI
<p>Kelas</p> <p>Nama Kelas</p> <p>+<i>attribute</i></p> <p>Operasi()</p>	<p>Kelas pada struktur sistem yang memiliki atribut dari setiap entitas dan memiliki operasi sendiri.</p>
<p>Antarmuka / <i>Interface</i></p>  <p>nama_interface</p>	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi</p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>

Asosiasi Berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi ini biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

### 3. Activity Diagram

Rosa (dalam Heriyanto, 2018) berpendapat bahwa *Activity Diagram* menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem. Sari & David (dalam Heriyanto, 2018) mengatakan bahwa ada beberapa simbol - simbol yang digunakan dalam *Activity Diagram*, seperti berikut.

Tabel 2. 4 Simbol - Simbol Activity Diagram

SIMBOL	DESKRIPSI
<i>Start</i> 	Simbol <i>Start</i> untuk menyatakan awal dari suatu proses.
<i>Stop</i> 	Simbol <i>Stop</i> untuk menyatakan akhir dari suatu proses.

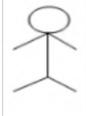
<p><i>Decision</i></p> 	<p>Simbol <i>Decision</i> digunakan untuk menyatakan kondisi dari suatu proses.</p>
<p><i>Action</i></p> 	<p>Simbol <i>Action</i> menyatakan aksi yang dilakukan dalam suatu arsitektur sistem.</p>

#### 4. *Sequence Diagram*

Jurdam (dalam Heriyanto, 2018) mengatakan *Sequence Diagram* adalah *tools* atau alat yang sangat populer dalam pengembangan sistem informasi secara *object-oriented* untuk menampilkan interaksi antar objek. Berdasarkan definisi tersebut, bahwa *Sequence Diagram* adalah *tools* yang digunakan dalam pengembangan sistem. Berikut adalah simbol – simbol yang biasa digunakan dalam *Sequence Diagram*.

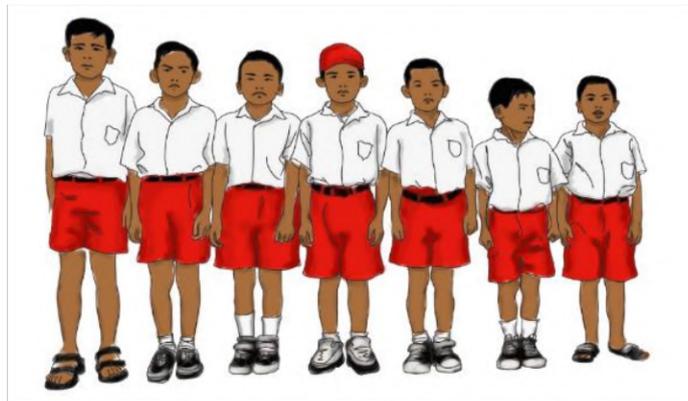
Tabel 2. 5 Simbol - Simbol dalam *Sequence Diagram*

NAMA KOMPONEN	KETERANGAN	SIMBOL
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> .	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . Mengindikasikan sebuah objek yang akan melakukan sebuah aksi.	
<i>Message</i>	<i>Message</i> digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara objek-objek.	

<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama <i>object</i> didalamnya yang diawali dengan sebuah titik koma.	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi.	

### F. *Stunting* (Pendek)

*Stunting* biasa diidentifikasi dengan membuat perbandingan antara ukuran tinggi seorang anak dengan standar tinggi anak pada populasi yang normal sesuai dengan usia dan jenis kelamin yang sama. Anak dikatakan *stunting* jika tingginya berada dibawah -2 SD dari standar WHO (*World Health Organization*) (Trihono et al., 2015).



Gambar 2. 24 Anak kelas 4 SD dengan tinggi yang berbeda

*Stunting* merupakan tragedi yang tersembunyi. *Stunting* terjadi karena dampak kekurangan gizi kronis selama 1.000 hari pertama kehidupan anak. Kerusakan yang terjadi mengakibatkan perkembangan anak yang *irreversible*

(tidak bisa diubah), anak tersebut tidak akan pernah mempelajari atau mendapatkan sebanyak yang dia bisa.

Salah satu indikator kesehatan yang dinilai keberhasilan pencapaiannya dalam MDGs adalah status gizi anak balita. Masa anak balita merupakan kelompok yang rentan mengalami kurang gizi salah satunya adalah *stunting*. *Stunting* merupakan gangguan pertumbuhan *linier* yang disebabkan adanya malnutrisi asupan zat gizi kronis atau penyakit infeksi kronis maupun berulang yang ditunjukkan dengan nilai *Z-Score* tinggi badan menurut umur (TB/U) kurang dari -2 SD (Aridiyah et al., 2015).

Nasikhah (2012) mengatakan bahwa berdasarkan data Riset Kesehatan Dasar (Riskesdas) 2010 diketahui bahwa prevalensi balita *stunting* di Indonesia mencapai angka 35,6% dengan kejadian yang tinggi pada balita usia 24 – 36 bulan (41,4%). Prevalensi *stunting* tersebut lebih tinggi dibandingkan angka prevalensi gizi kurang dan buruk (17,9%), balita kurus (13,3%) serta balita gemuk (14%). Kondisi *stunting* pada masa balita dapat menyebabkan gangguan perkembangan fungsi kognitif dan psikomotor serta penurunan produktivitas ketika dewasa. Beberapa faktor yang diduga berhubungan dengan kejadian *stunting* pada balita antara lain berat badan lahir balita, riwayat infeksi balita, riwayat penyakit kehamilan, tinggi badan orang tua dan faktor sosial ekonomi.

### **1. Faktor - Faktor yang Mempengaruhi *Stunting***

Berdasarkan hasil wawancara yang dilakukan peneliti dengan narasumber, dapat diperoleh beberapa faktor yang mempengaruhi seorang anak terkena *stunting*, seperti:

- a. Kurangnya pengetahuan gizi sebelum dan saat hamil sehingga tidak terpenuhinya gizi ibu saat hamil,
- b. Anak tidak mendapatkan ASI Eksklusif dari 0-6 bulan,
- c. Anak tidak mendapat MPASI (makanan pendamping ASI) yang bergizi dari 0-24 bulan,
- d. Kurangnya akses air bersih dan sanitasi,
- e. Bayi lahir BBLR (berat badan lahir rendah),
- f. Terbatasnya layanan kesehatan,
- g. Tingkat ekonomi keluarga yang rendah.

## **2. Ciri - Ciri dan Gejala *Stunting***

Berdasarkan hasil wawancara yang dilakukan peneliti dengan narasumber, dapat diperoleh beberapa ciri - ciri dan gejala yang terjadi pada seorang anak ketika terkena *stunting*, seperti:

- a. Wajah tampak lebih muda dari anak seusianya,
- b. Pertumbuhan tubuh dan gigi yang terlambat,
- c. Memiliki kemampuan fokus dan memori belajar yang kurang baik,
- d. Pubertas yang lambat,
- e. Saat menginjak usia 8-10 tahun, anak cenderung lebih pendiam dan tidak banyak melakukan kontak mata dengan orang sekitarnya,
- f. Berat badan lebih ringan untuk anak seusianya.

Setiawan et al., (2018) berpendapat bahwa masalah kurang gizi dan *stunting* merupakan dua masalah yang saling berhubungan. *Stunting* pada anak merupakan dampak dari defisiensi nutrisi selama seribu hari pertama

kehidupan. Hal ini menimbulkan gangguan perkembangan fisik anak yang *irreversible*, sehingga menyebabkan penurunan kemampuan kognitif dan motorik serta penurunan performa kerja. Anak *stunting* memiliki rata - rata skor *Intelligence Quotient* (IQ) sebelas poin lebih rendah dibandingkan dengan rata - rata skor IQ pada anak normal. Gangguan tumbuh kembang pada anak akibat kekurangan gizi bila tidak mendapatkan intervensi sejak dini akan berlanjut hingga dewasa.

Kusumawati et al., (2015) mengatakan bahwa Indonesia termasuk diantara 36 negara didunia yang memberi 90 % kontribusi masalah gizi dunia. Berdasarkan hasil Riset Kesehatan Dasar (Riskesdas) tahun 2010, provinsi Jawa Tengah memiliki prevalensi *stunting* sebesar 33,9% yang terdiri dari 16,9% sangat pendek dan 17% pendek. Berdasarkan pemantauan status gizi (PSG) Kabupaten Banyumas tahun 2011, prevalensi bawah tiga tahun (*batita*) *stunting* sebesar 28,37% dan Puskesmas Kedungbanteng prevalensi *stunting* sebesar 41,6% yang terdiri dari 18,8% sangat pendek dan 11,8% pendek.

Trihono et al., (2015) berpendapat bahwa dengan menggunakan pendekatan siklus kehidupan, dari ibu hamil, balita, usia sekolah, usia kerja, dan usia lanjut, serangkaian usulan intervensi program per kelompok umur tersebut dikemukakan, baik yang bersifat spesifik oleh jajaran kesehatan, maupun yang bersifat sensitif oleh sektor lain diluar kesehatan. Usulan program spesifik oleh jajaran kesehatan, antara lain:

1. Pemberian makanan tambahan TKPM (tinggi kalori, protein dan mikronutrien) untuk ibu hamil,

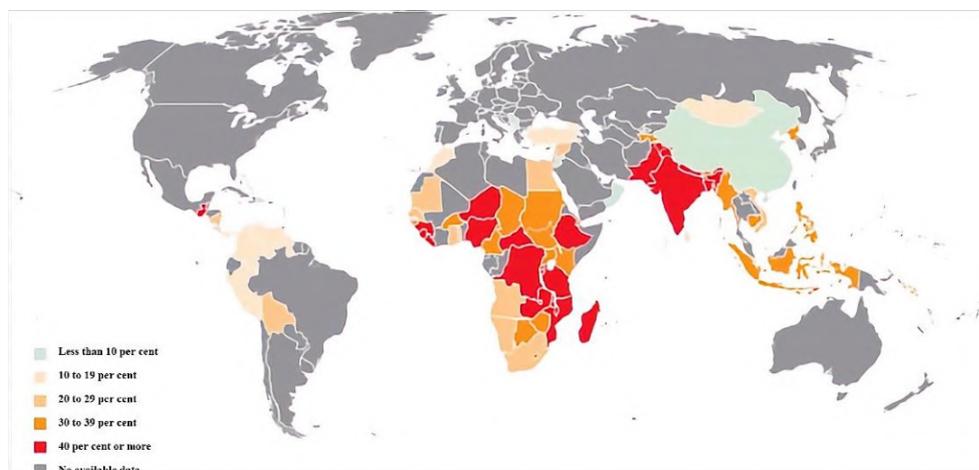
2. Peningkatan kualitas pelayanan kesehatan maternal dan anak,
3. Upaya kesehatan sekolah menjadi program wajib diseluruh PUSKESMAS,
4. Pendidikan kesehatan reproduksi bagi remaja,
5. Penyuluhan intensif tentang program keluarga berencana,
6. Pertolongan persalinan oleh tenaga kesehatan di fasilitas kesehatan,
7. Pemberian ASI eksklusif dan MPASI yang adekuat,
8. Pemantauan pertumbuhan balita,
9. Pemberian makanan tambahan dan mikronutrien bagi balita.

Adapun program sensitif oleh jajaran lintas sektor menurut Trihono et al., (2015) antara lain:

1. Pendidikan gizi di sekolah,
2. Perbaikan kesehatan lingkungan di sekolah dan rumah,
3. Pengentasan kemiskinan,
4. Wajib belajar 12 tahun,
5. Revisi undang - undang Nomor 1 tahun 1974 tentang Perkawinan, yaitu usia menikah dirubah menjadi minimal 20 tahun.

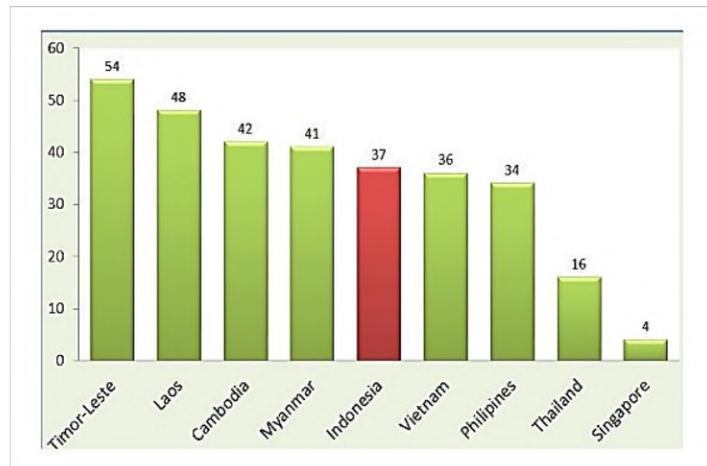
Ancaman permasalahan gizi di dunia, ada 165 juta anak dibawah 5 tahun dalam kondisi *stunting* dan 90% lebih berada di Afrika dan Asia. Target global adalah menurunkan *stunting* sebanyak 40% pada tahun 2025 (WHA, dalam Trihono et al., 2015). Untuk itu dibutuhkan penurunan 3,9% per tahun. Target global yang tercapai adalah menurunkan *stunting* 39,7% dari tahun 1990 menjadi 26,7% pada tahun 2010.

Dalam jangka waktu 20 tahun tersebut dapat diturunkan 1,6% per tahun. Penurunan sangat kecil terjadi di Afrika (40% menjadi 38%). Sedangkan penurunan yang cukup besar terjadi di Asia (dari 49% menjadi 28%), sekitar 2,9% per tahun. Penurunan yang terbesar ada di Tiongkok, pada tahun 1990 sebesar 30% menjadi 10% pada tahun 2011. Gambaran prevalensi *stunting* dapat dilihat pada gambar dibawah ini.



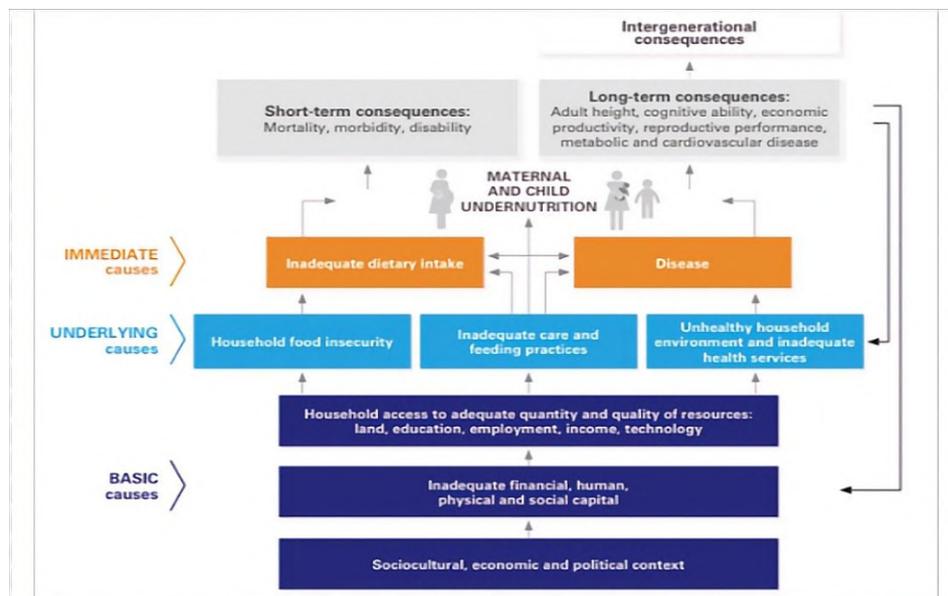
**Gambar 2. 25 Peta *Stunting* dunia, tahun 2007 – 2011**  
**Sumber : Trihono et al., 2015**

Dibandingkan dengan negara - negara di Asia Tenggara, prevalensi balita *stunting* di Indonesia berada tepat diatas Vietnam. Hasil dari *South East Asian Nutrition Survey* (SEANUTS) pada tahun 2010-2011 menempatkan Indonesia sebagai negara yang memiliki jumlah anak balita *stunting* terbesar, jauh diatas Malaysia, Thailand, serta Vietnam.



Gambar 2. 26 Perbandingan prevalensi *stunting* antar beberapa negara ASEAN  
Sumber : Trihono et al., 2015

Trihono et al., (2015) mengatakan bahwa informasi tentang *stunting* diperoleh dengan mengacu pada beberapa faktor dari “*Logical Framework of The Nutritional Problems*” atau dari *Conceptual Framework of The Determinans of The Child Undernutrition*, sebagai berikut:



Gambar 2. 27 Kerangka logis masalah gizi

Dari kerangka tersebut, dibuat kerangka pembahasan yang khusus untuk diperuntukkan kepada masalah *stunting* di Indonesia, yang bentuknya adalah sebagai berikut.



Gambar 2. 28 Kerangka pembahasan *stunting* di Indonesia

Berdasarkan gambar diatas, Trihono et al., (2015) menjelaskan alur pembahasan hasil kajian pada penelitiannya sebagai berikut.

1. Kecenderungan *stunting* di Indonesia, berdasarkan survei yang dilakukan dari tahun 2001 sampai dengan 2013.
2. Besarnya masalah status *stunting*, dalam hal ini jumlah mereka yang mempunyai status gizi *stunting*, termasuk besarnya disparitas antar daerah.
3. Beban yang akan ditimbulkan di masa depan akibat status *stunting* saat ini, baik dalam jangka pendek maupun jangka panjang.
4. Faktor determinan yang mempengaruhi terjadinya status *stunting*, baik yang berupa penyebab langsung, penyebab tak langsung maupun penyebab dasar.

Berdasarkan analisis faktor determinan ini kemudian disampaikan rekomendasi tentang strategi penanggulangan status *stunting*, baik yang berupa

program spesifik oleh jajaran Kementerian Kesehatan maupun program sensitif oleh jajaran non-kesehatan.

### 3. Cara Penilaian Status Gizi Balita

Menurut Trihono et al., (2015) status gizi balita dapat diukur berdasarkan umur, berat badan (BB), dan tinggi badan atau panjang badan (TB/PB). Variabel BB dan TB/PB balita disajikan dalam bentuk tiga indeks antropometri, yaitu BB/U, TB/U, dan BB/TB. Untuk menilai status gizi balita, maka angka berat badan dan tinggi badan setiap balita dikonversikan kedalam nilai standar (*Z-Score*) menggunakan baku antropometri balita WHO 2005. Selanjutnya dengan indikator tersebut ditentukan status gizi balita dengan batasan sebagai berikut:

- a. Klasifikasi status gizi berdasarkan indeks BB/U:
  - 1) Gizi buruk :  $Z\text{-Score} < -3,0$
  - 2) Gizi kurang :  $Z\text{-Score} \geq -3,0$  s/d  $< -2,0$
  - 3) Gizi baik :  $Z\text{-Score} \geq -2,0$
- b. Klasifikasi status gizi berdasarkan indikator TB/U:
  - 1) Sangat pendek :  $Z\text{-Score} < -3,0$
  - 2) Pendek :  $Z\text{-Score} \geq -3,0$  s/d  $< -2,0$
  - 3) Normal :  $Z\text{-Score} \leq -2,0$
- c. Klasifikasi status gizi berdasarkan indikator BB/TB:
  - 1) Sangat kurus :  $Z\text{-Score} < -3,0$
  - 2) Kurus :  $Z\text{-Score} \geq -3,0$  s/d  $< -2,0$
  - 3) Normal :  $Z\text{-Score} \geq -2,0$  s/d  $\leq 2,0$

4) Gemuk : Z-Score  $> 2,0$

d. Klasifikasi status gizi berdasarkan gabungan indikator TB/U dan BB/TB:

1) Pendek-kurus : Z-Score TB/U  $< -2,0$  dan BB/TB  $< -2,0$

2) Pendek-normal : Z-Score TB/U  $< -2,0$  dan BB/TB antara  $-2,0$  s/d  $2,0$

3) Pendek-gemuk : Z-Score TB/U  $< -2,0$  s/d BB/TB  $> 2,0$

4) TB Normal-kurus : Z-Score TB/U  $\geq -2,0$  dan BB/TB  $< -2,0$

5) TB Normal-normal : Z-Score TB/U  $\geq -2,0$  dan BB/TB antara  $-2,0$  s/d  $2,0$

6) TB Normal-gemuk : Z-Score TB/U  $\geq -2,0$  dan BB/TB  $> 2,0$

Perhitungan angka prevalensi dilakukan sebagai berikut:

a. Berdasarkan indikator BB/U:

1) Gizi buruk :  $(\sum \text{Balita gizi buruk} / \sum \text{Balita}) \times 100\%$

2) Gizi kurang :  $(\sum \text{Balita gizi kurang} / \sum \text{Balita}) \times 100\%$

3) Gizi baik :  $(\sum \text{Balita gizi baik} / \sum \text{Balita}) \times 100\%$

b. Berdasarkan indikator TB/U:

1) Sangat pendek :  $(\sum \text{Balita sangat pendek} / \sum \text{Balita}) \times 100\%$

2) Pendek :  $(\sum \text{Balita pendek} / \sum \text{Balita}) \times 100\%$

3) Normal :  $(\sum \text{Balita normal} / \sum \text{Balita}) \times 100\%$

c. Berdasarkan indikator BB/TB:

1) Sangat kurus :  $(\sum \text{Balita sangat kurus} / \sum \text{Balita}) \times 100\%$

- 2) Kurus :  $(\sum \text{Balita kurus} / \sum \text{Balita}) \times 100\%$
- 3) Normal :  $(\sum \text{Balita normal} / \sum \text{Balita}) \times 100\%$
- 4) Gemuk :  $(\sum \text{Balita gemuk} / \sum \text{Balita}) \times 100\%$

d. Berdasarkan gabungan indikator TB/U dan BB/TB:

- 1) Pendek-kurus :  $(\sum \text{Balita pendek-kurus} / \sum \text{Balita}) \times 100\%$
- 2) Pendek-normal :  $(\sum \text{Balita pendek-normal} / \sum \text{Balita}) \times 100\%$
- 3) Pendek-gemuk :  $(\sum \text{Balita pendek-gemuk} / \sum \text{Balita}) \times 100\%$
- 4) TB normal-kurus :  $(\sum \text{Balita normal-kurus} / \sum \text{Balita}) \times 100\%$
- 5) TB normal-normal :  $(\sum \text{Balita normal-normal} / \sum \text{Balita}) \times 100\%$
- 6) TB normal-gemuk :  $(\sum \text{Balita normal-gemuk} / \sum \text{Balita}) \times 100\%$

Dalam penelitian ini ada beberapa istilah status gizi yang digunakan, seperti:

- a. Gizi Kurang : Istilah untuk gabungan gizi buruk dan gizi kurang (*underweight*).
- b. Pendek : Istilah untuk gabungan antara sangat pendek dan pendek (*stunting*)
- c. Kurus : Istilah untuk gabungan antara sangat kurus dan kurus (*wasting*).

Menurut Trihono et al., (2015) Indikator status gizi berdasarkan indeks BB/U memberikan indikasi masalah gizi secara umum. Indikator ini tidak memberikan indikasi tentang masalah gizi yang sifatnya kronis ataupun akut karena berat badan berkorelasi positif dengan umur dan tinggi badan.

Indikator status gizi berdasarkan indeks TB/U memberikan indikasi masalah gizi yang sifatnya kronis sebagai akibat dari keadaan yang berlangsung lama. Misalnya kemiskinan, perilaku hidup tidak sehat, dan asupan makanan kurang dalam jangka waktu lama sejak usia bayi, bahkan semenjak janin.

Indikator status gizi berdasarkan indeks BB/TB memberikan indikasi masalah gizi yang sifatnya akut sebagai akibat dari peristiwa yang terjadi dalam waktu yang tidak lama (singkat). Misalnya terjadi wabah penyakit dan kekurangan makan (kelaparan) yang mengakibatkan anak menjadi kurus. Indikator BB/TB atau IMT/U dapat digunakan untuk identifikasi kurus dan gemuk.

#### **4. Cara Penilaian Status Gizi anak Umur 5-18 Tahun**

Menurut Trihono et al., (2015) indikator status gizi yang digunakan untuk kelompok umur 5-18 tahun didasarkan pada hasil pengukuran antropometri berat badan (BB) dan tinggi badan (TB) yang disajikan dalam bentuk tinggi badan menurut umur (TB/U) dan Indeks Massa Tubuh menurut umur (IMT/U). Berdasarkan baku antropometri WHO 2007 untuk anak umur 5-18 tahun, status gizi ditentukan berdasarkan nilai *Z-Score* TB/U dan IMT/U. Selanjutnya berdasarkan nilai *Z-Score* ini status gizi anak dikategorikan sebagai berikut:

a. Klasifikasi indikator TB/U:

- 1) Sangat pendek :  $Z\text{-Score} < -3,0$
- 2) Pendek :  $Z\text{-Score} \geq -3,0 \text{ s/d } < -2,0$

3) Normal : Z-Score  $\geq -2,0$

b. Klasifikasi indikator IMT/U:

1) Sangat kurus : Z-Score  $< -3,0$

2) Kurus : Z-Score  $\geq -3,0$  s/d  $< -2,0$

3) Normal : Z-Score  $\geq -2,0$  s/d  $\leq 1,0$

4) Gemuk : Z-Score  $> 1,0$  s/d  $\leq 2,0$

5) Obesitas : Z-Score  $> 2,0$

### 5. Cara Penilaian Status Gizi Dewasa (>18 tahun)

Menurut Trihono et al., (2015) status gizi dewasa adalah penilaian status gizi penduduk berumur  $\geq 18$  tahun yang dinilai dengan Indeks Massa Tubuh (IMT). Rumus perhitungan IMT adalah sebagai berikut:

$$\text{IMT} = \text{Berat Badan (kg)} \div \text{Tinggi Badan (m)}^2$$

Trihono et al., (2015) mengatakan bahwa ada batasan IMT yang digunakan untuk menilai status gizi penduduk dewasa adalah sebagai berikut:

- |                      |                            |
|----------------------|----------------------------|
| a. Kategori kurus    | IMT $< 18,5$               |
| b. Kategori normal   | IMT $\geq 18,5$ - $< 24,9$ |
| c. Kategori BB lebih | IMT $\geq 25,0$ - $< 27,0$ |
| d. Kategori obesitas | IMT $\geq 27,0$            |

Berdasarkan kategori tinggi badan, status gizi orang dewasa dikategorikan menjadi pendek bila TB  $< 150,1$  cm (perempuan) dan TB  $< 161,9$  cm (laki-laki). Pada penelitian ini pembahasan difokuskan pada masalah *stunting*, ditinjau dari besaran masalahnya, faktor determinan

penyebabnya dan intervensi apa yang harus dilakukan untuk mengatasinya, baik yang berupa intervensi spesifik oleh jajaran Kementerian Kesehatan maupun intervensi sensitif oleh sector non-kesehatan (Trihono et al., 2015).

## **BAB III**

### **METODE PENELITIAN**

#### **A. Instrumen Penelitian**

Jenis Penelitian yang digunakan adalah penelitian kualitatif atau *qualitative research* yang bertujuan untuk memahami kondisi suatu konteks dengan mengarahkan pada pendeskripsian secara rinci dan mendalam mengenai potret situasi dan kondisi dalam suatu konteks yang alami, tentang apa yang sebenarnya terjadi sesuai dengan keadaan nyata.

##### **1. Tempat Penelitian**

Penelitian ini dilakukan dengan mengolah dan menyusun data - data yang telah didapat, Penelitian dilakukan di Kecamatan Tampan, Kota Pekanbaru, Provinsi Riau.

##### **2. Alat Penelitian**

Dalam pelaksanaan penelitian ini, ada beberapa alat bantu yang dibutuhkan agar terlaksananya penelitian ini dengan lebih efektif dan efisien, antara lain:

###### **a. Perangkat Keras (*Hardware*)**

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut.

- 1) *Mini Computer* atau Laptop Acer Aspire, dengan spesifikasi:
  - a) *System Model* Aspire A515-44,
  - b) *Operating System* Windows 10 Home Single Language 64-bit,
  - c) *Processor* AMD Ryzen 5 4500U (6 CPU) 2,4GHz,
  - d) *Memory* 8192 MB.

- 2) *Mouse, Keyboard, dan Speakers,*
- 3) *Cooling Fan,*
- 4) *Printer.*

b. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam membangun penelitian ini adalah sebagai berikut.

- 1) *Operating System Windows 10 Home Single Language,*
- 2) Aplikasi XAMPP,
- 3) PHP sebagai bahasa pemrograman dasar yang digunakan,
- 4) *MySQL database,*
- 5) Mozilla Firefox dan Google Chrome sebagai *web browser* yang digunakan,
- 6) StarUML untuk merancang diagram UML,
- 7) Snip & Sketch untuk membuat *capture* gambar,
- 8) Mendeley sebagai aplikasi sitasi dan referensi,
- 9) Visual Studio Code dan Sublime Text 3 sebagai aplikasi *code editor,*
- 10) Microsoft Office Home and Student 2019 untuk mengolah data penelitian.

## **B. Teknik Pengumpulan Data**

### **1. Wawancara**

Menurut Raharjo (2011), Wawancara ialah proses komunikasi atau interaksi untuk mengumpulkan informasi dengan cara tanya jawab antara

peneliti dengan narasumber atau subjek penelitian. Dengan kemajuan teknologi saat ini, wawancara bisa dilakukan tanpa tatap muka, melalui perangkat elektronik digital yang telah hadir pada masa ini, hal itu dapat dilakukan dengan menggunakan media telekomunikasi.

Pada hakikatnya wawancara merupakan kegiatan untuk memperoleh informasi secara mendalam tentang isu atau tema yang diangkat dalam penelitian. Pada penelitian ini, peneliti melakukan wawancara tidak langsung terhadap narasumber yang bersangkutan.

## **2. Dokumen atau Studi Pustaka**

Peneliti juga mengumpulkan data penelitian melalui data yang tersimpan seperti jurnal, artikel, buku dan beberapa penelitian yang telah ada sebelumnya. Menurut Raharjo (2011), teknik pengumpulan data atau studi pustaka berupa pengumpulan data melalui dokumen untuk menggali informasi yang terjadi dimasa lampau. Peneliti perlu memiliki kepekaan teoretik untuk memaknai semua dokumen tersebut sehingga tidak sekadar barang yang tidak bermakna.

## **C. Jenis Data**

### **1. Data Primer**

Menurut Hanke dan Reitsch (dalam Hamid, 2015) Data primer biasanya diperoleh dengan survei lapangan yang menggunakan semua metode pengumpulan data orisinal. Data primer juga dapat didefinisikan sebagai data yang dikumpulkan dari sumber - sumber yang asli. Dalam penelitian

ini, data primer yang dikumpulkan berupa hasil wawancara dan kuesioner dengan ahli gizi mengenai kasus *stunting* yang telah dipilih.

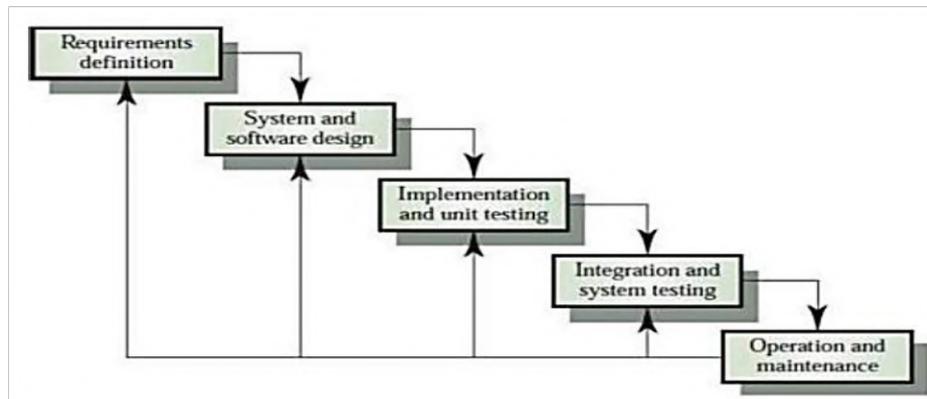
## **2. Data Sekunder**

Menurut Hanke dan Reitsch (dalam Hamid, 2015) Data sekunder adalah data diperoleh dari data yang telah dikumpulkan oleh lembaga pengumpul data dan dipublikasikan kepada masyarakat pengguna. Dapat disimpulkan bahwa data sekunder adalah pengambilan data dari data - data yang telah ada sebelumnya. Pada penelitian ini, data sekunder yang dikumpulkan diperoleh dari beberapa jurnal, artikel, buku dan penelitian yang telah ada sebelumnya.

## **D. Teknik Pengembangan Sistem**

### **1. Pengembangan Model *Waterfall***

Adapun metode pengembangan perangkat lunak pada penelitian ini menggunakan metode pengembangan *waterfall* model. Menurut Informasi et al., (2018), dalam *waterfall* terdapat beberapa tahapan utama yang menggambarkan aktivitas pengembangan perangkat lunak. Alasan peneliti menggunakan metode *waterfall* model karena tahap - tahap dalam pengembangan sistem pada *waterfall* model tergambar secara terstruktur dan jelas. Gambar berikut merupakan tahapan yang ada dalam *waterfall* model.



Gambar 3. 1 Waterfall Model

Berikut penjelasan tahapan - tahapan *waterfall* model menurut Informasi et al., (2018), antara lain:

a. Analisis Kebutuhan

Pada tahap ini dilakukan pengumpulan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Dalam pengumpulan data pada penelitian ini dilakukan dengan teknik wawancara, studi pustaka dan kuesioner.

b. Desain Sistem

Proses desain adalah proses multi langkah yang berfokus pada empat atribut yaitu: struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail prosedural. Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

c. Implementasi

Pembuatan program atau hasil rancangan ke dalam bahasa

pemrograman tertentu. Penulisan kode program sesuai dengan desain yang sudah ditentukan, sehingga menghasilkan aplikasi yang bermanfaat.

d. Pengujian Sistem

Melakukan pengujian sistem dengan cara agar sistem valid dan dapat digunakan dengan baik.

e. Pemeliharaan

Mengaplikasikan sistem yang sudah terintegrasi dan melakukan perawatan atau perbaikan terhadap kekeliruan.

### **E. Model Penelitian**

Metode penelitian digunakan sebagai acuan dalam melakukan penelitian agar hasil penelitian yang didapatkan tidak mengalami kekeliruan dari tujuan yang telah diharapkan. Penelitian ini menggunakan model penelitian OOAD (*Object Oriented Analysis Design*).

Menurut Informasi et al., (2018), Model OOAD dapat dibagi dua tahapan utama yaitu: OOA (*Object Oriented Analysis*) merupakan tahap pengidentifikasian tujuan - tujuan dari aplikasi atau sistem dan dilakukan pengidentifikasian syarat - syarat atau kebutuhan informasi bagi sistem yang diperoleh dari pengidentifikasian tujuan - tujuan tersebut. Selanjutnya OOD (*Object Oriented Design*) merupakan tahap perancangan terhadap proses - proses yang akan terjadi didalam sistem meliputi perancangan aplikasi. Peneliti menggunakan perancangan UML (*Unified Modeling Language*) sebagai gambaran perancangan sistem.

## F. Data Gejala *Stunting*

Data gejala *stunting* dapat dilihat pada tabel dibawah ini.

**Tabel 3. 1 Data Gejala *Stunting***

<b>No.</b>	<b>Kode Gejala</b>	<b>Keterangan</b>
1.	G001	Wajah tampak lebih muda dari anak seusianya
2.	G002	Pertumbuhan tubuh dan gigi yang terlambat
3.	G003	Memiliki kemampuan fokus dan memori belajar yang tidak baik
4.	G004	Pubertas yang lambat
5.	G005	Saat anak usia 8-10 tahun, anak cenderung lebih pendiam dan tidak banyak melakukan kontak mata dengan orang sekitar
6.	G006	Berat badan lebih ringan dari anak seusianya

## **BAB IV**

### **PERANCANGAN DAN IMPLEMENTASI**

#### **A. Analisis Sistem**

Berdasarkan dari prosedur penelitian yang telah dirangkum, pada analisis sistem untuk Sistem Pakar penyakit *stunting* ini dilakukan pengumpulan data dan analisis kebutuhan. Pengumpulan data yaitu, proses yang dilakukan untuk mengumpulkan informasi - informasi yang berkaitan dalam pembuatan aplikasi Sistem Pakar mendiagnosa penyakit *stunting* yaitu, berupa data gejala, data penyakit, dan data hubungan gejala dan penyakit. Analisis kebutuhan terdiri dari kebutuhan masukan, dan kebutuhan keluaran. Analisis kebutuhan masukan (*input*) terdiri dari dua, yaitu *admin* dan masukan *user*. Analisis kebutuhan keluaran yaitu berupa tampilan akhir dari proses diagnosa Sistem Pakar mengenai penyakit *stunting* pada anak.

#### **B. Pengumpulan Data**

Data – data yang diperoleh selama proses pengumpulan data terdiri dari data gejala, data penyakit, dan data hubungan gejala dan penyakit. Data – data tersebut diperoleh dengan melakukan wawancara dan kuesioner dengan tenaga kesehatan dan ahli gizi yang berhubungan dengan penyakit *stunting* pada anak.

##### **1. Data Gejala**

Data – data gejala yang digunakan dalam Sistem Pakar penyakit *stunting* ini berjumlah 6 gejala. Adapun data – data gejala tersebut dapat dilihat pada tabel 4.1 dibawah ini :

Tabel 4. 1 Kode data dan data gejala penyakit *stunting*

No.	Kode Gejala	Keterangan
1.	G001	Merasa kelelahan yang berat dan berlebihan
2.	G002	Memiliki kulit kering dan bersisik
3.	G003	Memiliki rambut yang terlihat kusam dan kering
4.	G004	Memiliki perut yang buncit
5.	G005	Mengalami hilangnya massa otot
6.	G006	Memiliki pembengkakan dibawah kulit (edema)
7.	G007	Mengalami perubahan mood yang sering
8.	G008	Susah menambah berat dan tinggi badan
9.	G009	Mengalami kondisi gigi mudah tanggal atau copot
10.	G010	Mengalami masalah perhambatan pada pertumbuhan
11.	G011	Mengalami diare kronis
12.	G012	Memiliki infeksi saluran pernapasan
13.	G013	Mengalami pusing yang terus-menerus
14.	G014	Sering merasa sangat lapar
15.	G015	Mengalami perubahan warna kulit yang pucat dan menguning
16.	G016	Mengalami detak jantung yang tidak teratur
17.	G017	Memiliki rasa dingin pada tangan dan kaki
18.	G018	Mengalami nyeri pada dada
19.	G019	Memiliki wajah yang tampak lebih muda dari anak seusianya
20.	G020	Memiliki kemampuan fokus dan memori yang kurang baik
21.	G021	Cenderung lebih pendiam dan tidak melakukan kontak mata dengan orang disekitarnya
22.	G022	Berat badan lebih ringan untuk anak seusianya

## 2. Data penyakit

Data penyakit yang diolah dalam Sistem Pakar penyakit *stunting* dapat dilihat pada tabel dibawah ini.

Tabel 4. 2 Data penyakit *stunting*

No.	Kode Penyakit	Keterangan
1.	P001	Kwashiorkor
2.	P002	Marasmus
3.	P003	Anemia
4.	P004	Stunting

Dari pengetahuan berupa gejala dan penyakit *stunting* pada anak, dapat dibuat basis pengetahuan berupa hubungan atau keterkaitan yang ada antara gejala dan penyakit *stunting* pada anak. Basis pengetahuan berikut dapat dilihat pada tabel dibawah ini.

Tabel 4. 3 Tabel basis pengetahuan penyakit *stunting* pada anak

Kode Gejala (G)	Kode Penyakit (P)			
	001	002	003	004
G001	*		*	
G002	*	*		
G003	*	*		
G004	*			
G005	*	*		
G006	*			
G007	*			
G008	*			*
G009	*			
G010	*	*		*
G011		*		
G012		*		
G013		*	*	
G014		*		
G015			*	
G016			*	
G017			*	
G018			*	

Kode Gejala (G)	Kode Penyakit (P)			
	001	002	003	004
G019				*
G020				*
G021				*
G022				*

### C. Desain Sistem

Desain sistem dalam Sistem Pakar ini dibagi menjadi beberapa bagian, yaitu perancangan UML (*Unified Modelling Language*), perancangan antarmuka (*Interface*), dan perancangan *database*.

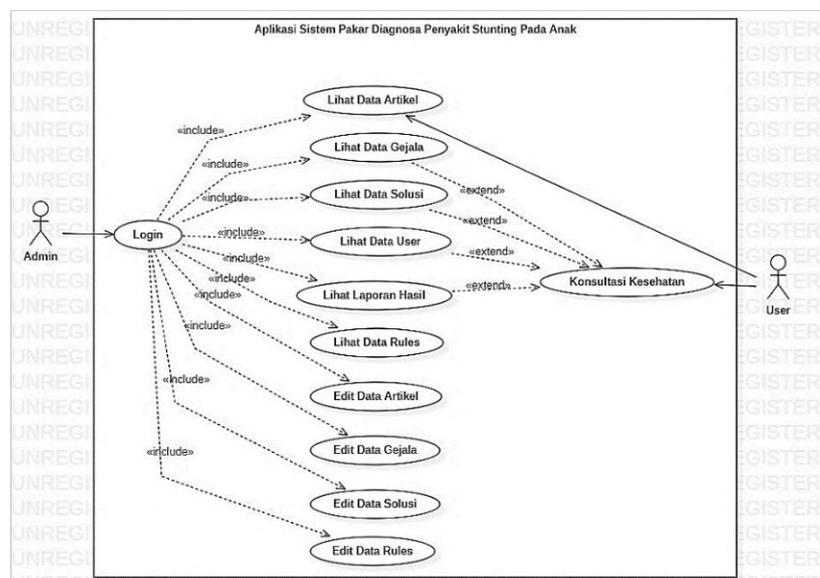
#### 1. Perancangan UML (*Unified Modelling Language*)

Pada sistem ini perancangan yang digunakan adalah perancangan diagram UML (*Unified Modelling Language*). Diagram UML dibuat dengan menggunakan aplikasi *StarUML v4.1.5*. Pada Sistem Pakar ini menggunakan empat jenis diagram UML. Berikut adalah diagram – diagram UML yang digunakan untuk membangun Sistem Pakar diagnosa penyakit *stunting*:

##### a. *Use Case Diagram*

Menurut Shalahuddin (dalam Heriyanto, 2018) mengungkapkan bahwa *Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah

sistem dan siapa saja yang berhak menggunakan fungsi - fungsi tersebut. Dalam sistem ini pengguna aplikasi terdiri dari *user* atau pasien dan *admin*. Perilaku pengguna adalah apa saja yang dapat dilakukan terhadap sistem. Perilaku pengguna dapat dilihat lebih jelas pada gambar berikut.



Gambar 4. 1 Perancangan *Use Case Diagram*

Berikut merupakan penjelasan perancangan *Use Case Diagram* pada gambar diatas:

Tabel 4. 4 Penjelasan perancangan *Use Case Diagram Admin*

Aktor	Input	Nama <i>Use Case</i>	Deskripsi
Admin	Login dengan memasukkan ID admin dan password	Lihat data artikel	<i>Use Case</i> ini berfungsi untuk melihat data artikel yang akan muncul dihalaman artikel user.
		Lihat data gejala	<i>Use Case</i> ini berfungsi untuk melihat data gejala dari penyakit.

<b>Aktor</b>	<b>Input</b>	<b>Nama Use Case</b>	<b>Deskripsi</b>
<i>Admin</i>	<i>Login dengan memasukkan ID admin dan password</i>	Lihat data solusi	<i>Use Case</i> ini berfungsi untuk melihat data solusi berdasarkan kriteria gejala dari penyakit.
		Lihat data <i>user</i>	<i>Use Case</i> ini berfungsi untuk melihat data pribadi yang <i>input</i> oleh <i>user</i> .
		Lihat laporan hasil	<i>Use Case</i> ini berfungsi untuk melihat laporan hasil diagnosa Sistem Pakar.
		Lihat data <i>rules</i>	<i>Use Case</i> ini berfungsi melihat data aturan atau <i>rules</i> pada basis pengetahuan Sistem Pakar.
		Edit data artikel	<i>Use Case</i> ini berfungsi untuk melakukan olah data pada data artikel.
		Edit data gejala	<i>Use Case</i> ini berfungsi untuk melakukan olah data pada data gejala.
		Edit data solusi	<i>Use Case</i> ini berfungsi untuk melakukan olah data pada data solusi.
		Edit data <i>rules</i>	<i>Use Case</i> ini berfungsi untuk melakukan olah data pada data <i>rules</i> .

Tabel 4. 5 Penjelasan perancangan *Use Case Diagram User*

<b>Aktor</b>	<b>Input</b>	<b>Nama Use Case</b>	<b>Deskripsi</b>
<i>User</i>	Konsultasi kesehatan	Lihat data artikel	<i>Use Case</i> ini berfungsi untuk melihat data artikel yang akan muncul dihalaman artikel <i>user</i> .
<b>Aktor</b>	<b>Input</b>	<b>Nama Use Case</b>	<b>Deskripsi</b>

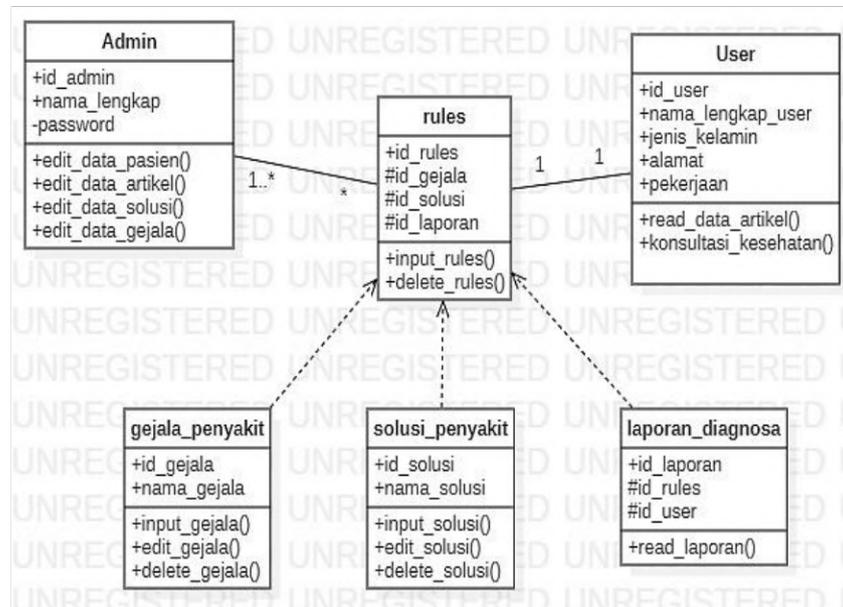
<i>User</i>	Konsultasi kesehatan	Lihat data gejala	<i>Use Case</i> ini berfungsi untuk melihat data – data gejala dari penyakit setelah melakukan konsultasi dengan sistem.
		Lihat data solusi	<i>Use Case</i> ini berfungsi untuk melihat data – data solusi berdasarkan kriteria gejala dari penyakit setelah melakukan konsultasi dengan sistem.
		Lihat data user	<i>Use Case</i> ini berfungsi untuk melihat data pribadi yang dimasukkan oleh user.
		Lihat laporan hasil	<i>Use Case</i> ini berfungsi untuk melihat laporan hasil diagnosa Sistem Pakar.

b. *Class Diagram*

Menurut Rossa (dalam Heriyanto, 2018) *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas - kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel - variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi - fungsi yang dimiliki oleh suatu kelas.

*Class Diagram* dapat memberi pandangan yang lebih besar tentang suatu sistem dengan menunjukkan kelas dan hubungannya. Dapat dikatakan bahwa *Class Diagram* bersifat statis, alasannya karena *Class Diagram* tidak menggambarkan apa yang terjadi jika terkait, melainkan menggambarkan hubungan yang terjadi. Berikut adalah *Class Diagram*

yang digunakan untuk merancang aplikasi Sistem Pakar ini.

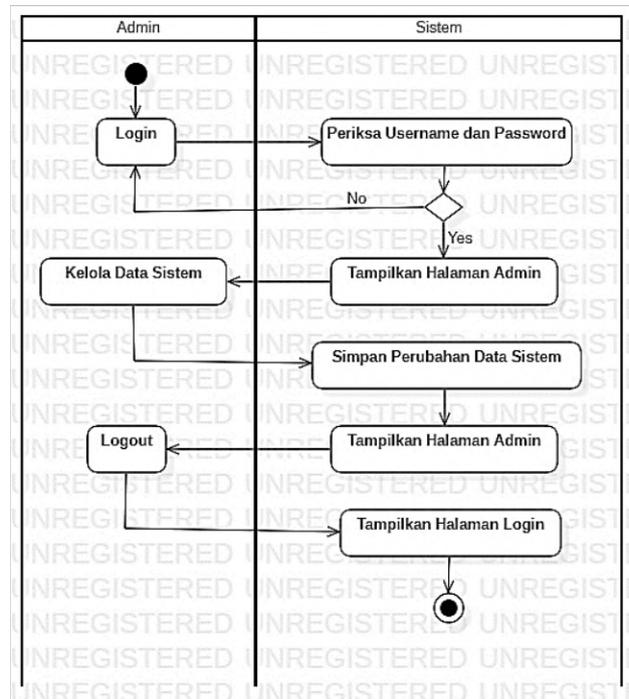


Gambar 4. 2 Perancangan *Class Diagram*

### c. *Activity Diagram*

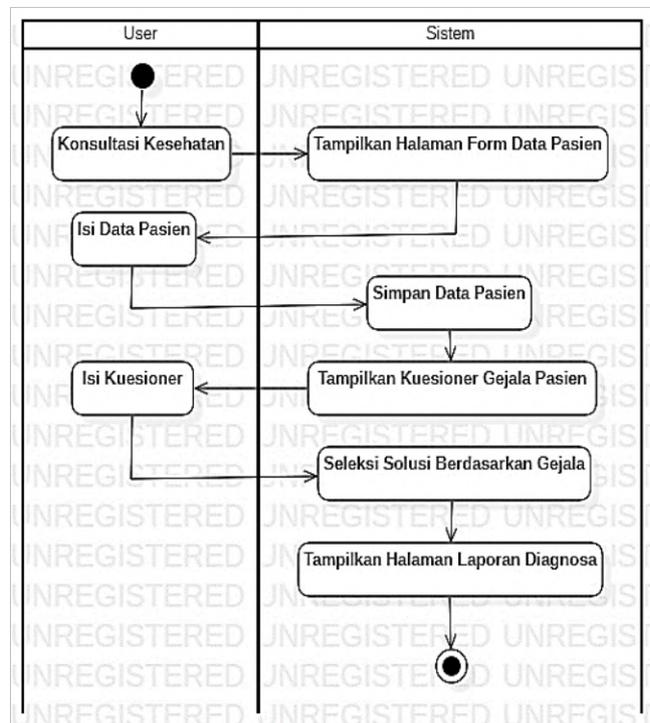
Rosa (dalam Heriyanto, 2018) berpendapat bahwa *Activity Diagram* menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem. Berikut adalah *activity diagram* yang digunakan dalam perancangan aplikasi Sistem Pakar ini.

#### 1) *Activity Diagram Admin*



Gambar 4.3 Perancangan Activity Diagram Admin

## 2) Activity Diagram User



Gambar 4.4 Perancangan Activity Diagram User

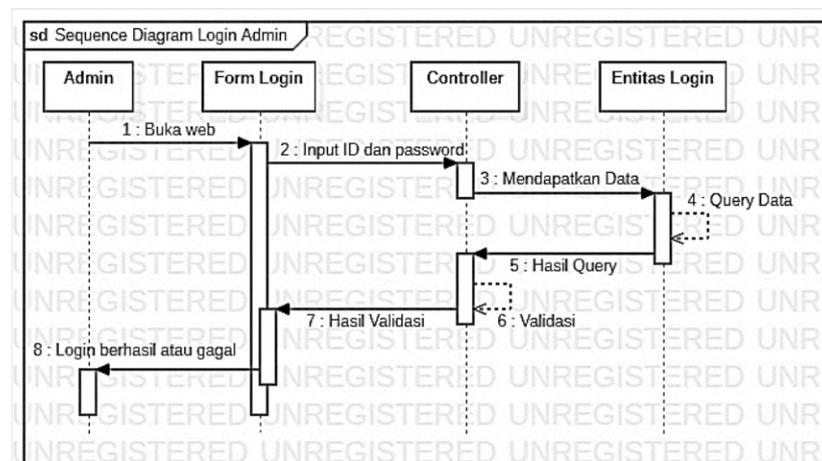
## d. Sequence Diagram

Jurdam (dalam Heriyanto, 2018) mengatakan *Sequence Diagram*

adalah *tools* atau alat yang sangat populer dalam pengembangan sistem informasi secara *object-oriented* untuk menampilkan interaksi antar objek. Berdasarkan definisi tersebut, bahwa *Sequence Diagram* adalah *tools* yang digunakan dalam pengembangan sistem.

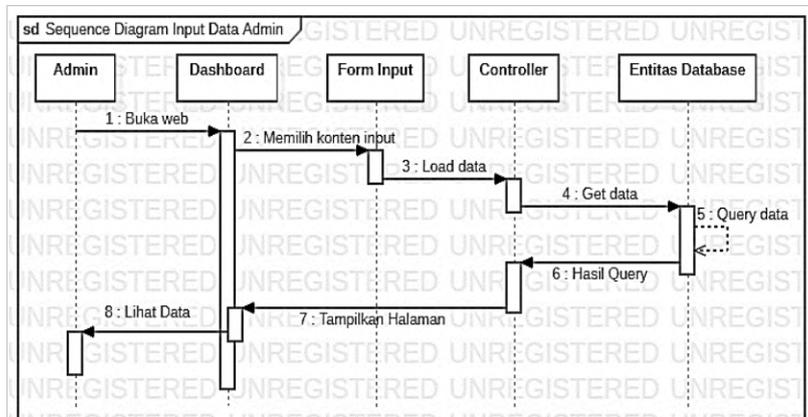
*Sequence Diagram* termasuk dalam jenis diagram interaktif. Sederhananya, diagram urutan adalah jenis diagram interaktif yang menggambarkan bagaimana dan dalam urutan apa sekelompok objek bekerja bersama. *Sequence Diagram* adalah diagram interaksi yang menjelaskan secara rinci bagaimana suatu aktivitas *engineering* dilakukan. Diagram urutan bekerja dengan menangkap interaksi antara objek dalam konteks kolaboratif. Berikut adalah perancangan *sequence diagram* yang digunakan dalam membangun aplikasi Sistem Pakar diagnosa penyakit *stunting*.

#### 1) *Sequence Diagram Login Admin*



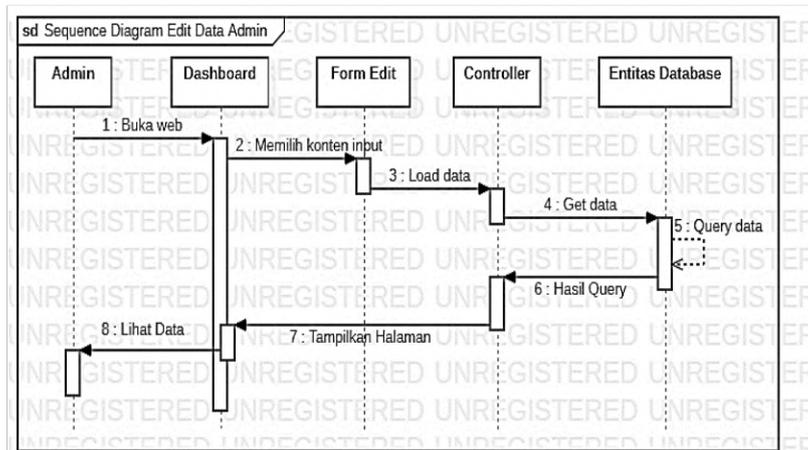
Gambar 4.5 Perancangan *Sequence Diagram Login Admin*

#### 2) *Sequence Diagram Input data Admin*



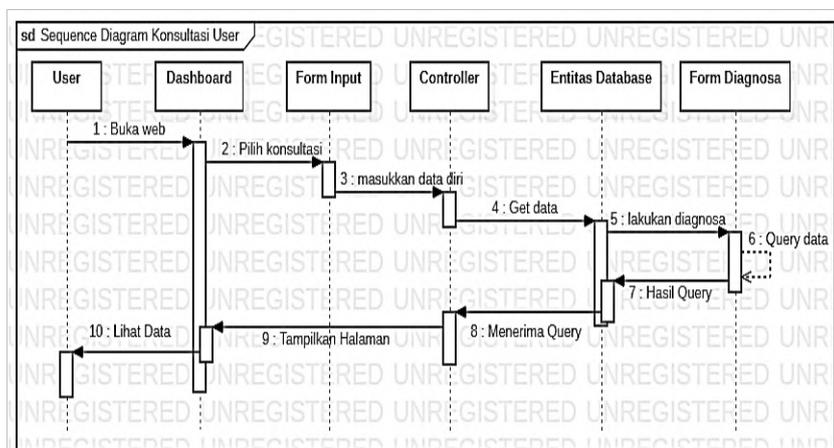
Gambar 4. 6 Perancangan Sequence Diagram Input Data Admin

3) Sequence Diagram Edit data Admin



Gambar 4. 7 Perancangan Sequence Diagram Edit Data Admin

4) Sequence Diagram Konsultasi User



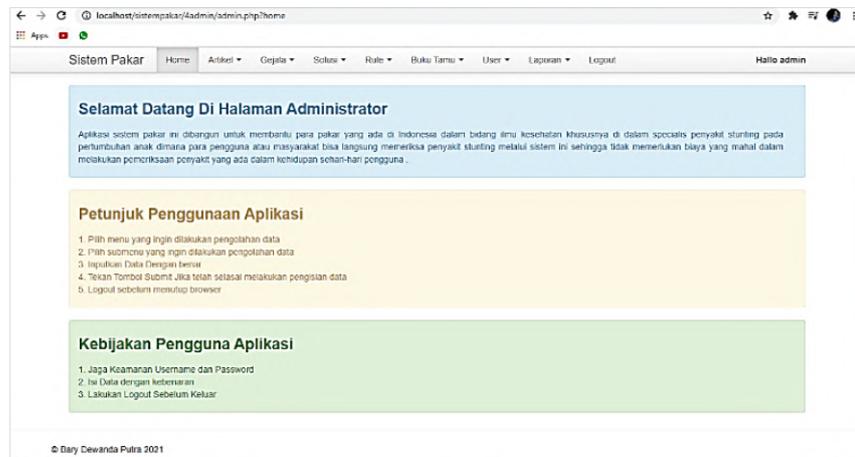
Gambar 4. 8 Perancangan Sequence Diagram Konsultasi User

## 2. Perancangan *Interface Web*

Berikut adalah rancangan *interface* aplikasi Sistem Pakar diagnosa penyakit *stunting*.

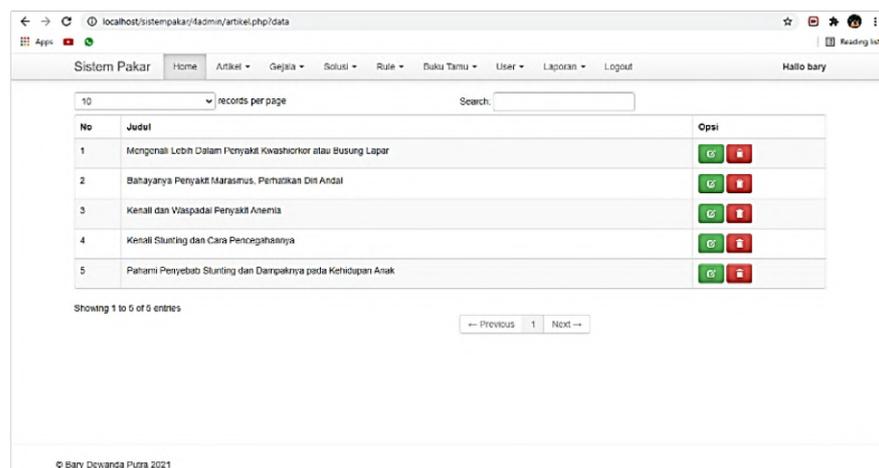
### a. *Interface* sistem pada *Admin*

#### 1) Halaman utama *admin*



Gambar 4. 9 Halaman utama *Admin*

#### 2) Halaman artikel kesehatan



Gambar 4. 10 Halaman artikel kesehatan

### 3) Halaman *form input* artikel kesehatan

The screenshot shows the 'Entri Artikel' form in the 'Sistem Pakar' application. The form is titled 'Entri Artikel' and contains several input fields: 'Judul Artikel', 'Isi Artikel' (with a rich text editor), 'Gambar' (with a 'Choose File' button), 'Keyword', 'Deskripsi', and 'Status' (with a dropdown menu). A green 'SIMPAN DATA' button is located at the bottom of the form. The application's navigation menu includes 'Home', 'Artikel', 'Gejala', 'Solusi', 'Rule', 'Buku Tamu', 'User', 'Laporan', and 'Logout'. The user is logged in as 'Halo admin'.

Gambar 4. 11 Halaman *form input* artikel kesehatan

### 4) Halaman data gejala

The screenshot shows the 'Halaman data gejala' in the 'Sistem Pakar' application. It displays a table with columns for 'No', 'Kode Gejala', 'Nama Gejala', and 'Ops'. The table lists 10 symptoms with their respective codes and names. A search bar and pagination controls are also visible.

No	Kode Gejala	Nama Gejala	Ops
	G001	Merasa lelah yang berat dan berlebihan	<input type="checkbox"/> <input type="checkbox"/>
	G002	Memiliki kulit kering dan bersisik	<input type="checkbox"/> <input type="checkbox"/>
	G003	Memiliki rambut yang terlihat kusam dan kering	<input type="checkbox"/> <input type="checkbox"/>
	G004	Memiliki penis yang buncit	<input type="checkbox"/> <input type="checkbox"/>
	G005	Mengalami hilangnya massa otot	<input type="checkbox"/> <input type="checkbox"/>
	G006	Memiliki pembengkakan di bawah kulit (edema)	<input type="checkbox"/> <input type="checkbox"/>
	G007	Mengalami perubahan mood yang sering	<input type="checkbox"/> <input type="checkbox"/>
	G008	Susah menambah berat dan tinggi badan	<input type="checkbox"/> <input type="checkbox"/>
	G009	Mengalami kondisi gigi mudah tanggal atau copot	<input type="checkbox"/> <input type="checkbox"/>
	G010	Mengalami masalah pertumbuhan pada pertumbuhan	<input type="checkbox"/> <input type="checkbox"/>

Showing 1 to 10 of 22 entries

Navigation: Previous 1 2 3 Next

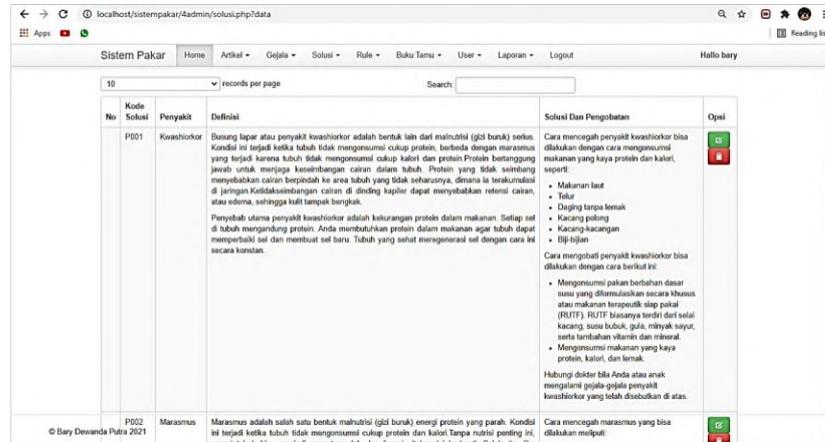
Gambar 4. 12 Halaman data gejala

### 5) Halaman *form input* data gejala

The screenshot shows the 'INPUT GEJALA PENYAKIT' form in the 'Sistem Pakar' application. The form is titled 'INPUT GEJALA PENYAKIT' and contains two input fields: 'Kode' (with the value 'G040') and 'Nama Gejala' (with a rich text editor). A green 'Simpan' button and a red 'Batal' button are located at the bottom of the form. The application's navigation menu includes 'Home', 'Artikel', 'Gejala', 'Solusi', 'Rule', 'Buku Tamu', 'User', 'Laporan', and 'Logout'. The user is logged in as 'Halo admin'.

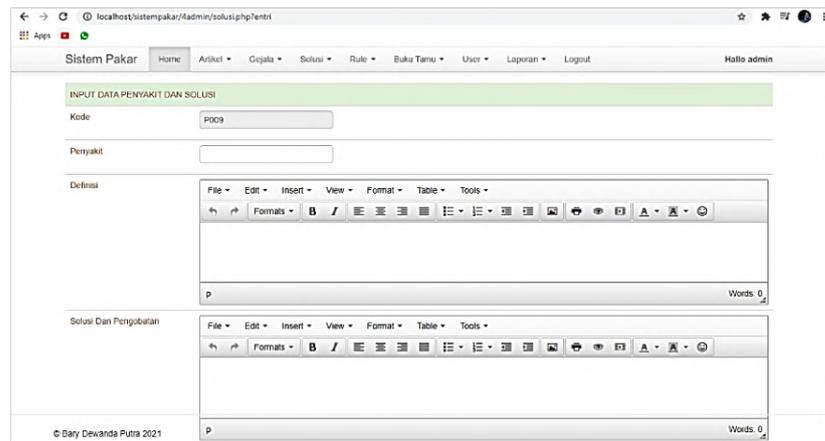
Gambar 4. 13 Halaman *form input* data gejala

6) Halaman data solusi



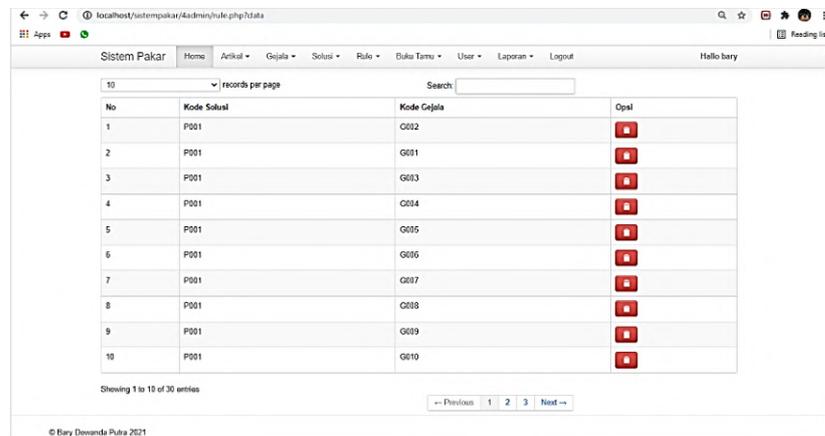
Gambar 4. 14 Halaman data solusi

7) Halaman form input data solusi



Gambar 4. 15 Halaman form input data solusi

8) Halaman data rules



Gambar 4. 16 Halaman data rules

9) Halaman untuk menentukan aturan *rules*

RULE GEJALAN PENYAKIT

Nama Penyakit Dan Solusi :

Daftar Gejala:

- (G001) Merasa kelelahan yang berat dan berlebihan
- (G002) Memiliki kulit kering dan bersisik
- (G003) Memiliki rambut yang terlihat kusam dan kering
- (G004) Memiliki perut yang buncit
- (G005) Mengalami hilangnya massa otot
- (G006) Memiliki pembengkakan dibawah kulit (edema)
- (G007) Mengalami perubahan mood yang sering

© Bary Dewanda Putra 2021 (G000)

Gambar 4. 17 Halaman untuk menentukan aturan *rules*

## 10) Halaman data buku tamu

No	Nama	Email	Isi	Aksi
1	bulan	bulan@gmail.com	Terima kasih website ini sangat bermanfaat	
2	Shena	shenaamalia12@gmail.com	Hasil yang memuaskan	

Showing 1 to 2 of 2 entries

© Bary Dewanda Putra 2021

Gambar 4. 18 Halaman data buku tamu

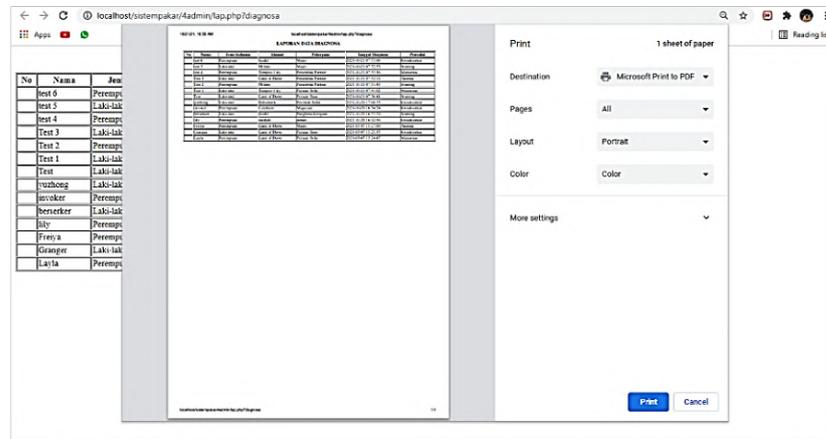
11) Halaman data *admin*

No	Nama Lengkap	Username	Ops
1	Bary Dewanda Putra	bary	
2	Administrator	admin	

© Bary Dewanda Putra 2021

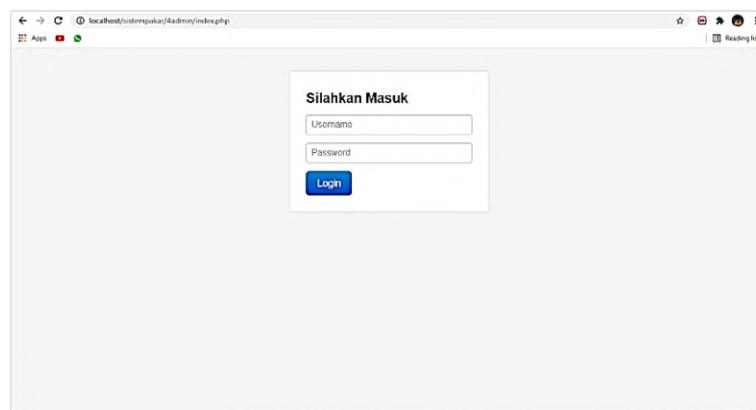
Gambar 4. 19 Halaman data *Admin*

## 12) Halaman laporan hasil diagnosa



Gambar 4. 20 Halaman laporan hasil diagnosa

## 13) Halaman *Login Admin*



Gambar 4. 21 Halaman *Login Admin*

### b. *Interface sistem pada User*

#### 1) Halaman utama *User*



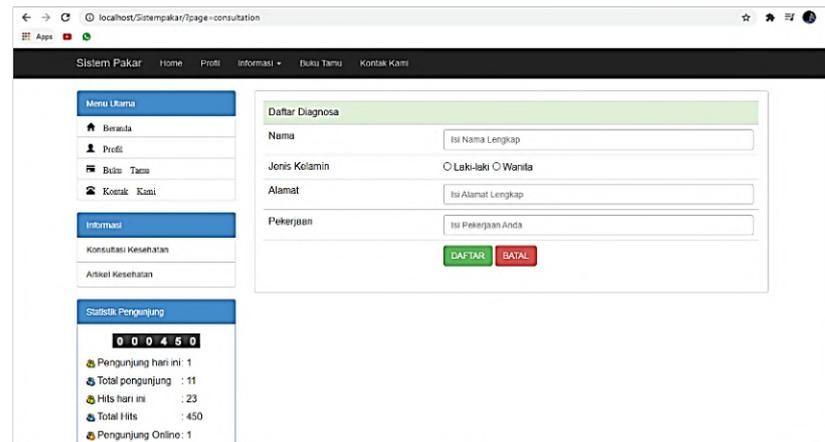
Gambar 4. 22 Halaman utama *User*

## 2) Halaman profil sistem pada *User*



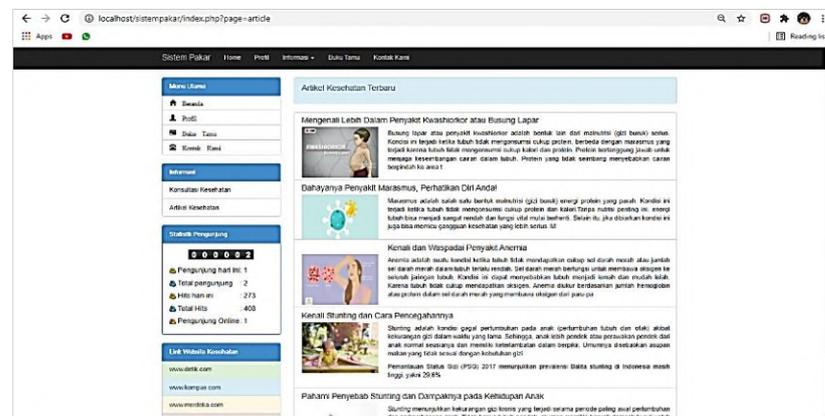
Gambar 4. 23 Halaman profil sistem pada *User*

## 3) Halaman *form input* konsultasi *User*



Gambar 4. 24 Halaman *form input* konsultasi *User*

## 4) Halaman artikel kesehatan pada *User*



Gambar 4. 25 Halaman artikel kesehatan pada *User*

5) Halaman buku tamu *User*

The screenshot shows a web browser window with the URL `localhost/SistemPakar/?page=guest`. The page has a dark header with navigation links: **Sistem Pakar**, Home, Profil, Informasi, **Buku Tamu**, and Kontak Kami. On the left, there is a sidebar with three sections: 'Menu Utama' (Beranda, Profil, Buku Tamu, Kontak Kami), 'Informasi' (Konsultasi Kesehatan, Artikel Kesehatan), and 'Statistik Pengunjung' (000453, Pengunjung hari ini: 1, Total pengunjung: 11, Hits hari ini: 26, Total Hits: 453, Pengunjung Online: 1). The main content area is titled 'BUKU TAMU' and contains a form with input fields for 'Nama', 'Email', and 'Isi Pesan', and a 'KIRIM' button.

Gambar 4. 26 Halaman buku tamu *User*

## 6) Halaman informasi kontak Admin

The screenshot shows a web browser window with the URL `localhost/SistemPakar/?page=contact`. The page has a dark header with navigation links: **Sistem Pakar**, Home, Profil, Informasi, **Kontak Kami**, and Buku Tamu. On the left, there is a sidebar with three sections: 'Menu Utama' (Beranda, Profil, Buku Tamu, Kontak Kami), 'Informasi' (Konsultasi Kesehatan, Artikel Kesehatan), and 'Statistik Pengunjung' (000454, Pengunjung hari ini: 1, Total pengunjung: 11, Hits hari ini: 27, Total Hits: 454, Pengunjung Online: 1). The main content area is titled 'Kontak Kami' and features an illustration of a doctor and a patient. Below the illustration, it says 'Info lebih lanjut: examplesturtung@gmail.com' and '+62-822-xxxx-xxxx'.

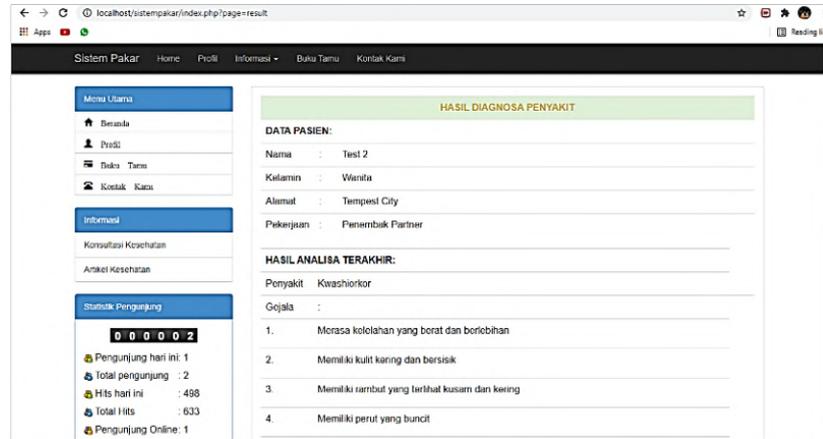
Gambar 4. 27 Halaman informasi kontak *Admin*

## 7) Halaman proses diagnosa penyakit

The screenshot shows a web browser window with the URL `localhost/SistemPakar/?page=diagnosa`. The page has a dark header with navigation links: **Sistem Pakar**, Home, Profil, Informasi, Buku Tamu, and Kontak Kami. On the left, there is a sidebar with three sections: 'Menu Utama' (Beranda, Profil, Buku Tamu, Kontak Kami), 'Informasi' (Konsultasi Kesehatan, Artikel Kesehatan), and 'Statistik Pengunjung' (000468, Pengunjung hari ini: 1, Total pengunjung: 15, Hits hari ini: 5, Total Hits: 468, Pengunjung Online: 1). The main content area is titled 'JAWABLAH PERTANYAAN BERIKUT .' and contains a question: 'Apakah Kurang di komunikasi dan Interaksi sosial?'. Below the question are two radio buttons: 'Benar (YA)' (selected) and 'Salah (TIDAK)'. A 'LANJUT >>' button is located below the radio buttons.

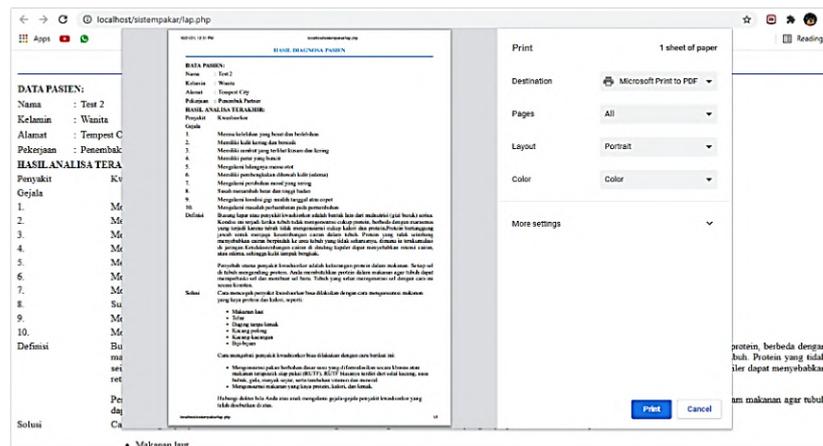
Gambar 4. 28 Halaman proses diagnosa penyakit

8) Halaman hasil diagnosa penyakit



Gambar 4. 29 Halaman hasil diagnosa penyakit

9) Halaman cetak laporan diagnosa pasien



Gambar 4. 30 Halaman cetak laporan diagnosa pasien

3. Perancangan Database

Menurut Lubis (2016) Basis data merupakan kumpulan *file - file* data yang dibentuk dengan suatu relasi yang membuatnya saling terhubung dan dapat diungkapkan dengan catatan serta bersifat independen. Berikut adalah rancangan *database* yang berisi tabel data yang digunakan.

a. Tabel Admin

Tabel Admin digunakan untuk menyimpan data tentang *admin*. Struktur tabel *admin* dapat dilihat pada tabel berikut.

Tabel 4. 6 Tabel *database Admin*

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
Id	<i>Integer</i>	15	<i>Primary Key</i>
nmuser	<i>Varchar</i>	25	-
Nmlogin	<i>Varbinary</i>	25	-
Pslogin	<i>Varchar</i>	55	-
Level	<i>Integer</i>	1	-

## b. Tabel Analisa Hasil

Tabel Analisa Hasil digunakan untuk menyimpan data mengenai hasil diagnosa sistem. Struktur tabel analisa hasil dapat dilihat pada tabel berikut.

Tabel 4. 7 Tabel *database analisa hasil*

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
Id	<i>Integer</i>	4	<i>Primary Key</i>
nama	<i>Varchar</i>	60	-
kelamin	<i>Enum</i>	'P','W'	-
alamat	<i>Varchar</i>	100	-
pekerjaan	<i>Varchar</i>	60	-
kd_solusi	<i>Char</i>	4	-
noip	<i>Varchar</i>	60	-
tanggal	<i>datetime</i>	<i>default</i>	-

## c. Tabel Data Artikel

Tabel data artikel digunakan untuk menyimpan data dari beberapa artikel kesehatan yang dibuat dan ditampilkan kedalam sistem. Berikut struktur *database* yang digunakan dalam tabel data artikel.

Tabel 4. 8 Tabel *database data artikel*

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
Id	<i>Integer</i>	15	<i>Primary Key</i>
Idadmin	<i>Varchar</i>	55	<i>Foreign Key</i>

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
tgl	<i>Varchar</i>	55	-
judul	<i>Varchar</i>	100	-
isi	<i>Text</i>	<i>default</i>	-
foto	<i>Varchar</i>	100	-
ket	<i>Enum</i>	'Y', 'T'	-
keyword	<i>Varchar</i>	100	-
deskripsi	<i>Text</i>	<i>default</i>	-

d. Tabel Buku Tamu

Tabel data buku tamu digunakan untuk menyimpan data responden dari *user* yang menggunakan Sistem Pakar ini. Struktur *database* yang digunakan dapat dilihat pada tabel berikut.

**Tabel 4. 9 Tabel *database* buku tamu**

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
Id	<i>Integer</i>	15	<i>Primary Key</i>
nama	<i>Varchar</i>	40	-
email	<i>Varchar</i>	55	-
isi	<i>Text</i>	<i>default</i>	-

e. Tabel Data Gejala

Tabel *database* gejala digunakan untuk menyimpan data gejala penyakit. Berikut struktur *database* yang digunakan dalam sistem.

**Tabel 4. 10 Tabel *database* gejala**

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
kd_gejala	<i>Char</i>	4	<i>Primary Key</i>
nm_gejala	<i>Varchar</i>	100	-

f. Tabel Data *Rules*

Tabel data *rules* digunakan untuk menyimpan data aturan atau *rules* yang ditentukan dan digunakan sebagai basis pengetahuan Sistem Pakar

ini. Struktur *database* dapat dilihat pada tabel berikut.

**Tabel 4. 11** Tabel *database rules*

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
kd_solusi	<i>Char</i>	4	-
kd_gejala	<i>Char</i>	4	-

g. Tabel Data Solusi

Tabel *database* solusi digunakan untuk menyimpan data solusi dari suatu penyakit berdasarkan gejala – gejala yang telah diberikan.

Struktur *database* data solusi dapat dilihat pada tabel berikut.

**Tabel 4. 12** Tabel *database* solusi

<b>Nama Field</b>	<b>Type</b>	<b>Length</b>	<b>Keterangan</b>
kd_solusi	<i>Char</i>	4	<i>Primary Key</i>
nm_solusi	<i>Varchar</i>	300	-
solusi	<i>Text</i>	<i>default</i>	-
definisi	<i>Text</i>	<i>default</i>	-

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. Kesimpulan**

Berdasarkan analisa dan perancangan sistem, implementasi dan pengujian sistem, maka dapat disimpulkan bahwa :

1. Dari Sistem Pakar yang telah dibuat telah dihasilkan suatu sistem yang dapat menghasilkan proses diagnosis penyakit *stunting* dan kurang gizi lainnya pada manusia dengan menggunakan metode *Forward Chaining*.
2. Sistem Pakar berbasis *website* yang dibuat mampu membantu pengguna untuk mendiagnosis penyakit dan memperoleh informasi mengenai penyakit *stunting* dan kurang gizi lainnya, karena sistem ini memiliki daftar bacaan atau artikel didalam fiturnya.
3. Sistem ini sudah dilakukan proses pengujian dengan menggunakan metode *user acceptance testing* yang dimana hasil pengujian ini mendapatkan hasil bahwa sistem berjalan dengan baik.
4. Sistem Pakar ini memiliki tindakan preventif atau mengetahui gejala awal yang terjadi yang bisa membantu masyarakat untuk mengetahui gejala suatu penyakit secara *online*.

#### **B. Saran**

Dari kesimpulan yang ada, maka dapat dikemukakan saran – saran yang akan sangat membantu untuk pengembangan perangkat lunak ini selanjutnya :

1. Program ini masih jauh dari sempurna, untuk itu perlu dilakukan perbaikan – perbaikan dan pengembangan sistem demi kesempurnaan program dan kemudahan *user* dalam menggunakan sistem ini.

2. Perlu diadakan penambahan data untuk jenis penyakit kurang gizi lainnya beserta gejala – gejala klinisnya sehingga informasi yang dimiliki akan semakin luas dan banyak.

## DAFTAR PUSTAKA

- Aridiyah, F. O., Rohmawati, N., & Ririanty, M. (2015). *Faktor-faktor yang Mempengaruhi Kejadian Stunting pada Anak Balita di Wilayah Pedesaan dan Perkotaan (The Factors Affecting Stunting on Toddlers in Rural and Urban Areas)*. 3(1).
- Erri Wahyu Puspitarini, D. W. P. A. P. N. (2016). *Game Edukasi Berbasis Android Sebagai Media Pembelajaran Untuk Anak Usia Dini*. J I M P - Jurnal Informatika Merdeka Pasuruan, 1(1), 46–58.
- Hamid, E. S., & Susilo, Y. S. (2015). *Strategi Pengembangan Usaha Mikro Kecil Dan Menengah Di Provinsi Daerah Istimewa Yogyakarta*. Jurnal Ekonomi Pembangunan: Kajian Masalah Ekonomi Dan Pembangunan, 12(1), 45.
- Haryono, A. T., Rahmawati, D., & Fitriastuti, T. (2016). *Aplikasi Program PHP & MySQL*.
- Haviluddin. (2011). *Memahami Penggunaan UML (Unified Modelling Language)*, 6(1), 1–15.
- Hayadi, B. H. (2018). *Sistem Pakar* (1st ed.).
- Heriyanto, Y. (2018). *Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT.APM Rent Car*. Jurnal Intra-Tech, 2(2), 64–77.
- Informasi, S., Belitung, B., Informasi, P. S., Penerimaan, P., Baru, S., & Web, C. P. (2018). *Sistem Pendaftaran Siswa Baru Pada SMP N 1 Kelapa Berbasis Web*. 07(September), 110–115.
- Islah Wahyuni. (2020). *Analisis Faktor Masalah Pertumbuhan (Status Gizi, Stunting) Pada Anak Usia < 5 Tahun Di Wilayah Kerja Puskesmas Sidomulyo Kota Pekanbaru*. Jurnal Kebidanan Mutiara Mahakam, 8(1), 51–70.
- Kusumawati, E., Rahardjo, S., & Sari, H. P. (2015). *Model of Stunting Risk Factor Control among Children under Three Years Old*. Kesmas: National Public Health Journal, 9(3), 249.
- Listiyono, H. (2008). *Merancang dan Membuat Sistem Pakar*. Jurnal Teknologi Informasi DINAMIK, XIII(2), 115–124.
- Lubis, A. (2016). *Basis Data Dasar* (1st ed.). Deepublish.

- Mulyani, S. (2017). *Analisis dan Perancangan Sistem Informasi Manajemen Keuangan Daerah: Notasi Pemodelan Unified Modeling Language (UML)* (1st ed.). Abdi Sistematika.
- Munti, N. Y. S., & Effindri, F. A. (2017). Perancangan Aplikasi Sistem Pakar Diagnosa Penyakit Ginekologi Menggunakan Metode Forward Chaining Berbasis Web Mobile. *Jurnal Media Infotama*, 13(2), 67–72. <https://doi.org/10.37676/jmi.v13i2.454>
- Mustakim. (2018). *Pemrograman Internet* (J. Kasdi (ed.)). Cahaya Firdaus.
- Nasikhah, R. (2012). *Faktor Risiko Kejadian Stunting Pada Balita*. 1–27.
- Nugrahani, F. (2014). *Metode Penelitian Kualitatif*. In *Journal of Physics A: Mathematical and Theoretical* (Vol. 44, Issue 8).
- Pamungkas, C. A. (2017). *Pengantar dan Implementasi Basis Data* (1st ed.). Deepublish.
- Prasetyadi, A. E. (2011). *Web 3.0: Teknologi Web Masa Depan*. 1(3), 1–6.
- Priyanti, D. (2013). *Sistem Informasi Data Penduduk Pada Desa Bogoharjo Kecamatan Ngadirojo Kabupaten Pacitan*. *IJNS - Indonesian Journal on Networking and Security*, 2(4), 56. [ijns.org](http://ijns.org)
- Raharjo, M. (2011). *Metode Pengumpulan Data Penelitian Kualitatif*. *Animal Genetics*, 39(5), 561.
- Ramadhan, P. S., & Pane, U. F. S. (2018). *Mengenal Metode Sistem Pakar* (Fungky (ed.); 1st ed.). Uwais Inspirasi Indonesia.
- Rosnelly, R. (2012). *Sistem Pakar Konsep dan Teori* (P. Y. Jati (ed.); 1st ed.). Penerbit ANDI.
- Setiawan, E., Machmud, R., & Masrul, M. (2018). *Faktor-Faktor yang Berhubungan dengan Kejadian Stunting pada Anak Usia 24-59 Bulan di Wilayah Kerja Puskesmas Andalas Kecamatan Padang Timur Kota Padang Tahun 2018*. *Jurnal Kesehatan Andalas*, 7(2), 275.
- Solichin, A. (2016). *Pemrograman Web dengan PHP dan MySQL*. Budi Luhur.
- Sutejo, S. (2016). *Pemodelan UML Sistem Informasi Geografis Pasar Tradisional Kota Pekanbaru*. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 7(2), 89–99.

- Trihono, Atmarita, Tjandrarini, D. H., Irawati, A., Utami, N. H., Tejayanti, T., & Nurlinawati, I. (2015). *Pendek (Stunting) di Indonesia, Masalah dan Solusinya*. In Lembaga Penerbit Balitbangkes.
- Wicaksono, S. R. (2011). *Rekayasa Perangkat Lunak*. Seribu Bintang.
- Widyawati, E. (2016). *Rancang Bangun Aplikasi Kependudukan Berbasis Web Di Desa Kedungrejo Waru-Sidoarjo*. *Jurnal Manajemen Informatika*, 6(1).