**PROPOSAL PENELITIAN**

**PERANCANGAN SISTEM PAKAR DALAM MENDIAGNOSA PENYAKIT PNEUMONIA PADA ANAK DENGAN MENGGUNAKAN METODE *CASE BASED REASONING* BERBASIS WEB**

****

**OLEH :**

**RINI SOVIA (Ketua)**

**EKA PRAJA WIYATA MANDALA (Anggota 1)**

**RAZKY JOSEFA (Anggota 2)**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PUTRA INDONESIA “YPTK”**

**PADANG**

**2019**

**RINGKASAN**

*Pneumonia* merupakan infeksi saluran pernapasan akut yang menjadi penyebab kematian utama pada balita di dunia, terutama di negara berkembang. *Pneumonia* yang terjadi pada balita akan memberikan gambaran klinik yang lebih buruk daripada orang dewasa karena pada balita sistem pertahanan tubuh yang dimiliki relatif rendah. Di saat sekarang ini masyarakat tidak terlalu perduli terhadap penyakit *pneumonia* ini di karenakan penyakit *pneumonia* ini memiliki gejala yang hampir sama dengan penyakit batuk, hal ini terjadi karena lemahnya pengetahuan masyarakat terhadap penyakit *pneumonia*. Informasi yang cepat dan tepat dari seorang pakar kesehatan sangat di butuhkan oleh orang tua tentang gejala penyakit *pneumonia* dan cara penanganannya yang lebih dini. Oleh karena itu dirancang sebuah sistem pakar untuk mengetahui klasifikasi penyakit *pneumonia* pada anak. Metode yang digunakan dalam sistem pakar adalah *Case Based Reasoning*. Sistem pakar ini dapat memberikan solusi serta cara pencegahan dini penyakit *pneumonia* pada anak dengan proses pendiagnosaan dilakukan secara efisien dan menghemat waktu dalam membantu masyarakat atau orang tua dalam pencegahan yang terjadi. Aplikasi sistem pakar berbasis web ini dibangun dengan bahasa pemograman PHP dan penyimpanan data MySQL.

**Kata kunci:** *Case Based Reasoning*, *Pneumonia*, Sistem Pakar, Web

**KATA PENGANTAR**

**S**egala puji kita ucapkan kehadirat Tuhan Yang Maha Esa yang telah memberikan Rahmat dan Hidayah-Nya sehingga penulis dapat menyusun dan menyelesaikan laporan.Hasil penelitian ini diharapkan dapat membantu pihak-pihak yang membutuhkan terutama untuk pengguna sehingga dapat meningkatkan efisien dan efektifitas kerja dalam menjalankan pekerjaan dan menyajikan informasi. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyelesaian skripsi ini, terutama kepada :

1. Bapak H. Herman Nawas, selaku Ketua Yayasan Perguruan Tinggi Komputer YPTK Padang.
2. Bapak Prof. Dr. Sarjon Defit, S.Kom, M.Sc, selaku Rektor Universitas Putra Indonesia YPTK Padang.
3. Bapak Dr. Julius Santony, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Putra Indonesia YPTK Padang.
4. Ibuk dr. Laura Zeffira, Sp.A, M.Biomed yang telah membantu penulis dalam memberikan informasi-informasi tentang penyakit pneumonia pada anak yang dibutuhkan untuk melakukan penelitian ini.

Pada penyelesaian Penelitian ini penulis sangat menyadari bahwa hasil dari Skripsi ini sangatlah jauh dari kesempurnaan. Namun tetap diharapkan hasil dari penelitian ini dapat memberikan manfaat, baik bagi penulis maupun pembaca dikemudian hari.

Padang, Januari 2019

Penulis

**DAFTAR ISI**

**HALAMAN JUDUL i**

**RINGKASAN ii**

**KATA PENGANTAR iii**

**DAFTAR ISI iv**

**BAB I PENDAHULUAN 1**

1.1 Latar Belakang 1

1.2 Perumusan Masalah 2

1.3 Hipotesa 3

1.4 Batasan Masalah 3

1.5 Tujuan Penelitian 3

1.6 Manfaat Penelitian 4

1.7 Gambaran Umum Pakar 4

**BAB II LANDASAN TEORI 6**

2.1 Rekayasa Perangkat Lunak 6

2.1.1 Perangkat Lunak 6

2.1.2 Rekayasa Perangkat Lunak 8

2.1.3 Proses Rekayasa Perangkat Luna 9

2.1.4 Model-model Rekayasa Perangkat Lunak 10

2.1.4.1 Model *Waterfall* 10

2.1.4.2 Model Prototipe 12

2.1.4.3 Model Spiral 13

2.2 *Unified Modeling Language* (UML) 14

2.2.1 Pengertian UML 14

2.2.2 Sejarah UML 15

2.2.3 Diagram UML 15

2.2.3.1 *Use Case Diagram* 16

2.2.3.2 *Class Diagram* 19

2.2.3.3 *Sequence Diagram* 21

2.2.3.4 *Communication Diagram* 23

2.2.3.5 *State Chart Diagram* 24

2.2.3.6 *Activity Diagram* 26

2.2.3.7 *Deployment Diagram* 27

2.3 Kecerdasan Buatan (*Artificial Inteligence****)*** 28

2.3.1 Definisi Kecerdasan Buatan 28

2.3.2 Kecerdasan Alami dan Kecerdasan Buatan 31

2.3.3 Ruang Lingkup Kecerdasan Buatan 34

2.4 Sistem Pakar 35

2.4.1 Definisi Sistem Pakar 36

2.4.2 Kelebihan Sistem Pakar 38

2.4.3 Kekurangan Sistem Pakar 39

2.4.4 Manfaat Sistem Pakar 39

2.4.5 Ciri-ciri Sistem Pakar 40

2.4.6 Konsep Dasar Sistem Pakar 40

2.5 Metode *Case Based Reasoning* 43

2.5.1 Definisi *Case Based Reasoning* 44

2.5.2 Tahap *Case Based* *Reasoning* 44

2.6 Sekilas Tentang *Web* 46

2.7 Pengenalan Aplikasi *Server* 47

2.7.1 *Web Server* 48

2.7.2 *Database Server* 48

2.7.3 *Hypertext Preprocessor* (PHP) 48

2.8 *Pneumonia*  49

**BAB III METODOLOGI PENELITIAN 52**

3.1 Kerangka Penelitian 52

3.2 Tahapan Penelitian 52

3.2.1 Penelitian Pendahuluan 53

3.2.2 Pengumpulan Data 53

3.2.3 Analisa 56

3.2.4 Perancangan 57

3.2.5 Implementasi Sistem 59

3.2.6 Pengujian 59

**BAB IV ANALISA DAN PERANCANGAN 61**

4.1 Analisa 61

4.1.1 Analisa Data 61

4.1.2 Analisa Proses 62

4.1.3 Analisa Sistem 68

4.2 Perancangan 69

4.2.1 Perancangan Model 69

4.2.1.1 *Use Case Diagram* 70

4.2.1.2 *Class Diagram* 91

4.2.1.3 *Sequence Diagram* 96

4.2.1.4 *Collaboration Diagram* 105

4.2.1.5 *State Chart Diagram* 115

4.2.1.6 *Activity Diagram* 120

4.2.1.7 *Deployment Diagram* 123

4.2.2 Perancangan *Interface* 123

**BAB V IMPLEMENTASI DAN PENGUJIAN 132**

5.1 Implementasi Sistem 132

5.1.1 Analisa Kebutuhan *Hardware* 132

5.1.2 Analisa Kebutuhan *Software* 132

5.2 Pengujian 134

5.2.1 Pengujian *Local* 134

5.2.2 Pengujian *Online* 137

5.2.3 Pengujian Aplikasi 143

5.2.4 Pengujian *Interface* 145

**BAB VI PENUTUP 154**

6.1 Kesimpulan 154

6.2 Keterbatasan Sistem 154

6.3 Saran 155

**DAFTAR PUSTAKA**

**LAMPIRAN**

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

*Pneumonia* merupakan infeksi saluran pernapasan akut yang menjadi penyebab kematian utama pada balita di dunia, terutama di negara berkembang. Pneumonia yang terjadi pada balita akan memberikan gambaran klinik yang lebih buruk daripada orang dewasa karena pada balita sistem pertahanan tubuh yang dimiliki relatif rendah. Bayi dan anak kecil lebih rentan terhadap penyakit ini karena respon imunitas mereka masih belum berkembang dengan baik.

Pada penelitian sebelumnya yang dilakukan oleh Diki Indra Lesmana pada tahun 2017 dengan judul “Sistem Pakar Mendiagnosa Penyakit Pneumonia dengan Penelusuran Forward Chaining Menggunakan Metode Certainty Factor” menjelaskan bahwa *pneumonia* adalah penyakit infeksi yang menyerang paru yang disebabkan oleh virus, bakteri atau jamur. Kondisi kesehatan ini sering kali disebut dengan paru-paru basah. Penyakit ini merupakan penyakit yang serius dan dapat mengancam nyawa manusia, *pneumonia* paling serius jika terjadi pada bayi dan anak-anak. Dalam sistem pakar terdapat suatu metode untuk menyelesaikan suatu masalah ketidak pastian data, salah satu metode yang dapat digunakan adalah faktor kepastian. *Certainty factor* (CF) merupakan nilai parameter klinis yang diberikan untuk menunjukan besarnya kepercayaan. Hasil penelitian perhitungan *certainty factor* yang dilakukan penyakit *pneumonia* ringan memiliki tingkat keyakinan sistem 20%.

Di saat sekarang ini masyarakat tidak terlalu perduli terhadap penyakit *pneumonia* ini di karenakan penyakit *pneumonia* ini memiliki gejala yang hampir sama dengan penyakit batuk, hal ini terjadi karena lemahnya pengetahuan masyarakat terhadap penyakit *pneumonia*. Informasi yang cepat dan tepat dari seorang pakar kesehatan sangat di butuhkan oleh orangtua tentang gejala-gejala penyakit *pneumonia* dan cara penanganannya yang lebih dini. Diagnosa awal sangat penting untuk menghindari ancaman penyakit *pneumonia*. Hal inilah yang mendorong perancangan sebuah sistem pakar untuk mendiagnosa penyakit *pneumonia* pada anak agar orangtua dapat mengetahui dan mendapat informasi dengan cepat gejala yg di rasakan yang berhubungan dengan penyakit *pneumonia*. Salah satu kemajuan teknologi informasi adalah sistem berbasis cerdas yang disebut sistem pakar. Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Sistem pakar berguna untuk menghasilkan suatu diagnosa seorang pakar untuk memecahkan suatu permasalah yang di hadapi.

Berdasarkan permasalahan tersebut maka penulis melakukan akan sebuah penelitian dengan judul **“PERANCANGAN SISTEM PAKAR DALAM MENDIAGNOSA PENYAKIT PNEUMONIA PADA ANAK MENGGUNAKAN METODE CASE BASED REASONING BERBASIS WEB”**

* 1. **Perumusan Masalah**

Dari uraian latar belakang masalah di atas dapat dirumuskan masalah yang dihadapi, yaitu :

1. Bagaimana membantu memberikan pengetahuan kepada orangtua informasi tentang penyakit *pneumonia* pada anak?
2. Bagaimana membantu memberikan pengetahuan kepada orangtua tentang gejala penyakit *pneumonia* pada anak?
3. Bagaimana membantu orangtua dengan cepat dan tepat untuk mendiagnosa awal dan melakukan penanganan dini pada anak yang terkena penyakit *pneumonia*?
   1. **Hipotesa**

Dari rumusan masalah diatas, dapat ditarik beberapa hipotesa :

1. Dengan adanya sistem pakar diharapkan dapat membantu orangtua untuk mengetahui informasi penyakit *pneumonia* pada anak.
2. Dengan adanya sistem pakar diharapkan dapat membantu orangtua mengetahui gejala penyakit *pneumonia* pada anak.
3. Dengan adanya sistem pakar diharapkan dapat mempermudah pengguna dalam mendiagnosa awal dan penanganan dini terhadap penyakit *pneumonia* pada anak.
   1. **Batasan Masalah**

Agar penelitian ini terarah dan tujuan dari penulis ini tercapai sesuai diharapkan, maka perlu adanya pembatasan masalah tidak menyimpang dari dari pokok permasalahan penelitian ini akan merancang sistem pakar mendiagnosa penyakit *pneumonia* pada anak dengan metode *case based reasoning* berbasis web.

* 1. **Tujuan Penelitian**

Adapun tujuan dari penelitian ini sebagai berikut :

1. Membantu orangtua dalam memahami dan mengetahui penyakit *pneumonia* pada anak.
2. Memberikan pengetahuan kepada orangtua tentang gejala penyakit *pneumonia* pada anak.
3. Membantu orangtua dalam mendiagnosa awal dan penanganan dini terhadap penyakit *pneumonia* pada anak.
   1. **Manfaat Penelitian**

Adapun manfaat yang diharapkan dalam penelitian ini adalah sebagai berikut :

1. Pengetahuan orangtua menjadi lebih paham tentang penyakit *pneumonia* pada anak.
2. Dengan sistem pakar ini membantu orangtua mengetahui gejala penyakit *pneumonia* pada anak.
3. Dengan sistem pakar ini orangtua bisa mengetahui lebih cepat dan tidak terlambat dalam penanganan penyakit *pneumonia* pada anak.
   1. **Gambaran Umum Pakar**

Pada penelitian ini dilakukan konsultasi dengan seorang pakar spesialis anak yang bernama dr. Laura Zeffira, Sp.A, M.Biomed. Beliau lahir di Dumai, 21 agustus 1985. Pendidikan terakhir beliau dengan program studi Biomedik pada tahun 2016. Sekarang beliau bekerja di Fakultas Kedokteran Baiturrahmah sebagai kepala prodi profesi dan RSI Siti Rahmah sebagai dokter spesialis anak.

Dokter Laura Zeffira biasa di panggil dokter laura, beliau melanjutkan sekolah ke jenjang perkuliahan di Fakultas Kedokteran Universitas Andalas pada tahun 2007. Beliau mengambil jenjang profesi dokter pada tahun 2009. Pada tahun 2016 beliau mengambil gelar spesialis anak dan menyelesaikan perkuliahan pascasarjana. Sebagai kepala prodi profesi dan dokter spesialis anak, beliau juga membuka praktek di Klinik My Lovely Child yang berada di Jalan Perintis Kemerdekaan No 19 Kec. Padang Timur.

Dokter Laura Zeffira sampai sekarang masih aktif menangani kasus-kasus penyakit *pneumonia* pada anak ini, keluhan-keluhan yang dilaporkan oleh orangtua banyak diselesaikan oleh dokter Laura Zeffira. Semasa pendidikan profesi dan spesialis anak selama 7 tahun, dokter Laura Zeffira juga telah banyak menyelesaikan masalah penyakit *pneumonia* pada anak.

**BAB II**

**LANDASAN TEORI**

1. **Rekayasa Perangkat Lunak**
   * 1. **Perangkat Lunak**

Perangkat lunak adalah instruksi langsung komputer untuk melakukan pekerjaan dan dapat ditemukan di setiap aspek kehidupan modern dari aplikasi yang kritis untuk hidup (*life-critical*), seperti perangkat pemantauan medis dan pembangkit tenaga listrik sampai perangkat hiburan, seperti video game (Simarmata, Janner, 2009, hal : 1).

Karakter perangkat lunak (S, Rosa A dan M. Shalahuddin, 2014, hal : 2):

* 1. Perangkat lunak dibangun dengan rekayasa (*software engineering*) bukan diproduksi secara manufaktur atau pabrikan.
  2. Perangkat lunak tidak pernah usang (“*wear out*”) karena kecacatan perangkat lunak dapat diperbaiki.
  3. Barang produksi pabrikan biasanya komponen barunya akan terus diproduksi, sedangkan perangkat lunak biasanya terus diperbaiki seiring bertambahnya kebutuhan.

Terdapat 8 kategori aplikasi pada perangkat lunak komputer yaitu(S, Rosa A dan M. Shalahuddin, 2014, hal : 2):

1. Perangkat lunak sistem (*system software*)

Merupakan kumpulan program yang dalam hal ini program yang satu ditulis untuk memenuhi kebutuhan program lainnya.

1. Perangkat lunak waktu nyata (*real-time software*)

Merupakan perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata (*real-time*). Reaksi yang dibutuhkan pada perangkat lunak harus langsung menghasilkan respon yang di inginkan.

1. Perangkat lunak bisnis (*business software*)

Merupakan pengelola informasi bisnis (seperti akuntasi, penjualan, pembayaran, penyimpanan (*inventory*)).

1. Perangkat lunak untuk keperluan rekayasa dan ilmuan (*engineering and scientific software*)

Merupakan perangkat lunak yang mengimplementasikan algoritma yang terkait dengan keilmuan ataupun perangkat lunak yang membantu keilmuan, misalkan perangkat lunak di bidang astronomi, di bidang matematika, dan lain sebagainya.

1. Perangkat lunak tambahan untuk membantu mengerjakan suatu fungsi dari perangkat lunak yang lainnya (*embedded software*)

Misalnya perangkat lunak untuk mencetak dokumen di tambahkan agar perangkat lunak yang memerlukan dapat mencetak laporan, maka perangkat lunak untuk mencetak dokumen ini disebut *embedded software*.

1. Perangkat lunak komputer personal (*personal computer software*)

Merupakan perangkat lunak untuk PC misalnya perangkat lunak pemrosesan teks, pemrosesan grafik dan lain sebagainya.

1. Perangkat lunak berbasis web (*web based computer*)

Merupakan perangkat perangkat lunak yang dapat di akses dengan menggunakan browser.

1. Perangkat lunak berintelijensia buatan (*artificial intelligence*)

Merupakan perangkat lunak yang menggunakan algoritma tertentu untuk mengelola data sehingga seakan-akan memiliki intelijensia seiring bertambahnya data yang diproses.

* + 1. **Rekayasa Perangkat Lunak**

Rekayasa perangkat lunak adalah disiplin rekayasa dengan perangkat lunak yang dikembangkan. Biasanya proses melibatkan penemuan pada keinginan klien, menyusun di dalam daftar kebutuhan, merancang arsitektur yang mampu mendukung semua kebutuhan, perancangan, pengodean, pengujian dan pengintegrasian bagian yang terpisah, menguji keseluruhan, penyebaran dan pemeliharaan perangkat lunak (Simarmata, Janner, 2010, hal : 11).

Rekayasa perangkat lunak merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin (S, Rosa A, M. Shalahuddin, 2013, hal : 4).

Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan (*customer*). Adapun ilmu komputer lebih fokus pada teori dan konsep dasar perangkat komputer. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut :

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintainability*)
2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*)
3. Efisien dari segi sumber daya dan penggunaan
4. Kemampuan untuk dipakai sesuai dengan kebutuhan (*usability*)

Dari kriteria diatas maka perangkat lunak yang baik adalah perangkat lunak yang dapat memenuhi kebutuhan pelanggan (*customer*) atau *user* (pemakai perangkat lunak) atau berorientasi pada pelanggan atau pemakai perangkat lunak, bukan berorientasi pada pembuat atau pengembang perangkat lunak.

* + 1. **Proses Rekayasa Perangkat Lunak**

Proses rekayasa perangkat lunak dilakukan selama pembangunan perangkat lunak. Proses-proses yang dilakukan dalam rekayasa perangkat lunak secara garis besar adalah sebagai berikut :

Mulai

**Sumber: S, Rosa A dan M. Shalahudin, hal : 8 , 2013**

**Gambar 2.1 Tahapan umum rekayasa perangkat lunak**

Proses-proses pada gambar di atas dapat dilakukan berulang kali sampai perangkat lunak memenuhi kebutuhan pelanggan atau *user*. Untuk membangun perangkat lunak yang benar-benar baik maka diperlukan tahapan-tahapan rekayasa perangkat lunak (S, Rosa A dan M. Shalahudin, hal : 9 , 2013).

Proses perangkat lunak (*software process*) adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari :

1. Pengumpulan Spesifikasi (*spesification*)

Mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.

1. Pengembangan (*Development*)

Pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak.

1. Validasi (*Validation*)

Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan (*customer*).

1. Evolusi (*Evolution*)

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*customer*).

* + 1. **Model-model Rekayasa Perangkat Lunak**

Rekayasa perangkat memiliki beberapa model dalam penerapan tahapan prosesnya, yaitu : (S, Rosa A, dan M. Shalahuddin, hal : 28, 2013) :

* + - 1. **Model *Waterfall***

Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun :

Analisis

Desain

Pengodean

Pengujian

**Sumber : S, Rosa A dan M. Shalahuddin, hal : 29, 2013**

**Gambar 2.2 Tahapan *Classic Life Cycle***

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhkan dilakukan secara intensif untuk menspesifikasi kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan user. Spesifikasi perangkat lunak pada tahap ini perlu untuk didokumentasikan.

1. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasikan kebutuhan desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

1. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

1. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji.

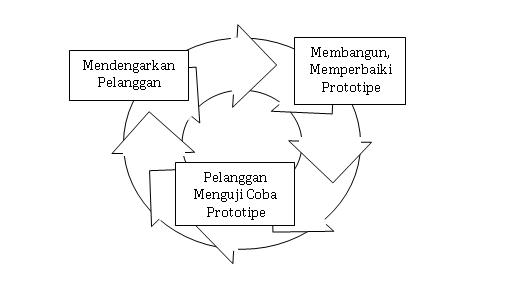
1. Pendukung (*support*)atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi untuk membuat perangkat lunak baru.

* + - 1. **Model Prototipe**

Model Prototipe dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknik dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.

Model prototype (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*. Berikut adalah gambar dari model prototipe.



**Sumber: S, Rosa A dan M. Shalahuddin, 2013, hal : 32**

**Gambar 2.3 Ilustrasi Model *Prototipe***

* + - 1. **Model Spiral**

Merupakan model proses perangkat lunak yang memadukan wujud pengulangan dari model prototyping dengan aspek pengendalian dan sistematika dari linear sequential model, dengan penambahan elemen baru yaitu analisis resiko. Model spiral dibagi menjadi beberapa kerangka aktifitas atau disebut juga wilayah kerja (*task region*). Banyaknya wilayah kerja biasanya diantara tiga sampai enam wilayah sebagai berikut :

1. Komunikasi dengan pelanggan (*customer communication*).

Aktifitas ini diperlukan untuk membengun komunikasi yang efektif antara pengembang (*developer*) dan pelanggan (*customer*).

1. Perencanaan (*planning*)

Aktifitas ini diperlukan untuk mendefenisikan sumber daya, waktu, dan informasi yang terkait dengan proyek.

1. Analisis risiko (*risk analysis*)

Aktifitas ini diperlukan untuk memperkirakan risiko dari segi teknis maupun manajemen.

1. Rekayasa (*engineering*)

Aktifitas ini diperlukan untuk membangun satu atau lebih representasi dari aplikasi perangkat lunak (dapat juga berupa prototipe).

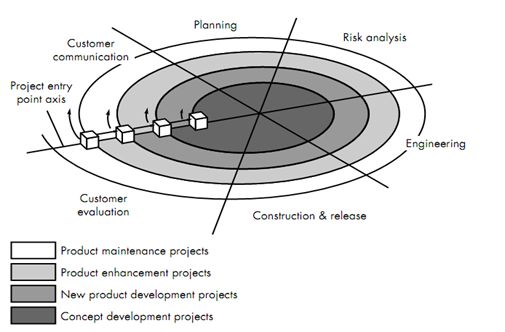
1. Konstruksi dan peluncuran (*contruction and release*)

Aktifias ini dibutuhkan untuk mengonstruksi, menguji, melakukan instalasi, dan menyediakan dukungan terhadap *user.*

1. Evaluasi pelanggan (*customer evaluation*)

Aktifitas ini dibutuhkan untuk mendapatkan umpan balik berdasarkan evaluasi representasi perangkat lunak yang dihasilkan dari proses rekayasa dan diimplementasikan pada tahap instansi. (S, Rosa A, M. Shalahuddin, hal : 39, 2014)

Berikut adalah gambar model spiral :



**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal : 40**

**Gambar 2.4 Ilustrasi Model *Spiral***

1. ***Unified Modelling Language* (UML)**
   * 1. **Pengertian UML**

*Unified Modeling Language* (UML) adalah salah satu standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan mengunakan diagram dan teks-teks pendukung (S, Rosa A. dan M. Shalahuddin, 2013, hal : 137).

* + 1. **Sejarah UML**

Bahasa pemrograman berorientasi objek pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Kemunculannya telah memberikan sumbangan yang besar pada developer pengembangan bahasa pemrograman berorientasi objek selanjutnya. Perkembangan aktif dari pemrograman berorientasi objek mulai ketika berkembangnya bahasa pemrograman Smalltalk yang kemudian diikuti dengan perkembangan bahasa pemrograman lainnya. Metode yang pertama diperkenalkan oleh Sally Shlaer dan Stephen Mellor (Shlaer-Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Gady Booch (Booch, 1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha-Premerlani-Eddy-Lorensen, 1991), dan masih banyak lagi.

Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram* *Interchange* *Specification*, UML Infrastructure, UML Superstructure, dan Object Constraint Language (OCL). Seluruh spesifikasi tersebut dapat diakses di website http://www.omg.org (S, Rosa A, dan M. Shalahuddin, 2013, hal : 138).

* + 1. **Diagram UML**

UML terdapat 7 macam diagram, diantaranya (S, Rosa A, M. Shalahuddin, 2013, hal : 140) :

* + - 1. ***Use Case Diagram***

*Use Case* atau diagram *use Case* merupakan pemodelan untuk kelakukan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sitem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Adadua hal utama pada *use case* yaitu pendefinisian yang di sebut aktor dan *use case*.

1. Aktor merupakan orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case:*

# Tabel 2.1 Tabel Simbol *Use Case Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| *Use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling betukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case* |
| Aktor / *actor*    nama aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal fase nama aktor. |
| Asosiasi / *association* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor |
| Ekstensi / *extend*  <<extand>> | Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misal:  <<extend>>  <<extend>>  Arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend-*nyamerupakan jenis yang sama dengan *use case* yang menjadi induknya. |
| Generalisasi / *generalization* | Hubungan generalisasi dan spesialisasi (umun– khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  arah panah mengarah pada *use case* yang menjadi generalisasinya (umum) |
| Menggunakan / *include / uses*    *<<*include*>>*  <<uses>> | Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat di jalankan *use case* ini.  Ada dua sudut pandang yang cukup besar mengenai include *use case:*   1. *Include* berarti *use case*yang di tambahkan akan selalu di panggil oleh *use case* tambahan dijalankan, misal pada kasus berikut:   <<include>>   1. *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *usse case* tambahan dijalankan, misal pada kasus berikut:   <<include>>  Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan |

**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal : 156**

* + - 1. ***Class Diagram***

Diagram kelas atau Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system. Kelas memiliki atribut (variabel-variabel yang dimiliki suatu kelas) dan operasi atau metode (fungsi yang dimiliki suatu kelas).

Kelas yang baik pada diagram kelas memiliki jenis kelas-kelas berikut:

* 1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

* 1. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan kepemakai.

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

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

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

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data (S, Rosa A dan M. Shalahudin, 2013).

Berikut adalah simbol-simbol yang ada pada diagram kelas :

**Tabel 2.2 Tabel Simbol *Class Diagram***

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| |  | | --- | | nama\_kelas | | +atribut | | +operasi() |   Kelas | Kelas pada struktur sistem |
| Antarmuka / *interface*  **nama\_interface** | Sama dengan konsep interface dalam pemograman berorientasi objek |
| Asosiasi / association | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
| Asosiasi berarah / directed association | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity* |
| Generalisasi | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus) |
| kebergantungan / dependency | Relasi antarkelas dengan makna kebergantungan antarkelas |
| Agregasi / aggregation | Relasi antarkelas dengan makna semua-bagian (whole-part) |

**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal : 146**

* + - 1. ***Sequence Diagram***

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objekdan message yang dikirimkan dan dirima antarobjek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skanario yang pada *use case*. (S, Rosa A dan M. Shalahudin, Rekayasa Perangkat Lunak, 2013).

Berikut adalah simbol-simbol yang ada pada diagram sequence :

**Tabel 2.3 Tabel Simbol *Sequence Diagram***

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Aktor  nama aktor  atau  nama aktor  tanpa waktu aktif | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang,tapi aktor belum tentu merupakan orang;biasanya dinyatakan menggunakan kata benda diawal frase nama aktor |
| Garis hidup / lifeline | Menyatakan kehidupan suatu objek |
| Objek  nama objek:nama kelas | Menyatakan objek yang berinteraksi pesan |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya misalnaya  1: login()  2: cekStatusLogin()  3: open()  Maka cek status login() dan open() dilakukan di dalam metode login ()  Aktor tidak memiliki waktu aktif |
| Pesan tipe *create*  <<create>> | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| Pesan tipe *call*  1 : nama\_metode() | Menyataka suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri,  1 : nama\_metode()  Arah panah mengarah pada objek yang memiliki operasi / metode, karna ini memanggil operasi / metode maka operasi / metode yang di panggil harus harus ada pada diagram kelas sesuai dengan kelas pbjek yang berorioentasi. |
| Pesan tipe send  1: masukan | Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi objek lainnya, arah panah mengarah pada objek yang di kirim |
| Pesan tipe return  1: keluaran | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| Pesan tipe destroy  <<destroy>> | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah menagarah pada objek yang diakhiri, sebaiknya jika ada kreate maka ada destroy |

**Sumber: S, Rosa A dan M. Shalahudin, 2013 , hal : 166**

* + - 1. ***Communication Diagram***

*Communication diagram* atau diagram komukasi pada UML, versi 2x adalah penyederhanaan dari diagram kolaborasi (*Collaboration diagram*) pada UML versi 1.x. *Collaboration diagram* sudah tidak muncul lagi pada UML versi 2.x.

Berikut simbol-simbol yang terdapat pada diagram kolaborasi :

**Table 2.4 Tabel Simbol *Commucation Diagram***

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Objek  nama objek:nama kelas | Objek yang melakukan interaksi pesan |
| Link | Relasi antar-objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri  nama objek:nama kelas |
| Arah pesan / *stimulus* | Arah pesan yang terjadi, jika pada suatu link ada dua arah pesan yang berbeda maka arah juga digambarkan dua arah pada sisi *link* |

**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal : 169)**

* + - 1. ***State Chart Diagram***

*Statechart* atau *state machine diagram* disebut sebuah diagram mesin status atau sering juga disebut diagram status yang digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika diagram sekuen digunakan untuk interaksi antar objek maka diagram status digunakan untuk interaksi didalam sebuah objek. Perubahan tersebut digambarkan dalam sebuah graf berarah. *State machine diagram* merupakan pengembangan dari lingkungan  *finite state automata* dengan penambahan beberapa fitur dan konsep baru (S, Rosa A dan M. Shalahuddin, 2013, hal : 163).

*State machine diagram* cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem. Berikut adalah contoh gambar diagram mesin status :

Status awal

Status 1

Status 3

Status 2

Status akhir

event triger

aksi

event triger

aksi

event triger

aksi

transaksi kediri sendiri

**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal 164**

**Gambar 2.5 Contoh *state machine diagram***

Berikut ini komponen–komponen dasar yang ada dalam *state machine diagram* :

**Table 2.5 Tabel Simbol *State Machine Diagram***

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| start / status awal (*initial state*) | *Star* atau *initial state* adalah *state* atau keadaan awal pada saat sistem mulai hidup |
| End / status akhir (*final state*) | *End* atau *final state* keadaan akhir dari daur hidupsuatu sistem |
| Event  Event | *Event* kegiatan yang menyebabkan perubahan setatus mesin |
| State  state | Sistem pada waktu tertentu. *State* dapat berubah jika ada *event* tertentu yang memicu perubahan tesebut |

**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal : 164**

* + - 1. ***Activity Diagram***

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

Diagram aktifitas juga digunakan untuk mendefinisikan hal berikut (S, Rosa A, dan M. Shalahuddin, 2013) :

1. Rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan peroses bisnis sistem yang di definisikan.
2. Urutan atuu pengelompokan tampilan dari sistem / *user intreface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian diamana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol- simbol yang ada pada *activity diagram* :

**Tabel 2.6 Tabel Simbol *Activity Diagram***

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Status awal | status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  aktivitas | aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kerja keras |
| Percabangan / decision | asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| Penggabungan / join | asosiasi penggabungan dimana lebih dari satu akvitas digabungkan menjadi satu |
| Status akhir | status akhir yang dilakukan sistem, sebuah diagram aktivitas memiki sebuah status akhir |
| Swimlane  **Nama swimlane** | Memisahkan organisasi bisnis yang bertanggung jawab terhadap yang terjadi |

**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal : 162**

* + - 1. ***Deployment Diagram***

Diagram *deployment* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut :

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node* dan *hardware*.
2. Sistem client/ server.

Berikut adalah simbol-simbol pada deployment diagram :

**Table 2.7 Tabel Simbol *Deployment Diagram***

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Package  package | *Package* merupakan sebuah bungkusan dari satu atau lebih *node* |
| Node    nama\_node | Biasanya mengacu pada perangkat keras (*hardware*), perangkat lunak yang tidak dibuat sendiri, jika di dalam *node* disertakan komponen mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen |
| Kebergantungan/*dependency* | Kebergantungan antar *node*, arah panah mengarah pada *node* yang dipakai |
| Link | Relasi antar *node* |

**Sumber: S, Rosa A dan M. Shalahudin, 2013, hal : 154**

1. **Kecerdasan Buatan (*Artificial Inteligence*)**

*Artificial Intelligence* (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang menaruh perhatian pada pengotomatisasi tingkah laku cerdas. Pernyataan tersebut juga dapat dijadikan defenisi dari AI. Defenisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada *sound theoretical* (teori suara) dan prinsip-prinsip aplikasi dari bidangnya. (Rahman, Fakhrul, Eka Praja Wiyata Mandala, dan Teri Ade Putra, 2017).

* + 1. **Definisi Kecerdasan Buatan**

Kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia (Hartati, Sri dan Sari Iswanti, 2008, hal : 1).

Kecerdasan buatan berasal dari bahasa inggris “*Artificial Intelligence*” atau disingkat AI didefinisikan sebagai kecerdasan entitas ilmiah. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), logika fuzzy, jaringan syaraf tiruan dan robotika (Mujilahwati, Siti, 2014).

Berikut definisi kecerdasan buatan yang telah didefinisikan oleh beberapa ahli (Sutojo, T, Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 2) :

1. Alan Turing

Menetapkan bahwa defenisi kecerdasan buatan yaitu jika komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan.

1. John McCarthy

Kecerdasan sebagai kemempuan untuk mencapai sukses dalam menyelesaikan suatu permasalahan.

1. Herbert Alexander Simon

Kecerdasan buatan merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

1. Rich and Knight

Kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang ada saat ini dapat dilakukan lebih baik oleh manusia.

1. Encuclopedi Britannica

Kecerdasan buatan merupakan cabang dari ilmu komputer yang dalam merepresentasikan pengetahuan lebih banyak menggukan bentuk simbol-simbol dari pada bilangan dan memproses informasi berdasarkan metode heuristis atau dengan berdasarkan sejumlah aturan.

Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peranan komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu yang bias dikerjakan manusia (Wijaya, Edi, 2013).

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidakakan dapat menyelesaikan masalah dengan baik. Demikian pula, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik. Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu pada AI akan mencoba untuk memberikan beberapa metoda untuk membekali komputer dengan kedua komponen tersebut agar komputer bias menjadi mesin yang pintar (Wijaya, Edi, 2013).

Pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain (Wijaya, Edi, 2013) :

1. Sudut pandang kecerdasan

Kecerdasan buatan akan membuat mesin menjadi ‘cerdas’ (mampu berbuat seperti apa yang dilakukan oleh manusia).

1. Sudut pandang penelitian

Kecerdasan buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

1. Sudut pandang bisnis

Kecerdasan buatan adalah kumpulan peralatan yang sangat *powerful* dan metodologis dalam menyelesaikan masalahmasalah bisnis.

1. Sudut pandang pemrograman

Kecerdasan buatan meliputi studi tentangpemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

* + 1. **Kecerdasan Alami dan Kecerdasan Buatan**

Menurut Turban, Jika dibandingkan kecerdasan alami, kecerdasan buatan memiliki keuntungan komersial, antara lain (Sutojo, T, Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 10):

1. Kecerdasan buatan lebih bersifat permanen.

Kecerdasan alami akan cepat mengalami perubahan. Kecerdasan buatan akan cepat mengalami perubahan. Kemampuan kecerdasan buatan tidak akan pernah berubah selama programnya tidak diubah oleh programmer.

1. Kecerdasan buatan lebih mudah untuk diduplikasi dan disebarkan.
2. Kecerdasan buatan lebih murah dibandingkan kecerdasan alami.
3. Kecerdasan buatan bersifat konsisten.
4. Kecerdasan buatan dapat didokumentasikan.

Solusi dan keputusan yang dibuat oleh kecerdasan buatan dapat didokumentasikan dengan mudah karena disimpan di dalam hard disk dan pencarian datanya relatif lebih mudah dilacak. Sedangkan kecerdasan alami, hal ini sangat sulit dilakukan.

1. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibandingkan dengan kecerdasan alami.
2. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami.

Sementara itu, kecerdasan alami memberikan keuntungan sebagai berikut (Sutojo, T, Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 11)

1. Kreatif

Pengetahuan seorang manusia selalu bertambah seiring dengan perkembangan waktu. Sifat bosan manusia pun mengakibatkan dia harus berfikir kreatif untuk mencari solusi-solusi terbaru. Berbeda dengan kecerdasan buatan, penambahan pengetahuan harus dilakukan pada sistem yang telah dibangun.

1. Kecerdasan alami memungkinkan orang menggunakan pengalaman secara langsung, sedangkan kecerdasan buatan harus bekerja dengan input-input simbolik.
2. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

Kecerdasan alamiah adalah kecerdasan yang dimiliki oleh manusia. Jika dibandingkan dengan kecerdasan buatan, ada beberapa keuntungan kecerdasan buatan dibanding kecerdasan alamiah, yaitu (Wijaya, Edi, 2013):

1. Lebih permanen

Kecerdasan alamiah akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang system komputer dan program tidak diubah.

1. Memberikan kemudahan dalam duplikasi dan penyebaran.

Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama, dan keahlian itu juga tidak akanpernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.

1. Relatif lebih murah dan kecerdasan alamiah.

Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

1. Konsisten dan teliti.

Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.

1. Dapat didokumentasi.

Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.

1. Dapat mengerjakan beberapa *task* dengan lebih cepat dan lebih baik dibanding manusia.

Sedangkan, keuntungan kecerdasan alamiah dibanding kecerdasan buatan (Wijaya, Edi, 2013) :

1. Bersifat lebih kreatif

Kemampuan untuk menambah ataupun memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.

1. Dapat melakukan proses pembelajaran secara langsung, sementara AI harus mendapatkan masukan berupa simbol dan representasi.
2. Fokus yang luas sebagai referensi untuk pengambilan keputusan, sebaiknya AI menggunakan fokus yang sempit.

Komputer dapat digunakan untuk mengumpulkan informasi tentang obyek, kegiatan (*events*), proses dan dapat memproses sejumlah besar informasi dengan lebih efisien dari yang dapat dikerjakan manusia. Namun di sisi lain, manusia dengan menggunakan insting dapat melakukan hal yang sulit diprogram pada komputer, yaitu kemampuan mengenali (*recognize*) hubungan antara hal-hal tersebut, menilai kualitas dan menemukan pola yang menjelaskan hubungan tersebut. (Wijaya, Edi, 2013).

* + 1. **Ruang Lingkup Kecerdasan Buatan**

Adapun ruang Lingkup utama dalam kecerdasan buatan adalah (Wijaya, Edi, 2013) :

1. Sistem Pakar (*Expert System*)

Disini, komputer digunakan sebagai sarana untuk menyimpan pengetahuan parapakar. Dengan demikian, komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.

1. Pengolahan Bahasa Alami (*Natural Language Processing*).

Dengan pengolahan bahasa alami ini diharapkan *user* dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.

1. Pengenalan Ucapan (*Speech Recognition*).

Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer menggunakan suara.

1. Robotika & Sistem Sensor (*Robotics & Sensory Systems*).
2. *Computer Vision*, mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer.
3. *Intelligent Computer-aided Instruction*.

Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

1. *Game Playing*.
2. **Sistem Pakar**

Sistem pakar adalah program komputer yang menyimulasikan penilaian dan perilaku manusia atau organisasi yang memiliki pengetahuan dan pengalaman ahli dalam bidang tertentu. Biasanya, sistem seperti ini berisi basis pengetahuan yang berisi akumulasi pengalaman dan satu set aturan untuk menerapkan pengetahuan dasar untuk setiap situasi tertentu (Budiharto, Widodo, dan Derwin Suhatono, 2014, hal : 132).

Sistem Pakar (*Expert System*) adalah salah satu cabang dari AI yang membuat pengguna secara luas *knowledge* yang khusus untuk menyelesaikan masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu yaitu pakar yanng mempunyai knowledge atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam yang di miliki (Gulo, Amonius Asmin Hardi Saputra, dan Muhammad Syahrizal, 2018)

* + 1. **Definisi Sistem Pakar**

Sistem Pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam (Kusrini, 2008).

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, T, Edy Mulyanto, dan Vincent Suhartono, 2011).

Definisi sistem pakar yang paling dikenal adalah (Hartati, Sri dan Sari Iswanti, 2008, hal : 3):

1. Sistem berbasis komputer yang menggunakan pengetahuan fakta, dan teknik penalaran dalam memecahkan masalah, yang biasanya hanya dapat diselesaikan oleh seorang pakar dalam bidang tertentu (Martin dan Oxman).
2. Sebuah model dan prosedur terkait yang memeaparkan, dalam satu domain tertentu, derajat keahlian dalam pemecahan masalah yang sebanding dengan seorang pakar manusia (lgnizio).
3. Sistem yang menggunakan pengetahuan manusia yang dimasukkan ke dalam komputer untuk memecahkan masalah-masalah yang biasanya diselesaikan oleh pakar (Turban dan Aronson).
4. Sistem pakar adalah sistem komputer yang mengemulasi kemamapuan pengambilan keputusan seorang manusia ahli. (Giarratano & Riley).

Untuk membuat system pakar yang efektif harus dipilih representasi pengetahuan yang tepat.Pemilihan representasi pengetahuan yang tepat akan membuat system pakar dapat mengakses basis pengetahuan tersebut untuk keperluan pembuatan keputusan.Beberapa model reprsentasi pengetahuan yang penting (Hartati, Sri dan Sari Iswanti, 2008, hal : 22):

1. Jaringan semantic (*Semantik Nets*)
2. Bingkai (*Frame*)
3. Kaidah Prosuksi (*Production Rule*)
4. Logika Predikat (*Predicate Logic*)

Komponen utama yang terlihat secara virtual di setiap sistem pakar adalah (Hartati, Sri dan Sari Iswanti, 2008, hal : 5):

1. Basis pengetahuan

Basis pengetahuan merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu.Pengetahuan ini diperoleh dari akumulasi pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti yang telah disebutkan sebelumnya.

1. Mesin Inferensi

Membuat inferensi merupakan otak dari sistem pakar,berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar,biasa dikatakan sebagai mesin pemikir (*Thinking Machine*).

1. Antarmuka Pengguna

Sistem Pakar menggantikan seorang pakar dalam suatu situasi tertentu,maka sistem harus menyediakan pendukung yang diperlukan oleh pemakai yang tidak memahami masalah teknis.

1. Memori Kerja

Merupakan bagian dari sistem pakar yang menyimpan fakta-fakta yang diperoleh saat dilakukan proses konsultasi.

1. Fasilitas Penjelasan

Proses menentukan keputusan yang dilakukan oleh mesin inferensi selama sesi konsultasi mencerminkan proses penalaran seorang pakar.

1. Fasilitas Akuisisi Pengetahuan
2. Pengetahuan pada sistem pakar dapat ditambahkan kapan saja pengetahuan baru diperoleh atau saat pengetahuan yang sudah ada tidak berlaku lagi.
   * 1. **Kelebihan Sistem Pakar**

Sistem pakar banyak digunakan pada aplikasi terkini dan kompleks karena (Budiharto, Widodo dan erwin Suharto, 2014):

1. Sistem pakar dapat bertindak sebagai konsultan, instruktur, atau pasangan/rekan.
2. Meningkatkan *availability* atau kepakaran tersedia pada semua perangkat komputer.
3. Mengurangi bahaya.
4. Permanen.
5. Pengetahuan dapat tidak lengkap, namun keahlian dapat diperluaskan sesuai kebutuhan. Program konvensional harus “lengkap” sebelum mereka dapat digunakan.
6. Database yang cerdas, sistem pakar dapat digunakan untuk mengakses database secara cerdas, misalnya data mining.
   * 1. **Kekurangan Sistem Pakar**

Ada beberapa kekurangan yang ada pada sistem pakar, diantaranya(Sutojo, T., Edy Mulyanto, and Vincent Suhartono, 2011, hal : 161).

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.
   * 1. **Manfaat Sistem Pakar**

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan, diantaranya (Sutojo, T., Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 160).

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi dilingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernak menjadi bosan dan kelelahan.
8. Meningkatkan kapabilitas sistem komputer.
9. Mampu bekerja dengan informasi yang tidak lengkap.
10. Bisa digunakan sebagai media pelengkapdalam pelatihan.
11. Meningkatkan kemampuan untuk menyelesaikanmasalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.
    * 1. **Ciri-ciri Sistem Pakar**

Ciri-ciri dari sistem pakar adalah sebagai berikut (Sutojo, T., Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 162):

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/*rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran
8. Sistem dapat mengaktifkan *rule* secara searah yang sesuai, dituntun oleh dialog dengan pengguna.
   * 1. **Konsep Dasar Sistem Pakar**
9. **Kepakaran**

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik dari pada seorang yang bukan pakar. (Sutojo, T., Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 163).

Kepakaran itu sendiri meliputi pengetahuan tentang :

1. Fakta-fakta tentang bidang permasalahan tertentu.
2. Teori-teori tentang bidang permasalahan tertentu.
3. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya.
4. Aturan *heuristic*  yang harus dikerjakan dalam suatu situasi tertentu.
5. Strategi global untuk pemecahan permasalahan.
6. Pengetahuan tentang pengetahuan (*meta knowlwdge*).
7. **Pakar**

Pakar adalah seseorang yang mempunyai pengetahuan, pengetahuan, pengalaman, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Jadi seorang pakar harus mampu melakukan kegiatan-kegiatan sebagai berikut (Sutojo, T., Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 163):

1. Mengenali dan memformulasikan permasalahan.
2. Memecahkan permasalahan secara cepat dan tepat.
3. Menerangkan pemecahannya.
4. Belajar dari pengalaman.
5. Merestrukturisasi pengetahuan.
6. Memecahkan aturan-aturan.
7. Menentukan relevansi.

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu. Seorang pakar memiliki kemampuan kepakaran, yaitu (Hartati, Sri, dan Sari Iswanti, 2008, hal : 11):

1. Dapat mengenali dan merumuskan masalah.
2. Menyelesaikan masalah dengan cepat dan tepat.
3. Menjelaskan solusi dari suatu masalah.
4. Restrukturisasi pengetahuan.
5. Belajar dari pengalaman.
6. Memahami batas kemampuan.

Selain itu, pakar juga memiliki kemampuan untuk mengapikasikan pengetahuannya dan memberikan saran serta pemecahan masalah pada domain tertentu. Ini merupakan pekerjaan pakar, memberikan pengetahuan tentang bagaimana seseorang melaksanakan tugas untuk menyelesaikan masalah.

1. **Pemindahan Kepakaran**

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu(Sutojo, T., Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 163):

1. Akuisisi pengetahuan(dari pakar atau sumber lain).
2. Representasi pengetahuan(pada komputer).
3. Inferensi pengetahuan.
4. Pemindahan pengetahuan ke pengguna.
5. **Inferensi**

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil keputusan berdasarkan basis pengetahuan yang dimilikinya (Sutojo, T., Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 164).

Inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, biasa dikatakan sebagai mesin pemikir (*Thinking Machine*). Pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan (Hartati, Sri dan Sari Iswanti, 2008, hal : 5).

1. **Aturan-aturan (*Rule*)**

Cara mempresentasikan pengetahuan berbasis kaidah memanfaatkan apa yang disebut dengan kaidah, yang tak lain adalah pertanyaan IF-THEN dimana bagian THEN akan bernilai benar jika satu atau lebih sekumpulan fakta atau hubungan antar fakta diketahui benar, memenuhi bagian IF (Hartati, Sri dan Sari Iswanti, 2008, hal : 41).

Kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based-systems*), yaitu pengetahuan disimpan terutama dalam bentuk *rule,* sebagai proedur-prosedur pemecahan masalah (Sutojo, T., Edy Mulyanto, and Vincent Suhartono, 2011, hal : 165).

1. **Kemampuan Menjelaskan (*Explanation Capability*)**

Faslitias lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (explanation). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya (Sutojo, T., Edy Mulyanto, dan Vincent Suhartono, 2011, hal : 165).

1. **Metode *Case Based Reasoning***

Metode *Case Based Reasoning* (CBR) adalah sistem yang bertujuan untuk menyelesaikan suatu kasus baru dengan cara mengadaptasi solusi-solusi yang terdapat pada kasus-kasus sebelumnya yang mirip dengan kasus-kasus sebelumnya. Kemiripan (*Similarity*) merupakan langkah yang digunakan untuk mengenali kesamaan atau kemiripan antara kasus-kasus yang tersimpan dalam basis kasus dengan kasus yang baru (Gulo, Amonius Asmin Hardi Saputra, dan Muhammad Syahrizal, 2018).

* + 1. **Definisi *Case Based Reasoning***

*Case Based Reasoning* (CBR) merupakan sistem penalaran komputer yang menggunakan pengetahuan kasus pada masa lama, lalu menggunakannya kembali dan memberikan solusi terhadap kasus baru dengan melihat kasus lama yang paling mendekati kasus baru. (Minarni, Indra Warman, dan Wenda Handayani, 2017 ).

*Case Based reasoning* (CBR) merupakan sebuah paradigma utama dalam penalaran otomatis (*automated reasoning*) dan mesin pembelajaran (*machine learning*). Didalam CBR, seseorang yang melakukan penalaran dapat menyelesaikan masalah baru dengan memperhatikan kesamaannya dengan satu atau beberapa penyelesaian dari permasalahan sebelumnya (Utomo, Dito Putro, dan Surya Darma Nasution).

* + 1. **Tahap *Case Based Reasoning***

*Case Based Reasoning* (CBR) menggunakan pendekatan kecerdasan buatan (*Artificial Intelligent*) yang menitikberatkan pemecahan masalah dengan didasarkan pada *knowledge* dari kasus-kasus sebelumnya (Kusuma, Diki Andita, dan Chairani, 2018 )

Adapun siklus dari metode CBR dapat dilihat pada Gambar 2.6 :



**Sumber: Gulo, Amonius Asmin Hardi S, dan Muhammad Syahrizal, 2018**

**Gambar 2.6 Siklus Metode *Case Based Reasoning***

Secara detail *Case Based Reasoning* terbagi dalam empat tahap, yaitu :

1. *Retrieve*

Pada saat terjadi permasalahan baru, pertama-tama sistem akan melakukan proses *retrieve*. Proses ini akan melakukan dua langkah pemrosesan, yaitu pengenalan masalah dan pencari persamaan masalah pada *database*. Memperoleh kembali kasus yang paling menyerupai/relevan (*similar*) dengan kasus yang baru. Tahap *retrieve* ini dimulai dengan menggambarkan/menguraikan sebagian masalah, dan diakhiri jika ditemukannya kecocokan terhadap masalah sebelumnya yang tingkat kecocokannya terhadap masalah sebelumnya yang tingkat kecocokan paling tinggi. Dengan rumus sebagai berikut

***Similarity (****p,q) =* ......................................(2.1)

**Sumber : Gulo, Amonius Asmin Hardi Saputra, and Muhammad Syahrizal**

Keterangan :

*p* : kasus baru

*q* : kasus yang ada dalam penyimpanan *(case)*

*w* : *weight* (bobot yang diberikan pada atribut ke-*i)*

*s* : *similarity* ( nilai kemiripan )

1. *Reuse*

Menggunakan kembali informasi dan pengetahuan dari basis kasus untuk memecahkan masalah kasus baru, sehingga menghasilkan usulan solusi dimana mungkin diperlukan suatu adaptasi dengan masalah yang baru tersebut

1. *Revise*

Proses ini informasi tersebut akan dikalkulasi, dievaluasi, dan diperbaiki kembali untuk mengatasi kesalahan-kesalahan yang terjadi pada permasalahan baru.

1. *Retain*

Proses ini akan mengindeks, mengintegrasi, dan mengekstrak solusi yang baru. Selanjutnya, solusi baru itu akan disimpan ke dalam *knowledge base* untuk menyelesaikan permasalahan yang akan datang. Tentunya, permasalahan yang akan diselesaikan adalah permasalahan yang memiliki kesamaan dengannya (Gulo, Amonius Asmin Hardi Saputra, dan Muhammad Syahrizal, 2018).

1. **Sekilas Tentang *Web***

*Website* adalah sekumpulan halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan sebuah komponen yang terdiri dari teks, gambar, suara animasi sehingga menjadi media informasi yang menarik untuk dikunjungi oleh orang lain (Mandala, Eka Praja Wiyata, 2015, hal : 2).

Secara umum, *website* digolongkan menjadi tiga jenis, yaitu (Mandala, Eka Praja Wiyata, 2015, hal : 3) :

1. *Website* Statis

*Website* statis merupakan website yang memiliki isi tidak dimaksudkan untuk diperbarui secara berkala, sehingga pengaturan ataupun pemutakhiran isi atas situs *web* tersebut dilakukan secara manual.

1. *Website* Dinamis

*Website* dinamis merupakan *website* secara spesifik didesain agar isi yang terdapat dalam situs tersebut dapat diperbarui secara berkala dengan mudah. Sesuai dengan namanya, isi yang terkandung dalam *website* ini umumnya akan berubah setelah melewati satu periode tertentu.

1. *Website* Interaktif

*Website* interaktif adalah *website* yang saat ini memang sedang banyak digemari. Salah satu contoh *website* interaktif ini adalah blog dan forum. Di *website* ini *user* bisa berinteraksi dan beradu argument mengenai apa yang menjadi pemikiran mereka. Biasanya *website* seperti memiliki moderator untuk mengatur supaya topik yang diperbincangkan tidak melenceng dari alur pembicaraan.

1. **Pengenalan Aplikasi Server**

Aplikasi *Server* adalah aplikasi pada sistem komputer yang berfungsi melayani permintaan akses dari komputer pengguna atau klien (Mandala, Eka Praja Wiyata, 2015, hal : 16).

* + 1. ***Web Server***

*Web server* itu dapat dikatakan sebagai suatu program komputer yang memiliki tugas menerima permintaan HTTP dari komputer klien, yang dikenal dengan nama *web browser* dan melayani mereka dengan menyediakan respon HTTP berupa konten data, biasanya berupa halaman web yang terdiri dari dokumen HTML dan objek yang terkait seperti gambar, text, suara, dan sebagainya (Mandala, Eka Praja Wiyata, 2015, hal : 17).

* + 1. ***Database Server***

*Database Server* adalah sebuah program komputer yang menyediakan layanan pengelolaan basis data dan melayani komputer atau program aplikasi basis data yang menggunakan model klien/*server.* Istilah ini juga merujuk kepada sebuah komputer (umumnya merupakan *server)* yang didedikasikan untuk menjalankan program yang bersangkutan. *Database Management system* (DBMS) pada umumnya menyediakan fungsi-fungsi *server* basis data, dan beberapa DBMS (seperti halnya MySQL atau Microsoft SQL Server) sangat bergantung kepada model klien-server untuk mengakses basis datanya (Mandala, Eka Praja Wiyata, 2015, hal :17).

* + 1. ***Hypertext Preprocessor* (PHP)**

*Hypertext Preprocessor (PHP)*adalah bahasan skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS (Content Management System) (Mandala, Eka Praja Wiyata, 2015, hal : 24).

1. ***Pneumonia***

*Pneumonia* adalah proses inflamasi pada parenkim paru dan merupakan penyebab utama morbiditas dan mortalitas anak berusia di bawah lima tahun, terutama di negara berkembang. *Pneumonia* merupakan masalah kesehatan yang menyebabkan angka kematian tinggi di dunia dan menjadi salah satu dari 5 penyebab utama kematian pada anak usia di bawah 5 tahun di negara berkembang, dengan jumlah kematian sekitar 3 juta kematian/tahun. Tingkat kematian anak dibawah usia lima tahun di sebagian besar negara berkembang berkisar 60-100 per 1000 kelahiran hidup, seperlima dari kematian ini disebabkan oleh *pneumonia*. *Pneumonia* yang terjadi pada balita akan memberikan gambaran klinik yang lebih buruk daripada orang dewasa karena pada balita sistem pertahanan tubuh yang dimiliki relatif rendah (Monita, Osharinanda, Finny Fitry Yani, dan Yuniar Lestari, 2015).

Pola bakteri pentebab *pneumonia* biasanya berubah sesuai dengan distribusi umur pasien. Walaupun *pneumonia* viral dapat ditatalaksana tanpa antibiotic, tapi umumnya sebagian besar pasien diberi antibiotik karena infeksi bakteri sekunder tidak dapat disingkirkan. Di Negara berkembang, *pneumonia* pada anak terutama disebabkan oleh bakteri. Berdasarkan tempat terjadinya infeksi, dikenal dua bentuk *pneumonia* yaitu :

1. *Pneumonia* masyarakat (*community-acquired pneumonia*) bila infeksinya terjadi di masyarakat
2. *Pneumonia* RS atau *pneumonia* nosokomial (*hospital-acquired pneumonia*) bila infeksinya didapat di rumah sakit.

Selain berbeda dalam lokasi tempat terjadinya infeksi, kedua bentuk *pneumonia* ini juga berbeda dalam spectrum etiologi, gambaran klinis, penyakit dasar atau penyakit penyerta, dan prognosisnya. *Pneumonia* yang didapat di rumah sakit sering merupakan infeksi sekunder pada berbagai penyakit dasar yang sudah ada, sehingga spectrum etiologinya berbeda dengan infeksi yang terjadi di masyarakat. *Pneumonia* yang didapat di rumah sakit memerlukan penanganan khusus sesuai dengan penyakit dasarnya (Rahajoe, Nastiti N, Bambang Supriyanto, Darmawan Budi Setyanto, 2008, hal : 362).

Pada penelitian ini, adapun dr. Laura Zeffira sebagai pakar menyatakan ada 2 klasifikasi *pneumonia* yaitu :

1. *Pneumonia* Ringan

Gejala-gejalanya :

1. Batuk
2. Pilek
3. Demam
4. Muntah
5. Cuping hidung
6. Tarikan dinding dada atau retraksi
7. Warna kulit kebiru-biruan atau sianosis
8. Napas cepat atau takipnea
9. Lingkungan terjangkit asap rokok
10. Anak belum diberikan vitamin A
11. Berat anak lahir kurang

Solusinya :

* 1. Diberikan antibiotik amoksisilin atau kotrimoksazol
  2. Menurunkan panas anak
  3. Pengobatan simptomatis atau metode pengobatan untuk mengurangi dan meringankan gejala penyakit yang dirasakan pasien

1. *Pneumonia* berat
2. Sesak napas
3. Cuping hidung
4. Tarikan dinding dada atau retraksi
5. Warna kulit kebiru-biruan atau sianosis
6. Belum imunisasi
7. Napas cepat atau takipnea
8. Ada bunyi ronki

Solusinya :

* 1. Untuk memastikan diagnosis, dilakukan rontgen
  2. Dilakukan rawat inap dirumah sakit
  3. Pemberian antibiotik beta-laktam, ampisilin, atau amoksisilin, dikombinasikan dengan kloramfenikol
  4. Terapi antibiotik dilakukan selama 7-10 hari pada pasien dengan pneumonia tanpa komplikasi

Diagnosis etiologik berdasarkan pemeriksaan mikrobiologis atau serologis merupakan dasar terapi yang optimal. Akan tetapi, penemuan bakteri penyebab tidak selalu mudah karena memerlukan laboratorium penunjang yang memadai. Akibat tingginya angka morbilitas dan mortalitas *pneumonia* pada balita, maka dalam upaya penanggulangannya, WHO mengembangkan pedoman diagnosis dan tatalaksana yang sederhana. Gejala klinis sederhana tersebut meliputi napas cepat, sesak napas, dan berbagai tanda bahaya agar anak segera dirujuk ke pelayanan kesehatan. Tanda bahaya pada anak berusia 2 bulan-5 tahun adalah tidak dapat minum, kejang, kesadaran menurun, stridor, dan gizi buruk. Sedangkan tanda bahaya untuk bayi di bawah 2 bulan adalah malas minum, kejang, kesadaran menurun, demam atau badan terasa dingin.

**BAB III**

**METODOLOGI PENELITIAN**

* 1. **Kerangka Penelitian**

Kerangka penelitian adalah urutan kegiatan yang akan dilakukan dalam suatu penelitian. Agar langkah-langkah yang diambil penulis dalam perancangan ini tidak melenceng dari pokok pembicaraan dan lebih mudah dipahami. Maka penulis membentuk kerangka penelitian sebagai berikut :

**Penelitian Pendahuluan**

**Pengumpulan Data**

**Analisa**

**Perancangan**

**Pengujian**

**Implementasi**

**Gambar 3.1 Kerangka Penelitian**

* 1. **Tahapan Penelitian**

Tahapan penelitian adalah tahapan yang akan dilakukan untuk mempermudah dalam melakukan penelitian. Adapun tahapan-tahapan dalam penelitian ini adalah sebagai berikut :

* + 1. **Penelitian Pendahuluan**

Tingginya angka kematian anak, terutama penyebabnya penyakit *pneumonia* yang menyerang anak. Tingginya insiden *pneumonia* yaitu faktor lingkungan, status imunisasi dan status gizi pada anak. Faktor yang sangat mempengaruhi penyakit *pneumonia* adalah penyebaran mikroorganisme. Kurangnya pengetahuan orangtua tentang penyakit *pneumonia* dan cara penanganan dini. Orangtua juga mengalami kendala dalam menentukan dan mengidentifikasi jenis penyakit *pneumonia* berdasarkan level yang menyerang anak.

Penelitian pendahuluan merupakan langkah pertama dalam melakukan suatu penelitian. Tujuan dari tahap ini adalah untuk mengetahui permasalahan yang terjadi secara tepat, sehingga diharapkan penelitian dapat memberikan solusi yang paling optimal terhadap pemecahan permasalahan tersebut.

* + 1. **Pengumpulan Data**

Dalam melakukan proses pengumpulan data, penulis melakukan wawancara secara langsung ke Ibu dr. Laura Zeffira, M.Biomed yang bekerja sebagai dokter anak di Rumah Sakit Islam Siti Rahmah kota Padang. Penulis mendapatkan beberapa informasi penting yang berhubungan dengan penelitian yang sedang dilakukan.

1. **Jadwal Penelitian**

Penelitian dilakukan dengan memproses data-data yang telah didapat oleh peneliti, pengambilan data dilakukan dari bulan Oktober 2018 sampai dengan selesai. Untuk lebih rincinya dapat dilihat pada tabel dibawah ini :

**Tabel 3.1 Jadwal Penelitian**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **Bulan Ke-** | | | | | | | | | | | | | | | |
| **1** | | | | **2** | | | | **3** | | | | **4** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** | **Perencanaan Penelitian** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | **Pengumpulan Data** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | **Analisa Data dan Sistem** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | **Perancangan Aplikasi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** | **Pembuatan Aplikasi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6** | **Uji Coba** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7** | **Implementasi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8** | **Bimbingan Skripsi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **9** | **Penyusunan Skripsi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Tempat Penelitian**

Adapun tempat melakukan penelitian ini dilakukan di Rumah Sakit Islam Siti Rahmah Kota Padang. Dan juga melakukan penelitian langsung ke Rumah Sakit Siti Rahmah tepatnya di bagian poli anak di Jalan Bypass no 15 Aie Pacah Kec. Koto Tangah.

1. **Metode Penelitian**

Adapun metode penelitian yang dilakukan sebagai berikut :

1. **Penelitian Lapangan**

Pada penelitian lapangan ini dilakukan wawancara dengan dokter anak yang pernah menangani kasus penyakit *pneumonia* pada anak, untuk menganalisis masalah, serta memperoleh informasi yang dibutuhkan. Serta bagaimana seorang dokter anak melakukan mekanisme kerja dalam pendiagnosaan penyakit sehingga mendapatkan hasil diagnosa yang tepat.

1. **Penelitian Kepustakaan**

Dalam metode ini penulis membaca dan mempelajari sumber-sumber yang akan mendukung penulisan ini. Sumber tersebut dapat berupa buku-buku dan hasil penelitian. Untuk hasil penelitian dapat berupa laporan penelitian, jurnal ilmiah dan skripsi. Termasuk dalam kategori ini bahan-bahan yang dipublikasikan secara *online* (akses internet).

1. **Penelitian Laboratorium**

Penelitian yang dilakukan di laboratorium komputer dalam pembuatan aplikasi sistem pakar tentang penyakit *pneumonia* pada anak. Perangkat keras yang digunakan dalam pembangunan Aplikasi ini adalah sebagai berikut:

1. Laptop ASUS A456U
2. *Processor* Intel Core I5-7200U CPU 2.7 GHz
3. RAM 8,00 GB
4. *Harddisk* 1000 GB
5. Monitor 14”
6. Media input yaitu *Keyboard, Mouse,* dan *Flashdisk* 16GB.

Sedangkan perangkat lunak yang digunakan dalam pembangunan Aplikasi ini adalah sebagai berikut :

1. Sistem Operasi Windows 10 Home
2. Bahasa Pemrograman PHP & Database MySQL
3. *Microsoft Word* 2013
4. *Rational Rose Enterprise V7*
5. *Mowes Portable*
6. *Adobe Dreamweaver*
7. *Mozilla Firefox*
   * 1. **Analisa**
8. **Analisa Data**

Berdasarkan penelitian pendahuluan di atas, maka dibutuhkan analisa data terlebih dahulu. Hal ini bertujuan agar pemecahan masalah dapat mengasilkan solusi, bukan menjadi sebuah masalah yang baru. Data yang diperoleh berupa hasil wawancara dari dr. Laura Zeffira, M.Biomed sebagai dokter anak. Data tersebut berisi jenis penyakit *pneumonia* berdasarkan level, gejala dari penyakit tersebut serta cara menanggulanginya.

1. **Analisa Proses**

Dalam mengembangkan sistem pakar ini berdasarkan penyakit dan gejala data kasus lama. Dalam metode *Case Based Reasoning* seluruh data penyakit dan gejala diinputkan ke database, dalam perhitungannya *Case Based Reasoning* akan mencari tingkat kemiripan antara data kasus lama dengan data kasus baru yang dipilih oleh *user*, dan selanjutnya hasil dengan nilai terbesar dianggap sebagai kasus yang paling mirip.

1. **Analisa Sistem**

Analisa ini dilakukan untuk mengetahui apa saja yang dibutuhkan dalam perancangan sistem. Sehingga menghasilkan sebuah sistem yang efektif dan efisien dalam implementasinya nanti. Dimana program yang akan dibuat menggunakan bahasa pemograman *PHP* dan *MySQL* berbasis web.

* + 1. **Perancangan**

Pada tahap ini akan membuat sebuah perancangan sistem yang akan dijalankan, mulai dari menganalisa program yang sedang berjalan, dan merancang program yang akan kita jalankan tersebut.

1. **Perancangan Model**

Dalam tahap perancangan ini, Penulis menggunakan metode UML (*Unified Modelling language*)dalam melakukan perancangan model pada sistem pakar ini. UML (*Unified Modelling language*)yang akan digunakan sebagai berikut :

1. *Use Case Diagram*

Diagram inimenjelaskan tentang sebuah interaksi antara satu atau lebih aktor. Interaksi tersebut mengacu pada *case* yang adadidalam sebuah aplikasi sistem pakar ini. Dimana *Use* *case* diagram terdiri atas diagram untuk *use* *case* dan aktor.

1. *Class Diagram*

Pada diagram *class* menggambarkan struktur sistem dari segi pendefenisiaan kelas-kelas yang akan dibuat. Diagram ini akan memperlihatkan dengan jelas struktur dari setiap tabel yang ada serta menunjukkan hubungan antar tabel. Kelas-kelas tersebut terdiri dari kelas admin, gejala, konsultasi, hasil konsultasi, penyakit, user, dan pesan yang ada pada sebuah sistem pakar.

1. *Sequence Diagram*

Pada s*equence* *diagram* akan menggambarkan langkah-langkah pada *use case* yang terjadi dan respon dari aplikasi untuk setiap proses yang terjadi pada aplikasi sistem pakar. Prosesnya seperti aktivitas aktor, mulai dari *login* sampai nantinya *logout*.

1. *Communication Diagram*

Dalam *Collaboration Diagram* akan digambarkan bentuk fisik dari aplikasi sistem pakarini. Dimana interaksi antara user dan sistemnya digambarkan sesuai dengan proses yang terjadi seperti saat user menggunakan aplikasi ini. Jika diagram sekuen lebih menekankan kepada urutan waktu, diagram ini lebih menekankan kepada peran setiap objeknya.

1. *Activity Diagram*

*Diagram* ini akan menggambarkan perubahan status yang dialami oleh user pada saat menggunakan aplikasi sistem pakar. Perubahan yang terjadi itu akan digambarkan dalam bentuk graf berarah. Diagram ini nantinya akan menggambarkan semua kondisi yang mungkin muncul sebagai sebuah objek begitu pula dengan event yang ada.

1. *State Chart Diagram*

Pada diagram *activity* digambarkan alur dari aktivitas yang terjadi di dalam aplikasi sistem pakar. Dimulai dari bagaimana user mendaftarkan biodatanya, selanjutnya melakukan pemilihan terhadap gejala-gejala yang sedang dialami sampai keluarnya hasil diagnosis.

1. *Deployment Diagram*

*Deployment Diagram* akan menggambarkan detail bagaimana komponen di-*deploy* dalam sistem dan di *node* (pada mesin, server atau piranti keras) mana akan ditempatkan. *Diagram* ini juga mendefinisikan hubungan antar *node* dan *requirement.* Aplikasi sistem pakar berhubungan dengan *web server* untuk proses pemanggilan database sehingga *user* akan mendapatkan keputusan sesuai dengan gejala yang telah di inputkan oleh *user* ke dalam aplikasi sistem pakar tersebut.

1. **Perancangan *Interface***

Perancangan *interface* adalah bentuk rancangan tampilan sementara dari pembuatan aplikasi sistem pakar ini. Perancangan ini dibuat untuk memberikan penjelasan tentang tampilan yang dihadapi oleh *user* pada saat menggunakan sistem, sehingga dapat mempermudah dalam mengimplementasikan atau menggunakan aplikasi serta akan mempermudah pembangunan aplikasi antarmuka yang baik.

* + 1. **Implementasi Sistem**

Implementasi sistem merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehingga pengguna dapat memberi masukan kepada pengembangan sistem. Pada tahap ini perancangan sistem dilakukan dengan menggunakan bahasa pemrograman *PHP* dan *MySQL*.

* + 1. **Pengujian**

Pengujian ini dilakukan dengan membandingkan keluaran atau *Output* dari sistem dengan hasil perhitungan manual terhadap sejumlah rumus yang ada dalam metode CBR *.* Pengujian aplikasi dilakukan dengan melihat kesesuaian antara *output* yang diberikan sebagai hasil analisis dari aplikasi dengan kondisi yang sebenarnya. Dan setelah pengkodean selesai maka akan dilakukan proses pengujian terhadap aplikasi yang dihasilkan untuk mengetahui apakah aplikasi yang dirancang sudah berjalan dengan benar dan sesuai dengan perancangan yang dilakukan.

1. **Pengujian *Local***

Pada pengujian secara local, peneliti akan memanfaatkan sistem jaringan LAN (*Local Area Network*). Pada pengujian ini peneliti akan menggunakan tiga buah komputer yaitu dua komputer sebagai *client* dansatu komputer lainnyasebagai *server,* untuk melihat apakah website yang dijalankan di *server* bisa juga diakses melalui *client*.

1. **Pengujian *Online***

Pada pengujian ini peneliti menguji dengan cara *hosting* atau mendaftarkan *website* yang telah dibangun ke internet agar *website* bisa diakses secara *online* melalui *web browser*.

1. **Pengujian Aplikasi**

Pengujian ini dilakukan untuk pada mengetahui apakah hasil sesuai atau tidak dengan yang diharapkan. Pengujian aplikasi dilakukan dengan *black box testing* dimana pengujian yang dilakukan oleh user hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak berdasarkan aplikasi seperti tampilan aplikasi, fungsi yang ada di aplikasi.

1. **Pengujian *Interface***

Pengujian sistem merupakan tahap melakukan *testing* dengan menggunakan *smartphone* untuk mengetahui kesalahan dan kekurangan dalam sistem ini. Pengujian ini difokuskan pada fungsi dari sistem yang meliputi kesalahan dan kekurangan *interface*.

**BAB IV**

**ANALISA DAN PERANCANGAN**

1. **Analisa**
   * 1. **Analisa Data**

Tahap analisa data merupakan tahap yang paling penting dalam pengembangan sebuah sistem.Analisa data bertujuan untuk membatasi subjek dan objek yang akan diteliti agar menjadi sebuah informasi yang lebih sistematis dan mudah dimengerti. Untuk memperoleh data atau informasi pada penelitian ini, penulis terlebih dahulu melakukan kegiatan pengumpulan data yang berguna sebagai penunjang untuk menentukan objek penelitian.

Berikut ini tingkatan klasifikasi dari penyakit *pneumonia* pada anak dapat dilihat pada table berikut ini :

**Table 4.1 Data Klasifikasi *Pneumonia***

|  |  |
| --- | --- |
| **Kode Klasifikasi** | **Nama Penyakit** |
| P01 | *Pneumonia* Ringan |
| P02 | *Pneumonia* Berat |

Data gejala dari setiap penyakit *pneumonia* pada anak dapat dilihat pada tabel berikut ini :

**Tabel 4.2 Gejala Penyakit *Pneumonia* pada Anak**

|  |  |
| --- | --- |
| **Kode Gejala** | **Nama Gejala** |
| G01 | Batuk |
| G02 | Pilek |
| G03 | Demam |
| G04 | Muntah |
| G05 | Sesak napas |
| G06 | Cuping hidung |
| G07 | Tarikan dinding dada / retraksi |
| G08 | Warna kulit kebiru-biruan / sianosis |
| G09 | Belum imunisasi |
| G10 | Napas cepat / takipnea |
| G11 | Ada bunyi ronki |
| G12 | Lingkungan terjangkit asap rokok |
| G13 | Anak belum diberikan vitamin A |
| G14 | Berat anak lahir kurang |

* + 1. **Analisa Proses**

Berdasarkan wawancara dengan pakar bahwa pengelompokan gejala dapat dibagi menjadi dua, yaitu :

1. Pengelompokan gejala ringan dengan bobot parameter (w) : 3
2. Pengelompokan gejala berat dengan bobot parameter (w) : 8

(Sumber : dr. Laura Zeffira)

**Tabel 4.3 Gejala *Pneumonia* Ringan**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Klasifikasi *Pneumonia*** | **Gejala awal** | **Bobot** |
| 1 | *Pneumonia* Ringan | Batuk | 3 |
| Pilek | 3 |
| Demam | 3 |
| Muntah | 3 |
| Warna kulit kebiru-biruan / sianosis | 8 |
| Napas cepat atau takipnea | 3 |
| Cuping hidung | 8 |
| Tarikan dinding dada atau retraksi | 8 |
| Lingkungan terjangkit asap rokok | 8 |
| Anak belum diberikan vitamin A | 3 |
| Berat anak lahir kurang | 3 |

**Tabel 4.4 Gejala *Pneumonia* Berat**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Klasifikasi  *Pneumonia*** | **Gejala awal** | **Bobot** |
| 1 | *Pneumonia* Berat | Sesak napas | 3 |
| Warna kulit kebiru-biruan / sianosis | 8 |
| Napas cepat atau takipnea | 3 |
| Tarikan dinding dada atau retraksi | 8 |
| Belum imunisasi | 8 |
| Cuping hidung | 8 |
| Adanya bunyi ronki | 8 |

Tahapan analisa proses dilakukan dengan menggunakan metode *Case Based Reasoning* terdapat empat tahapan proses didalamnya, yaitu *retrive*, *reuse*, *revise* dan *retain*. Sistem ini pada umumnya berpedoman pada basis pengetahuan yang dimiliki oleh sistem itu sendiri.

Basis pengetahuan bersumber dari gejala kasus-kasus yang pernah di diagnosa sebelumnya oleh seorang pakar dan dihitung dengan tingkat kemiripannya dengan kasus baru. Berdasarkan tingkat kemiripan inilah sistem akan mendiagnosa penyakit *pneumonia* pada anak berdasarkan klasifikasi penyakitnya.

Sebagai contoh, berikut dialog *user* dengan sistem pakar serta data gejala penyakit *pneumonia* pada anak pada aplikasi sistem pakar :

Dialog *user* dengan sistem pakar :

SP : ”Apakah anak anda mengalami batuk?”

User : “YA”

SP : ”Apakah anak anda mengalami pilek?”

User : “TIDAK”

SP : ”Apakah anak anda mengalami demam?”

User : “YA”

SP : ” Apakah anak anda mengalami muntah?”

User : “ TIDAK”

SP : ”Apakah anak anda mengalami sesak napas?”

User : “TIDAK”

SP : ”Apakah terlihat cuping hidung pada anak anda?”

User : “YA”

SP : ”Apakah terlihat tarikan dinding dada pada anak anda?”

User : “TIDAK”

SP : ”Apakah anak anda mengalami warna kulit kebiru-biruan?”

User : “TIDAK”

SP : ”Apakah anak anda belum imunisasi”?

User : “YA”

SP : ”Apakah anak anda mengalami napas cepat atau takipnea?”

User : “YA”

SP : ”Apakah ada bunyi ronki pada anak anda?”

User : “TIDAK”

SP : ”Apakah lingkungan terjangkit asap rokok?”

User : “YA”

SP : ”Apakah anak belum diberikan vitamin A?”

User : “SUDAH”

SP : ”Apakah berat anak lahir kurang?”

User : “TIDAK”

**Tabel 4.5 Kasus Baru**

|  |  |  |
| --- | --- | --- |
| Kode | Gejala yang dirasakan | Bobot |
| G01 | Batuk | 3 |
| G03 | Demam | 3 |
| G06 | Cuping hidung | 8 |
| G09 | Belum imunisasi | 8 |
| G10 | Napas cepat atau takipnea | 3 |
| G12 | Lingkungan terjangkit asap rokok | 8 |

1. **Proses *Retrieve***

Proses *retrieve* merupakan proses pencarian kemiripan kasus baru dengan kasus yang ada pada basis pengetahuan. Pencarian kemiripan tersebut dilakukan dengan cara mencocokkan gejala yang di inputkan oleh pengguna dengan gejala yang ada pada basis pengetahuan. Pada proses *retrieve* ini akan dilakukan pembobotan dengan menggunakan algoritma *Nearest Neighbour Retrieval*.

Pada awal proses diagnosa pengguna akan menginputkan gejala-gejala yang dialaminya secara, selanjutnya pengguna akan mendapatkan hasil diagnosa. Sistem akan melakukan pembobotan dengan melakukan pencocokan satu per satu antara gejala-gejala yang ada didalam basis pengetahuan.

Adapun tahap pemrosesan kemiripan bobot yang dilakukan sistem akan dihitung dengan rumus berikut :

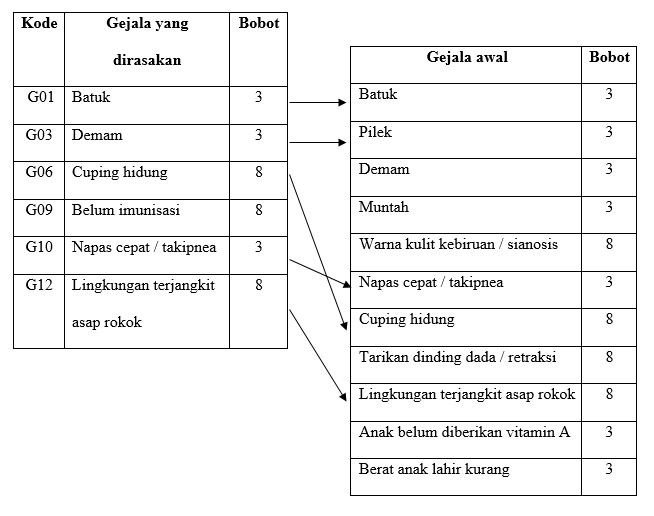
*Similiarity =*

Keterangan :

S = *similiarity* (nilai kemiripan), pada *similarity* jika terdapat kemiripan kasus akan bernilai 1, sedangkan tidak mirip, maka akan bernilai 0.

W = *weight* (bobot yang diberikan).

1. **Perhitungan Kasus *Pneumonia* Ringan**

**Gambar 4.1 Skema Kasus *Pneumonia* Ringan**

*Similiarity* (*x,x*) *=*

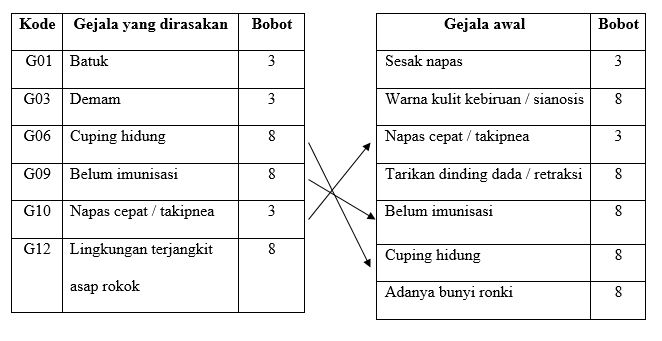
=

= = 0.471

= 47.1 %

Dari perhitungan kasus diatas memiliki tingkat kemiripan dengan kasus lama. sehingga dari perhitungan *similarity* sebesar 47.1%.

1. **Perhitungan Kasus *Pneumonia* Berat**

**Gambar 4.2 Skema Kasus *Pneumonia* Berat**

*Similiarity* (*x,x*) *=*

=

= = 0.455

= 45.5 %

Dari perhitungan kasus diatas memiliki tingkat kemiripan dengan kasus lama. sehingga dari perhitungan *similarity* sebesar 45.5%.

1. **Proses *Reuse***

Dari perhitungan kasus diatas, kasus yang memiliki bobot kemiripan tertinggi dengan kasus lama adalah *pneumonia* ringan dengan nilai 47.1%.

Pada proses *reuse*, solusi yang diberikan adalah solusi dengan bobot kemiripan kasus yang ada pada pengetahuan dengan kasus baru yang paling tinggi.

1. **Proses *Revise***

Proses ini dilakukan untuk memperbaiki solusi yang diusulkan kemudian mengujinya pada kasus nyata (simulasi) dan diperlukan kembali untuk memperbaiki solusi tersebut agar cocok dengan kasus yang baru.

1. **Proses *Retain***

Proses *retain* yaitu menyimpan pengalaman untuk memecahkan masalah yang akan datang kedalam basis pengetahuan.

* + 1. **Analisa Sistem**

Pada tahap ini, dilakukan analisa bagaimana sistem pakar diagnosa penyakit *pneumonia* pada anak yang akan dirancang. Melalui perancangan sistem ini apakah *user* dapat terbantu dengan permasalahan yang dialaminya atau *user* merasa bingung dan tidak mengerti tentang informasi yang telah diberikan.

Perancangan sistem pakar ini akan dirancang dalam bentuk website. Seorang *user non member* dapat mengakses sistem yang terdapat dimenu diantaranya menu *home* yang berisi tentang penyakit *pneumonia* pada anak, menu petunjuk cara konsultasi, menu registrasi dan menu login yang digunakan untuk login ke sistem. Menu konsultasi berisi pelaksaan konsultasi terhadap sistem pakar sebelumnya harus login sebagai *member* terlebih dahulu. Untuk admin tersedia beberapa menu yaitu menu beranda untuk admin, menu kelola tingkat klasifikasi *pneumonia* pada anak, menu kelola gejala penyakit *pneumonia* pada anak, menu data user konsultasi, menu data hasil konsultasi.

Adapun dalam penelitian yang dilakukan menggunakan perangkat komputer sebagai alat bantu dalam penyelesaian masalah. Adapun spesifikasi *hardware* dan *software* yang digunakan adalah sebagai berikut :

1. Perangkat Keras (*Hardware*)
2. Laptop ASUS A456U
3. Processor Intel Core i5-7200U CPU 2.7 GHz
4. Memory 8 GB
5. Hardisk 1000 GB
6. Flashdisk kapasitas 16 GB
7. Serta *hardware* pendukung lainnya.
8. Perangkat Lunak (*Software*):
9. Sistem Operasi Windows 10 Home
10. *Microsoft Word* 2013
11. *Rational Rose Enterprise V7*
12. Mowes, Database MySQL
13. Adobe Dreamweaver CS3
14. *Mozilla Firefox*
15. Serta *software* pendukung lainnya.
16. **Perancangan**

Perancangan dimulai dengan mengumpulkan data dan fakta yang mendukung perancangan sistem dengan mengadakan konsultasi bersama pakar. Selanjutnya akan dilakukan tahap perancangan yang terbagi menjadi perancangan model dan perancangan interface. Dalam Perancangan model ini akan menggunakan 7 diagram *Unified Modelling Language* (UML) untuk perancangan model sistem pakar ini.

* + 1. **Perancangan Model**

Adapun perancangan model *Unified Modeling Language* (UML) dalam penelitian ini adalah sebagai berikut :

* + - 1. ***Use case Diagram***

*Use Case Diagram* menggambarkan bagaimana proses–proses interaksi yang dilakukan oleh aktor terhadap sistem.

1. Tabel Definisi aktor

Aktor adalah seseorang yang menggunakan aplikasi sistem pakar.

**Tabel 4.6 Definisi Aktor**

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| **1.** | *Non member* | Aktor sebagai pengunjung umum atau sebagai calon member sistem pakar. |
| **2** | *Member* | Aktor yang bisa melakukan kegiatan konsultasi pada sistem pakar. |
| **3** | *Admin* | Aktor yang mengelola dan mengontrol segala kegiatan yang ada pada sistem pakar. |

1. Tabel Definisi pada *Use Case*

*Use case* adalah aktivitas atau kegiatan-kegiatan yang akan terjadi di dalam sistem pakar antara aktor dan *use case*.

**Tabel 4.7 Definisi *Use case***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | ***Usecase*** | **Deskripsi** | **Aktor** |
|  | Melihat *Home* | Berandamerupakan tampilan awal dari sistem yang berisi tentang informasi aktivitas yang dapat dilakukan oleh *non member* dan *member* | *Non Member*, *Member*, *Admin* |
|  | Melihat Info *Pneumonia* pada Anak | Tampilan berisi informasi tentang penyakit *pneumonia* pada anak. | *Non Member*, *Member*, *Admin* |
|  | Melakukan Registrasi | Tampilan untuk pengunjung yang ingin mendaftar sebagai member sistem pakar. | *Non Member* |
|  | *Login* | *Login* merupakan aktivitas untuk masuk ke dalam sistem pakar. | *Member*, *Admin* |
|  | Melihat Petunjuk Penggunaan | Halaman yang berisi tentang petunjuk penggunaan sistem. | *Non Member*, *Member*, *Admin* |
|  | Melakukan Konsultasi | Merupakan proses untuk melakukan konsultasi di sistem. | *Member* |
|  | Melihat *Home Admin* | *Home Admin* merupakan tampilan dari sistem yang berisi tentang informasi aktivitas untuk *admin* | *Admin* |
|  | Lihat Data Penyakit | Merupakan aktivitas yang ada pada mengelola penyakit yaitu melihat data penyakit. | *Admin* |
|  | Tambah Data Penyakit | Merupakan aktivitas yang ada pada mengelola penyakit yaitu menambah data penyakit. | *Admin* |
|  | Edit Data Penyakit | Merupakan aktivitas yang ada pada mengelola penyakit yaitu mengedit data penyakit. | *Admin* |
|  | Hapus Data Penyakit | Merupakan aktivitas yang ada pada mengelola penyakit yaitu menghapus penyakit. | *Admin* |
|  | Lihat Data Gejala | Merupakan aktivitas yang ada pada mengelola data gejala yaitu melihat gejala. | *Admin* |
|  | Tambah Data Gejala | Merupakan aktivitas yang ada pada mengelola data gejala yaitu menambah gejala. | *Admin* |
|  | Edit Data Gejala | Merupakan aktivitas yang ada pada mengelola data gejala yaitu mengedit gejala. | *Admin* |
|  | Hapus Data Gejala | Merupakan aktivitas yang ada pada mengelola data gejala yaitu menghapus gejala. | *Admin* |
|  | Lihat Data *Rule* | Merupakan aktivitas yang ada pada mengelola data rule yaitu melihat *rule*. | *Admin* |
|  | Tambah Data *Rule* | Merupakan aktivitas yang ada pada mengelola data rule yaitu menambah *rule*. | *Admin* |
|  | Hapus Data *Rule* | Merupakan aktivitas yang ada pada mengelola data rule yaitu menghapus data rule. | *Admin* |
|  | Lihat *Member* | Merupakan aktivitas yang ada pada mengelola *member* yaitu melihat data *member*. | *Admin* |
|  | Tambah *Member* | Merupakan aktivitas yang ada pada mengelola user yaitu menambah *member* | *Admin* |
|  | Hapus *Member* | Merupakan aktivitas yang ada pada mengelola *member* yaitu menghapus data *member*. | *Admin* |
|  | Lihat Hasil Konsultasi | Merupakan aktivitas yang ada pada mengelola konsultasi yaitu melihat data konsultasi. | *Admin* |
|  | Hapus Hasil Konsultasi | Merupakan aktivitas yang ada pada mengelola konsultasi yaitu menghapus data konsultasi. | *Admin* |

1. Tabel Skenario pada *Use Case*
2. Skenario *Non Member*

Berikut ini adalah scenario dari aktor *non* *member* yang telah didefinisikan sebelumnya :

Nama *Use case* : Lihat *home*

Skenario :

**Tabel 4.8 Skenario Lihat *Home***

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu *home* |  |
|  | 1. Menampilkan halaman *home* |

Nama *Use case* : Lihat Info Penyakit

Skenario :

**Tabel 4.9 Skenario Lihat Info Penyakit**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih Info Penyakit *Pneumonia* pada Anak |  |
|  | 1. Menampilkan halaman Info Penyakit *Pneumonia* pada Anak |

Nama *Use* *case* : Tambah Registrasi

Skenario :

**Tabel 4.10 Skenario Tambah Registrasi**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu register |  |
|  | 1. Menampilkan halaman register |
| 1. Melakukan pendaftaran |  |
|  | 1. Data Berhasil diinput |
|  | 1. Menampilkan pesan data berhasil disimpan. |
| **Skenario Alternatif** |  |
| 1. Memilih menu register |  |
|  | 1. Menampilkan halaman registrasi |
|  | 1. Data gagal diinput |
| 1. Memperbaiki data masukan |  |
|  | 1. Menyimpan data yang telah teregister |
|  | 1. Menampilkan pesan sukses disimpan |

Nama *Use* *case* : Lihat Petunjuk

Skenario :

**Tabel 4.11 Skenario Lihat Petunjuk**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih Petunjuk |  |
|  | 1. Menampilkan halaman Petunjuk Penggunaan |

1. Skenario *Member*

Berikut ini adalah skenario dari aktor *member* yang telah didefinisikan sebelumnya :

Nama *Use case* : Lihat *home member*

Skenario :

**Tabel 4.12 Skenario Lihat *Home* *Member***

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih Menu home member |  |
|  | 1. Menampilkan halaman home member |

Nama *Use* *case* : Lihat Info Penyakit

Skenario :

**Tabel 4.13 Skenario Info Penyakit**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih Info*Pneumonia* Pada Anak |  |
|  | 1. Menampilkan halaman Info Penyakit *Pneumonia* Pada Anak |

Nama *Use* *case* : Melakukan Konsultasi

Skenario :

**Tabel 4.14 Skenario Melakukan Konsultasi**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu konsultasi |  |
|  | 1. Menampilkan halaman konsultasi |
| 1. Melakukan konsultasi |  |
|  | 1. Menerima dan mengecek data konsultasi |
|  | 1. Menampilkan hasil konsultasi |
| **Skenario Alternatif** |  |
| 1. Memilih menu konsultasi |  |
|  | 1. Menampilkan halaman konsultasi |
| 1. Melakukan konsultasi |  |
|  | 1. Salah memasukan data konsultasi |
| 1. Memperbaiki data masukan yang salah |  |
|  | 1. Menyimpan data yang telah melakukan konsultasi |
|  | 1. Menampilkan hasil konsultasi. |

Nama *Use* *case* : Lihat Petunjuk

Skenario :

**Tabel 4.15 Skenario Lihat Petunjuk**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih Petunjuk |  |
|  | 1. Menampilkan halaman petunjuk penggunaan |

1. Skenario *Admin*

Berikut ini adalah skenario dari aktor admin yang telah didefinisikan sebelumnya :

Nama *Use* *case* : Lihat *home admin*

Skenario :

**Tabel 4.16 Skenario Lihat *Home* Admin**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih Menu home admin |  |
|  | 1. Menampilkan halaman home admin |

Nama *Use* *Case* : Lihat Data Penyakit

Skenario :

**Tabel 4.17 Skenario Lihat Penyakit**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar penyakit |  |
|  | 1. Menampilkan halaman penyakit |

Nama *Use* *case* : Tambah data penyakit

Skenario :

**Tabel 4.18 Skenario Tambah Penyakit**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar penyakit |  |
|  | 1. Menampilkan halaman penyakit |
| 1. Melakukan kegiatan input |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses melakukan input |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar penyakit |  |
|  | 1. Menampilkan halaman penyakit |
| 1. Melakukan kegiatan input |  |
|  | 1. Terjadi kesalahan input saat melakukan kegiatan |
| 1. Memperbaiki kesalahan input yang dilakukan |  |
|  | 1. Menyimpan data penyakit |
|  | 1. Menampilkan pesan sukses |

Nama *Use* *case* : Edit penyakit

Skenario :

**Tabel 4.19 Skenario Edit Penyakit**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar penyakit |  |
|  | 1. Menampilkan halaman penyakit |
| 1. Melakukan kegiatan edit |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses melakukan input |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar penyakit |  |
|  | 1. Menampilkan halaman penyakit |
| 1. Melakukan kegiatan edit |  |
|  | 1. Terjadi kesalahan edit saat melakukan kegiatan |
| 1. Memperbaiki kesalahan edit yang dilakukan |  |
|  | 1. Menyimpan data penyakit |
|  | 1. Menampilkan pesan sukses |

Nama *Use* *case* : Hapus penyakit

Skenario :

**Tabel 4.20 Skenario Hapus Penyakit**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar penyakit |  |
|  | 1. Menampilkan halaman penyakit |
| 1. Melakukan kegiatan hapus |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses menghapus |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar penyakit |  |
|  | 1. Menampilkan halaman penyakit |
| 1. Melakukan kegiatan hapus |  |
| 1. Memilih batal menghapus |  |
|  | 1. Menampilkan daftar penyakit |

Nama *Use* *case* : Lihat data gejala

Skenario :

**Tabel 4.21 Skenario Lihat Data Gejala**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar gejala |  |
|  | 1. Menampilkan halaman gejala |

Nama Use case : Tambah data gejala

Skenario:

**Tabel 4.22 Skenario Tambah Data Gejala**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar gejala |  |
|  | 1. Menampilkan halaman gejala |
| 1. Melakukan kegiatan input |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses melakukan input |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar gejala |  |
|  | 1. Menampilkan halaman gejala |
| 1. Melakukan kegiatan input |  |
|  | 1. Terjadi kesalahan input saat melakukan kegiatan |
| 1. Memperbaiki kesalahan input yang dilakukan |  |
|  | 1. Menyimpan data gejala |
|  | 1. Menampilkan pesan sukses |

Nama Use case : Edit Data Gejala

Skenario :

**Tabel 4.23 Skenario Edit Gejala**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar gejala |  |
|  | 1. Menampilkan halaman gejala |
| 1. Melakukan kegiatan edit |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses melakukan input |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar gejala |  |
|  | 1. Menampilkan halaman gejala |
| 1. Melakukan kegiatan edit |  |
|  | 1. Terjadi kesalahan edit saat melakukan kegiatan |
| 1. Memperbaiki kesalahan edit yang dilakukan |  |
|  | 1. Menyimpan data gejala |
|  | 1. Menampilkan pesan sukses |

Nama *Use* *case* : Hapus data gejala

Skenario :

**Tabel 4.24 Skenario Hapus Data Gejala**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar gejala |  |
|  | 1. Menampilkan halaman gejala |
| 1. Melakukan kegiatan hapus |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses menghapus |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar gejala |  |
|  | 1. Menampilkan halaman gejala |
| 1. Melakukan kegiatan hapus |  |
| 1. Memilih batal menghapus |  |
|  | 1. Menampilkan daftar gejala |

Nama *Use* *case* : Lihat *member*

Skenario :

**Tabel 4.25 Skenario Lihat *Member***

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar member |  |
|  | 1. Menampilkan halaman member |

Nama *Use* *case* : Tambah *member*

Skenario :

**Tabel 4.26 Skenario Tambah *Member***

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar member |  |
|  | 1. Menampilkan halaman member |
| 1. Melakukan kegiatan input |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses melakukan input |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar member |  |
|  | 1. Menampilkan halaman member |
| 1. Melakukan kegiatan input |  |
|  | 1. Terjadi kesalahan input saat melakukan kegiatan |
| 1. Memperbaiki kesalahan input yang dilakukan |  |
|  | 1. Menyimpan data member |
|  | 1. Menampilkan pesan sukses |

Nama *Use* *case* : Hapus *member*

Skenario :

**Tabel 4.27 Skenario Hapus *Member***

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar member |  |
|  | 1. Menampilkan halaman member |
| 1. Melakukan kegiatan hapus |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses menghapus |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar member |  |
|  | 1. Menampilkan halaman member |
| 1. Melakukan kegiatan hapus |  |
| 1. Memilih batal menghapus |  |
|  | 1. Menampilkan daftar member |

Nama *Use* *case* : Lihat hasil konsultasi

Skenario :

**Tabel 4.28 Skenario Melihat Hasil Konsultasi**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu hasil konsultasi |  |
|  | 1. Menampilkan halaman hasil konsultasi |
| 1. Melakukan kegiatan cetak |  |
|  | 1. Menampilkan data cetak |
|  | 1. Mencetak hasil konsultasi |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar hasil konsul |  |
|  | 1. Menampilkan halaman hasil konsul |
| 1. Melakukan kegiatan cetak |  |
|  | 1. Gagal menampilkan halaman |
|  | 1. Menampilkan halaman hasil konsultasi |

Nama *Use* *case* : Hapus Hasil Konsultasi

Skenario :

**Tabel 4.29 Skenario Hapus Hasil Konsultasi**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu hasil konsultasi |  |
|  | 1. Menampilkan halaman hasil konsultasi |
| 1. Melakukan kegiatan hapus |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses menghapus |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar hasil konsultasi |  |
|  | 1. Menampilkan halaman halaman hasil konsultasi |
| 1. Melakukan kegiatan hapus |  |
| 1. Memilih batal menghapus |  |
|  | 1. Menampilkan daftar user |

Nama *Use* *case* : Lihat data relasi

Skenario :

**Tabel 4.30 Skenario Lihat Data Relasi**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar relasi |  |
|  | 1. Menampilkan halaman relasi |

Nama *Use* *case* : Tambah data relasi

Skenario :

**Tabel 4.31 Skenario Tambah Data Relasi**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar relasi |  |
|  | 1. Menampilkan halaman relasi |
| 1. Melakukan kegiatan input |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses melakukan input |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar relasi |  |
|  | 1. Menampilkan halaman relasi |
| 1. Melakukan kegiatan input |  |
|  | 1. Terjadi kesalahan input saat melakukan kegiatan |
| 1. Memperbaiki kesalahan input yang dilakukan |  |
|  | 1. Menyimpan data relasi |
|  | 1. Menampilkan pesan sukses |

Nama Use case : Hapus data relasi

Skenario:

**Tabel 4.32 Skenario Hapus Data Relasi**

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** |  |
| 1. Memilih menu daftar relasi |  |
|  | 1. Menampilkan halaman relasi |
| 1. Melakukan kegiatan hapus |  |
|  | 1. Menerima data masukan |
|  | 1. Menampilkan pesan sukses menghapus |
| **Skenario Alternatif** |  |
| 1. Memilih menu daftar relasi |  |
|  | 1. Menampilkan halaman rule |
| 1. Melakukan kegiatan hapus |  |
| 1. Memilih batal menghapus |  |
|  | 1. Menampilkan daftar relasi |

Adapun *use* *case* *diagram* dari sistem yang akan dibuat dapat kita lihat pada Gambar 4.4 berikut :

**Gambar 4.3 *Use Case Diagram***

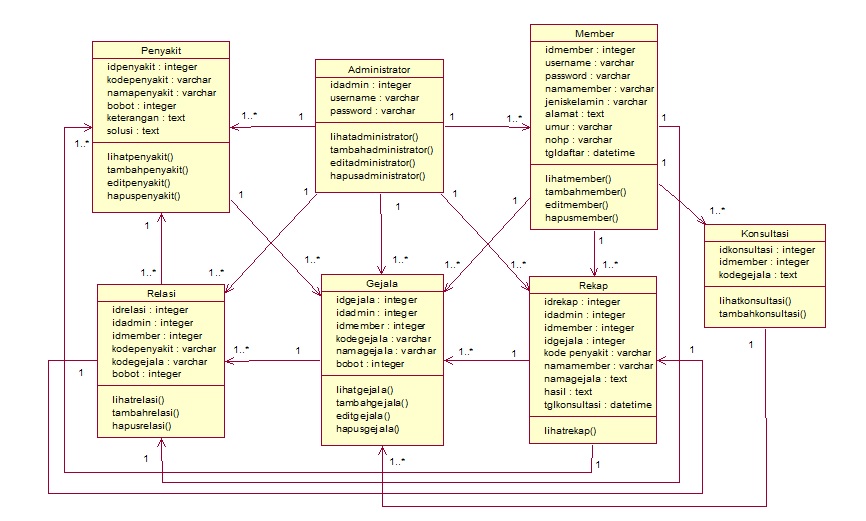
Gambar di atas merupakan *use* *case* *diagram* dari sistem yang di buat, pada *admin* terdapat beberapa *use* *case* diantaranya adalah melihat home, mengelola member, mengelola user, mengelola penyakit, mengelola gejala, dan mengelola konsultasi. *Member* mempunyai *use* *case* diantaranya melihat home, melihat info penyakit, melihat petunjuk, melakukan konsultasi. Sedangkan *non* *member* mempunyai use case melihat home, melihat petunjuk, melihat info penyakit, dan melakukan registrasi.

* + - 1. ***Class Diagram***

*Class Diagram* sering digunakan untuk menjelaskan bagaimana hubungan antara kelas-kelas yang ada di dalam sebuah sistem.

**Tabel 4.33 Definisi *Class Diagram***

|  |  |  |
| --- | --- | --- |
| **No.** | **Class** | **Deskripsi** |
| 1. | *Administrator* | Berisikan data admin, seperti *username* dan *password* untuk melakukan *login*. |
| 2. | *Member* | Berisikan data-data pengunjung member, seperti *username, password*, nama lengkap, alamat, jenis kelamin, umur. |
| 3. | Gejala | Digunakan untuk menyimpan semua data gejala. |
| 4. | Penyakit | Digunakan untuk menyimpan semua data penyakit. |
| 7. | Konsultasi | Berisikan data-data gejala yang di inputkan user. |
| 8. | Hasil Konsultasi | Berisikan data-data hasil konsultasi user. |

Adapun *class diagram* dari sistem yang akan dibuat dapat kita lihat pada Gambar 4.4 berikut :

**Gambar 4.4 Class Diagram**

Gambar di atas merupakan kelas diagram yang berisi relasi antara tabel yang digunakan dalam sistem, di antaranya ada administrator, member, penyakit, gejala, relasi, dan konsultasi.

Berikut adalah keterangan dari isi kelas-kelas yang ada :

1. Tabel *Administrator*

Tabel ini digunakan sebagai tempat penyimpanan data *username* dan *password* admin. Dimana *username* dan *password* nantinya dapat digunakan sebagai login pada sistem.

Berikut bentuk tabel admin yang ada di dalam database dapat lihat pada Tabel 4.34 berikut ini :

**Tabel 4.34 Tabel Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Field** | **Type** | **Keterangan** |
| 1. | idadmin | Int(3) | Primary Key |
| 2. | username | Varchar(15) |  |
| 3. | password | Varchar(15) |  |
| 4. | nama | Varchar (25) |  |

1. Tabel Gejala

Tabel ini digunakan sebagai tempat penyimpanan data gejala dari penyakit.

Berikut bentuk tabel gejala yang ada di dalam database dapat lihat pada Tabel 4.35 berikut ini :

**Tabel 4.35 Tabel Gejala**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Field** | **Type** | **Keterangan** |
| 1. | idgejala | Int(3) | Primary Key |
| 2. | kodegejala | Varchar(4) |  |
| 3. | gejala | Varchar(40) |  |
| 4. | bobot | Int(3) |  |

1. Tabel Konsultasi

Tabel ini digunakan sebagai tempat penyimpanan data gejala yang diinputkan oleh *member*.

Berikut bentuk tabel konsultasi yang ada di dalam database dapat lihat pada Tabel 4.36 berikut ini :

**Tabel 4.36 Tabel Gejala**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Field** | **Type** | **Keterangan** |
| 1. | idkonsultasi | Int(4) | Primary Key |
| 2. | idmember | Int(4) |  |
| 3. | kodegejala | Text |  |

1. Tabel *Member*

Tabel member yaitu tempat penyimpanan data dari member yang mendaftar sebagai member pada sistem.

Berikut bentuk tabel *member* yang ada di dalam database dapat lihat pada Tabel 4.37 berikut ini :

**Tabel 4.37 Tabel *Member***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Field** | **Type** | **Keterangan** |
| 1. | idmember | Int(4) | *Primary Key* |
| 2. | username | Varchar(15) |  |
| 3. | password | Varchar(15) |  |
| 4. | namamember | Varchar(25) |  |
| 5. | alamat | Text |  |
| 6. | umur | Varchar(3) |  |
| 7. | nohp | Varchar(15) |  |
| 8. | tgldaftar | Datetime |  |

1. Tabel Penyakit

Tabel ini digunakan sebagai tempat penyimpanan data dari jenis-jenis penyakit.

Berikut bentuk tabel penyakit yang ada di dalam database dapat lihat pada Tabel 4.38 berikut ini:

**Tabel 4.38 Tabel Penyakit**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Field** | **Type** | **Keterangan** |
| 1. | idpenyakit | Int(4) | Primary Key |
| 2. | kodepenyakit | Varchar(4) |  |
| 3. | namapenyakit | Varchar(20) |  |
| 4. | solusi | text |  |

1. Tabel Rekap

Tabel rekap digunakan untuk menyimpan data-data hasil konsultasi yang dilakukan *member*.

Berikut bentuk tabel rekap yang ada di dalam database dapat lihat pada Tabel 4.39 berikut ini :

**Tabel 4.39 Tabel Rekap**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Field** | **Type** | **Keterangan** |
| 1. | idrekap | Int(4) | *Primary Key* |
| 2 | idmember | Int(4) |  |
| 3 | kodegejala | Text |  |
| 4. | kodepenyakit | Varchar(4) |  |
| 5. | namamember | Varchar(25) |  |
| 6. | hitungan1 | Text |  |
| 7. | hitungan2 | Text |  |
| 8. | hasil | Text |  |
| 9. | tglkonsultasi | Datetime |  |

1. Tabel Relasi

Tabel relasi digunakan untuk pemyimpanan data-data relasi dari proses konsultasi.

Berikut bentuk tabel relasi yang ada di dalam database dapat lihat pada Tabel 4.40 berikut ini :

**Tabel 4.40 Tabel Relasi**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Field** | **Type** | **Keterangan** |
| 1. | idrelasi | Int(4) | *Primary Key* |
| 2 | kodegejala | Varchart(4) |  |
| 3 | kodepenyakit | Varchar(4) |  |
| 4. | bobot | Int(3) |  |

* + - 1. ***Sequence Diagram***

*Sequence* *diagram* menggambarkan kegiatan aktor untuk melakukan kegiatan pada sistem yang dilakukan secara teratur. Kegiatan yang digambarkan adalah proses setiap kegiatan yang dilakukan oleh aktor. Berikut gambar-gambar *sequence* *diagram* yang terdapat pada sistem, diantaranya :

1. *Sequence diagram* untuk login *admin*

*Sequence diagram* login *admin* dapat dilihat pada gambar berikut ini :

**Gambar 4.5 *Sequence Diagram login admin***

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan login ke dalam sistem dengan mengisi form login terlebih dahulu.

1. *Sequence diagram* untuk *login member*

*Sequence diagram* login *member* dapat dilihat pada gambar berikut ini :

**Gambar 4.6 *Sequence Diagram login member***

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *member* dalam melakukan login ke dalam sistem dengan mengisi form login terlebih dahulu.

1. *Sequence diagram* registrasi

Menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *non* *member* dalam melakukan registrasi.

**Gambar 4.7 *Sequence Diagram registrasi***

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *non member* dalam melakukan registrasi untuk mendaftarkan diri sebagai *member.*

1. *Sequence diagram* konsultasi

Menggambarkan atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *member* dalam melakukan dan melihat konsultasi. Sequence diagram dapat dilihat pada gambar berikut ini :

**Gambar 4.8 *Sequence Diagram* konsultasi**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *member* dalam melakukan konsultasi untuk mendapatkan hasil diagnosa dari sistem.

1. *Sequence diagram* kelola penyakit

Menggambarkan interaksi dengan sistem yang dilakukan oleh aktor admin dalam melakukan pengelolaan data penyakit. *Sequence diagram* dapat dilihat pada gambar berikut ini :



**Gambar 4.9 *Sequence Diagram* kelola penyakit**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor admin dalam melakukan pengelolaan penyakit yang ada.

1. *Sequence diagram* kelola gejala

Menggambarkan interaksi aktor dengan sistem yang dilakukan oleh aktor *admin* dalam melakukan pengelolaan data gejala. *Sequence* *diagram* dapat dilihat pada gambar berikut ini :



**Gambar 4.10 *Sequence Diagram* kelola gejala**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan kegiatan mengelola gejala yang ada, seperti menambah, edit, dan hapus gejala.

1. *Sequence diagram* kelola member

Menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan pengelolaan data *member*. *Sequence* *diagram* dapat dilihat pada gambar berikut ini :



**Gambar 4.11 *Sequence Diagram* Kelola *Member***

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan pengelolaan *member*.

1. *Sequence diagram* kelola rule

Menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan pengelolaan data rule. *Sequence* *diagram* dapat dilihat pada gambar berikut ini :



**Gambar 4.12 *Sequence Diagram* Kelola *Rule***

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan kegiatan mengelola *rule*, seperti melihat, menambah, dan menghapus *rule*.

1. *Sequence diagram* hasil konsultasi

Menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *member*  dalam melakukan pengelolaan konsultasi. *Sequence diagram* dapat dilihat pada gambar berikut ini :

**Gambar 4.13 *Sequence Diagram* Kelola** **Hasil Konsultasi**

* + - 1. ***Collaboration Diagram***

*Collaboration* *diagram* adalah bentuk lain dari *sequence* *diagram*. *Collaboration* *diagram* menggambarkan interaksi antar objek atau bagian dalam bentuk urutan pengiriman pesan.

Berikut gambar-gambar *collaboration* *diagram* yang terdapat dalam sistem adalah :

1. *Collaboration diagram* login *admin*

Diagram ini menjelaskan urutan langkah-langkah yang dilakukan admin untuk login ke sistem. Diagram kolaborasi login admin dapat dilihat pada gambar berikut :



**Gambar 4.14 *Collaboration* *Diagram* Login Admin**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor admin dalam melakukan login ke dalam sistem dengan mengisi form login terlebih dahulu.

1. *Collaboration diagram* login *member*

Diagram ini menjelaskan urutan langkah-langkah yang dilakukan *member* untuk login ke sistem. Diagram kolaborasi login *member* dapat dilihat pada gambar berikut :



**Gambar 4.15 *Collaboration* *Diagram* Login *Member***

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *member* dalam melakukan login ke dalam sistem dengan mengisi form login terlebih dahulu.

1. *Collaboration diagram* registrasi

*Collaboration diagram* ini menjelaskan urutan langkah-langkah yang dilakukan *non* *member* untuk melakukan registrasi dalam sistem. *Collaboration diagram* registrasi dapat dilihat pada gambar berikut :

**Gambar 4.16 *Collaboration Diagram* Registrasi**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *non* *member* dalam melakukan registrasi untuk mendaftarkan diri sebagai *member*.

1. *Collaboration diagram* konsultasi

Diagram ini menjelaskan urutan langkah-langkah yang dilakukan *member* untuk melakukan konsultasi pada sistem. Diagram kolaborasi konsultasi dapat dilihat pada gambar berikut :



**Gambar 4.17 *Collaboration* *Diagram* Konsultasi**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor member dalam melakukan kegiatan konsultasi, seperti memasukkan konsultasi

1. *Collaboration diagram* kelola penyakit

Pada diagram kolaborasi kelola penyakit, diagram ini akan menjelaskan urutan langkah-langkah yang dilakukan aktor yaitu admin untuk mengelola data penyakit. Diagram kolaborasi kelola penyakit dapat dilihat pada gambar berikut :



**Gambar 4.18 *Collaboration* *Diagram* Kelola Penyakit**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* pada sistem dalam mengelola penyakit seperti melihat, menambah, mengedit, menghapus penyakit.

1. *Collaboration diagram* kelola gejala

Pada diagram kolaborasi kelola gejala, diagram ini akan menjelaskan urutan langkah-langkah yang dilakukan aktor yaitu *admin* untuk mengelola data gejala, seperti melihat, menambah, mengubah, dan menghapus data gejala yang ada. Diagram kolaborasi kelola gejala dapat dilihat pada gambar berikut:

**Gambar 4.19 *Collaboration* *Diagram* Kelola Gejala**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan kegiatan mengelola gejala yang ada, seperti menambah, edit, dan hapus gejala.

1. *Collaboration diagram* kelola member

Pada diagram kolaborasi kelola *member* akan menjelaskan urutan langkah-langkah yang dilakukan aktor yaitu *admin* untuk mengelola data *member*. Diagram kolaborasi kelola *member* dapat dilihat pada gambar berikut :



**Gambar 4.20 *Collaboration* *Diagram* Kelola *Member***

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan kegiatan mengelola *member*, seperti melihat, menambah dan menghapus member.

1. *Collaboration diagram* kelola rule

Pada diagram kolaborasi kelola rule, diagram ini akan menjelaskan urutan langkah-langkah yang dilakukan aktor yaitu *admin* untuk mengelola data rule pada sistem. Diagram kolaborasi kelola data rule dapat dilihat pada gambar berikut :

**Gambar 4.21 *Collaboration* *Diagram* Kelola Rule**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor *admin* dalam melakukan kegiatan mengelola rule seperti melihat, menambah, dan menghapus *rule*.

1. *Collaboration diagram* hasil konsultasi

Pada diagram kolaborasi hasil konsultasi akan menjelaskan urutan langkah-langkah yang dilakukan aktor yaitu *admin* untuk melihat data hasil konsultasi. Diagram kolaborasi lihat hasil konsultasi dapat dilihat pada gambar berikut :

**Gambar 4.22 *Collaboration* *Diagram* Hasil Konsultasi**

Gambar di atas menggambarkan interaksi aktor pada sistem yang dilakukan oleh aktor member dalam melakukan kegiatan konsultasi, seperti memasukkan konsultasi



* + - 1. ***State Chart Diagram***

*State Chart Diagram* menggambarkan transisi dan perubahan status yang terjadi pada sistem. Pada umumnya *state chart diagram* mengambarkan alur interaksi pengguna dengan sistem.

1. *State chart diagram* registrasi *non member*

Diagram ini menjelaskan proses penambahan *member* yang dilakukan oleh *non member* kemudian disimpan dalam *database*.



**Gambar 4.23 *Statechart Diagram* Registrasi**

Gambar diatas menggambarkan diagram *state* *chart* untuk *admin* dalam melakukan penambahan rule seperti kode penyakit,kode gejala,bobot.

1. *State chart diagram* konsultasi member

Diagram ini menjelaskan proses konsultasi yang dilakukan oleh *member* pada sistem.



**Gambar 4.24 *Statechart diagram* konsultasi *member***

Gambar diatas menggambarkan diagram *state* *chart* untuk *member* dalam melakukan konsultasi dengan login mengisi *username* dan *password* kemudian melakukan konsultasi.

1. *State chart diagram* penyakit

*State* *chart* *diagram* menggambarkan tentang kelakuan sistem yang akan dibangun. *State* *chart* *diagram* *admin* akan mengelola penyakit. Berikut *state* *chart* *diagram* pada kelola penyakit :

**Gambar 4.25 *Statechart diagram* kelola penyakit**

Gambar diatas menggambarkan diagram *state* *chart* untuk *admin* dalam melakukan proses penambahan gejala penyakit dengan mengisi kode penyakit, nama penyakit, keterangan, solusi.

1. *State chart diagram* gejala

Diagram ini menjelaskan proses proses penambahan penyakit oleh *admin* di sistem. Diagram *state chart* gejala dapat dilihat pada gambar berikut :



**Gambar 4.26 *Statechart diagram* kelola gejala**

Gambar diatas menggambarkan diagram state chart untuk *admin* dalam menambahkan gejala seperti kode gejala dan nama gejala.

1. *State chart diagram* rule

Diagram ini menjelaskan proses proses penambahan rule oleh *admin* di sistem. Diagram state gejala dapat dilihat pada gambar berikut :

**Gambar 4.27 *Statechart Diagram* Kelola Rule**

Gambar diatas menggambarkan diagram *state* *chart* untuk *admin* dalam melakukan penambahan rule seperti kode penyakit, kode gejala, bobot.

* + - 1. ***Activity Diagram***

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

1. *Activity diagram* untuk *non member*

*Activity Diagram non member* menggambarkan segala aktivitas yang bisa dilakukan oleh non *member* terhadap sistem. Diagram aktivitas member dapat dilihat pada gambar berikut :

**Gambar 4.28 *Activity Diagram Non* *Member***

1. *Activity diagram* untuk *member*

Activity Diagram member menggambarkan segala aktivitas yang bisa dilakukan oleh member terhadap sistem. Diagram aktivitas member dapat dilihat pada gambar berikut :

**Gambar 4.29 *Activity Diagram Member***

1. *Activity diagram* untuk *admin*

*Activity* *Diagram* untuk *admin* menggambarkan segala aktivitas yang bisa dilakukan oleh *admin* terhadap sistem. Diagram aktivitas *admin* dapat dilihat pada gambar berikut :

**Gambar 4.30 *Activity Diagram Admin***

* + - 1. ***Deployment Diagram***

*Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. . Pada perancangan sistem ini menggunakan *web* *server* (*Apache*) dan *database* *server* (*MySQL*). sedangkan *user* dapat melakukan akses dengan menggunakan web browser seperti *Mozilla firefox*, dan *Google Chrome*.



**Gambar 4.31 *Deployment Diagram***

* + 1. **Perancangan *Interface***

Perancangan interface adalah tahapan dalam merancang tampilan dari pembuatan website sistem pakar ini. Perancangan ini mengacu kepada pembuatan sistem yang sudah di rencanakan berdasarkan alur dari perancangan model sebelumnya. Langkah-langkah dalam perancangan desain program sistem pakar *pneumonia* pada anak antara lain :

1. Tampilan *Non Member*
2. Tampilan *Home*

Pada tampilan ini berisikan halaman *home* awal yang berisi sekilas tentang sistem pakar. Tampilan *home* dapat dilihat pada gambar berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| **Home** | Info  Penyakit | Petunjuk | Registrasi Member | LoginMember |
| Home | | | | |
|  | | | | |

**Gambar 4.32 Tampilan Home**

1. Tampilan Info Penyakit *Pneumonia*

Pada tampilan ini berisikan halaman yang berisi keterangan tentang klasifikasi *pneumonia* pada anak. Tampilan info penyakit dapat dilihat pada gambar berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| Beranda | **Info**  **Penyakit** | Petunjuk | Registrasi Member | LoginMember |
| Informasi Penyakit | | | | |
|  | | | | |

**Gambar 4.33 Tampilan Info Penyakit**

1. Tampilan Petunjuk

Pada tampilan ini berisikan halaman yang berisi petunjuk penggunaan sistem pakar. Tampilan petunjuk dapat dilihat pada gambar berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| Beranda | Info  Penyakit | **Petunjuk** | RegistrasiMember | LoginMember |
| Petunjuk | | | | |
|  | | | | |

**Gambar 4.34 Tampilan Petunjuk**

1. Tampilan Registrasi

Pada tampilan ini berisikan halaman yang berisi petunjuk penggunaan sistem pakar. Tampilan registrasi dapat dilihat pada gambar berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| Beranda | Tentang  Penyakit | Cara Pemakaian | **Registrasi Konsultasi** | LoginMember |
| **xxxxxx**  Username :  **xxxxxx**  Password :    **xxxxxx**  Nama :  **xxxxxx**  Umur :  Jenis Kelamin :  **xxxxxx**    **xxxxxx**  Alamat :  No. HP :  **DAFTAR**  **xxxxxx** | | | | |

**Gambar 4.35 Tampilan Registrasi**

1. Tampilan Login

Pada tampilan ini berisikan halaman yang berisi halaman untuk melakukan login. Tampilan login dapat dilihat pada gambar berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| Beranda | Tentang  Penyakit | Cara Pemakaian | Registrasi Konsultasi | **Login Member** |
| **xxxxxx**  Username :  **xxxxxx**  Password :  Belum daftar? Silahkan Register  **LOGIN** | | | | |

**Gambar 4.36 Tampilan Login**

1. Tampilan *Member*
2. Tampilan *Home*

Pada tampilan ini berisikan halaman *home* awal yang berisi sekilas tentang sistem pakar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| **Home** | Tentang  Penyakit | Cara Pemakaian | Konsultasi | Logout |
| Sistem Pakar | | | | |
|  | | | | |

**Gambar 4.37 Tampilan *Home***

1. Tampilan Info Penyakit *Pneumonia*

Pada tampilan ini berisikan halaman yang berisi keterangan tentang penyakit *pneumonia* pada anak

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| Beranda | **Info**  **Penyakit** | Cara Pemakaian | Konsultasi | Logout |
| Informasi Penyakit Pneumonia | | | | |
|  | | | | |

**Gambar 4.38 Tampilan Info Penyakit**

1. Tampilan Petunjuk

Pada tampilan ini berisikan halaman yang berisi cara pemakaian sistem pakar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar Pneumonia Pada Anak | | | | |
| Beranda | Tentang  Penyakit | **Petunjuk** | Konsultasi | Logout |
| Petunjuk | | | | |
|  | | | | |

**Gambar 4.39 Tampilan Petunjuk**

1. Tampilan Konsultasi

Pada tampilan ini berisikan halaman yang berisi tampilan konsultasi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sistem Pakar *Pneumonia* Pada Anak | | | | |
| Beranda | Tentang  Penyakit | Cara Pemakaian | **Konsultasi** | Logout |
| Konsultasi | | | | |
| |  |  | | --- | --- | | Gejala | Pilih Gejala | |  |  | |  |  | |  |  | | | | | |

**Gambar 4.40 Tampilan Konsultasi**

1. Tampilan *Admin*
2. Tampilan *Home*

Pada tampilan ini berisikan halaman home awal yang berisi sekilas tentang tulisan selamat datang untuk admin.

**Home**

Daftar Member

**Registrasi**

**Login**

**Bantuan Penggunaan**

Selamat Datang Administrator

Sistem Pakar Pneumonia Pada Anak

Daftar Gejala

Daftar Penyakit

Daftar Rule

Konsultasi

**Gambar 4.41 Tampilan Home *Admin***

1. Tampilan Data Member

Pada tampilan ini berisikan halaman yang berisi tampilan untuk mengelola *member* yang ada.

Home

**Daftar Member**

Sistem Pakar Pneumonia Pada Anak

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nama | JK | alamat | umur | username | password |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Daftar Gejala

Daftar Penyakit

Daftar Rule

Konsultasi

**Gambar 4.42 Tampilan Data *Member***

**Pesan**

1. Tampilan Data Gejala

Pada tampilan ini berisikan halaman yang berisi tampilan untuk mengelola gejala *pneumonia* pada anak.

Home

DaftarMember

**Registrasi**

**Login**

**Bantuan Penggunaan**

Sistem Pakar Pneumonia Pada Anak

**Daftar Gejala**

Daftar Penyakit

Daftar Rule

Konsultasi

Kode Gejala Nama Gejala Action

NO

**Gambar 4.43 Tampilan Data Gejala**

1. Tampilan Data Penyakit

Pada tampilan ini berisikan halaman yang berisi tampilan untuk mengelola klasifikasi *pneumonia* pada anak.

Home

Daftar Member

**Registrasi**

**Login**

**Bantuan Penggunaan**

Sistem Pakar Pneumonia Pada Anak

Daftar Gejala

**Daftar Penyakit**

Daftar Rule

Konsultasi

Kode Penyakit Nama Penyakit Keterangan Action

NO

**Gambar 4.44 Tampilan Data Penyakit**

1. Tampilan Data Rule

Pada tampilan ini berisikan halaman yang berisi tampilan untuk mengelola rule penyakit *pneumonia* pada anak.

Home

DaftarMember

**Registrasi**

**Login**

**Bantuan Penggunaan**

Sistem Pakar Pneumonia Pada Anak

Daftar Gejala

Daftar Penyakit

**Daftar Rule**

Konsultasi

Kode Gejala Kode Penyakit Action

NO

**Gambar 4.45 Tampilan Data Rule**

1. Tampilan Data Konsultasi

Pada tampilan ini berisikan halaman yang berisi tampilan untuk melihat hasil konsultasi yang dilakukan *user*.

Home

DaftarMember

**Registrasi**

**Login**

**Bantuan Penggunaan**

Sistem Pakar Pneumonia Pada Anak

Daftar Gejala

Daftar Penyakit

Daftar Rule

**Konsultasi**

Hasil diagnosa Gejala tingkat keyakinan waktu konsultasi

NO

**Gambar 4.46 Tampilan Konsultasi**

**BAB V**

**IMPLEMENTASI DAN PENGUJIAN**

1. **Implementasi sistem**

Implementasi sistem merupakan hasil dari analisa dan perancangan ke dalam sistem, sehingga akan diketahui apakah sistem yang dibuat telah menghasilkan tujuan yang diinginkan. Implementasi ini dilakukan bertujuan untuk dapat bermanfaat bagi *user* dan dapat digunakan dengan baik.

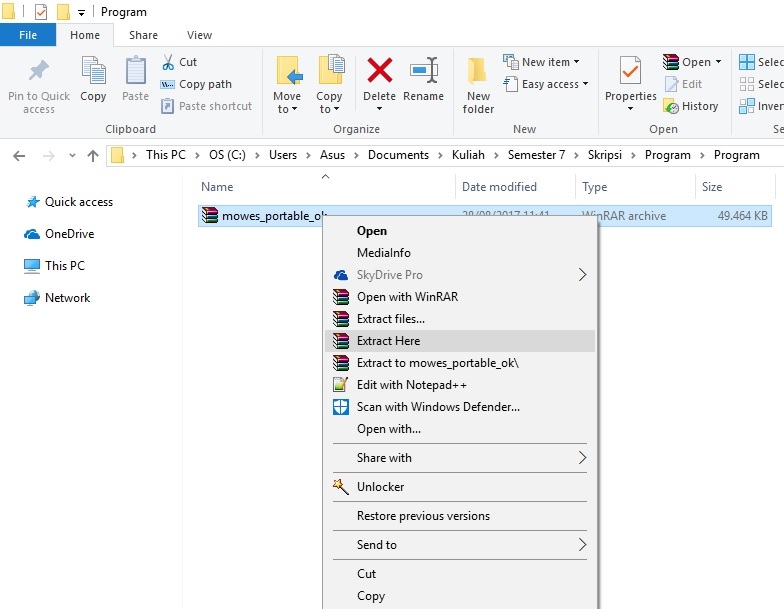
* + 1. **Analisa Kebutuhan *Hardware***

Dalam membangun sistem ini dibutuhkan perangkat keras untuk mendukungnya implementasi pembuatan sistem ini adapun perangkat keras yang dibutuhkan sebagai berikut :

1. Laptop ASUS A456U
2. *Processor* Intel Core i5-7200U CPU 2.7 GHz
3. Memory 8 GB
4. Hardisk 1000 GB
5. Flashdisk kapasitas 16 GB
6. Serta *hardware* pendukung lainnya.
   * 1. **Analisa Kebutuhan *Software***

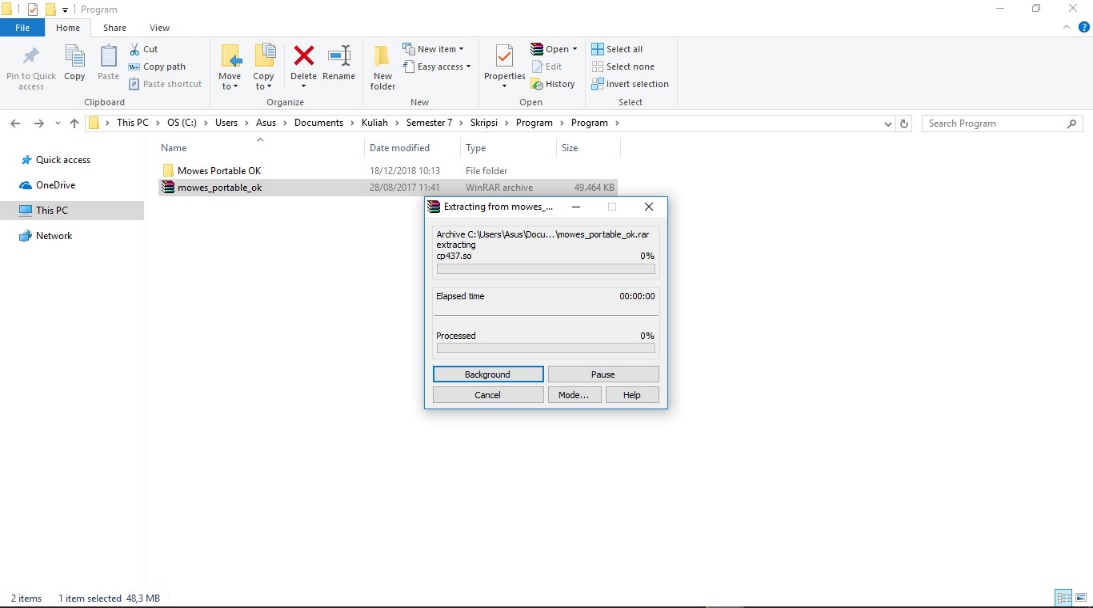
Selain perangkat keras, perangkat lunak sangat dibutuhkan dalam menunjang pembuatan sistem pakar ini. Perangkat lunak yang digunakan diantaranya :

1. Sistem Operasi Windows 10 Home
2. *Microsoft Word* 2013
3. *Rational Rose Enterprise* V7
4. Mowes, Database MySQL
5. Adobe Dreamweaver CS3
6. *Mozilla Firefox*
7. Serta software pendukung lainnya.
8. **Instalasi Mowes**
9. Klik File mowes\_portable\_ok kemudian extract here seperti gambar berikut :



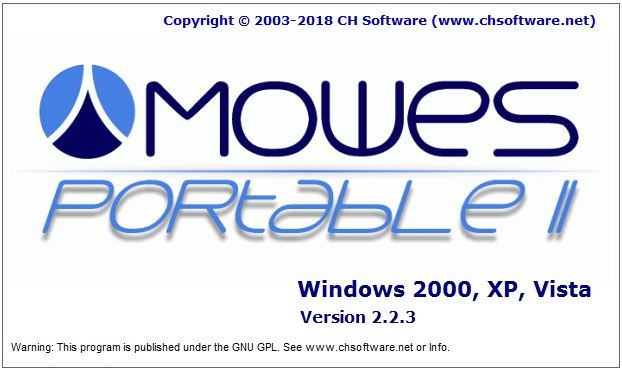
**Gambar 5.1 *Extract* aplikasi Mowes Portable II**

1. Selanjutnya tunggu proses extract file mowes tersebut selesai

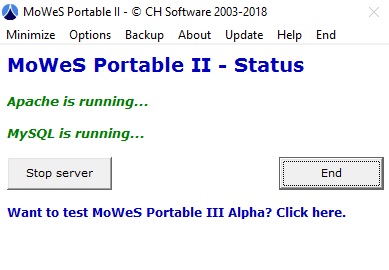


**Gambar 5.2 Proses *Extract* aplikasi Mowes Portable II**

1. Setelah proses extract selesai lalu buka aplikasi mowes



**Gambar 5.3 Tampilanaplikasi Mowes Portable II**

1. Aplikasi mowes sudah aktif dan sudah bisa digunakan

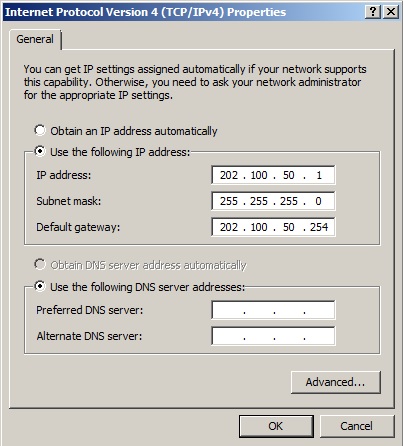
**Gambar 5.4 Tampilan Mowes Portable II saat dijalankan.**

1. **Pengujian**
   * 1. **Pengujian *Local***

Pada tahap pengujian ini, aplikasi sistem pakar ini juga dapat dijalankan secara LAN (*Local Area Network),* yaitu dengan menyambungkan sebuah *device* (komputer) dengan *device* (komputer) lainnya sehingga dapat saling berkomunikasi. Pada *client-side* dapat mengakses data pada komputer *server*, sedangkan pada *server-side* menyediakan informasi yang dibutuhkan oleh *client-side.*

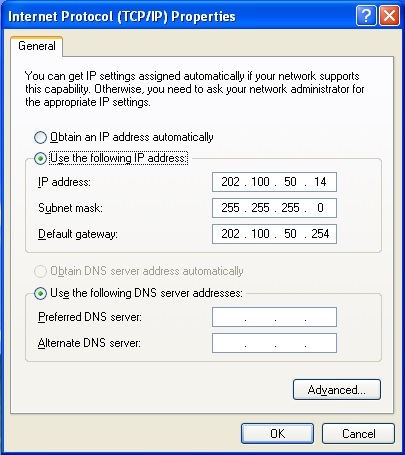
Penerapan dari konsep jaringan LAN (*Local Area Network)* yang terdiri dari minimal 3 buah *device (*komputer) yaitu sebagai berikut :

1. Dalam pengujian ini menggunakan kabel *straight* untuk menghubungkan sebuah *device* (komputer) dengan *switch*.
2. Agar semua komputer saling terhubung, maka pada masing-masing komputer dengan cara pilih *menu Start* lalupilih *Control Panel,* lalu *Network and Internet* kemudian *Network Connections* pada kedua *device* (komputer).
3. Klik kanan pada *Local Area Network* lalu pilih *Properties*.
4. Pada tab bar *“Networking”* pilih *“Internet Protocol Version 4 (TCP/IPv4)*.
5. Kemudian lakukan setting pada masing – masing komputer seperti gambar di bawah ini :

****

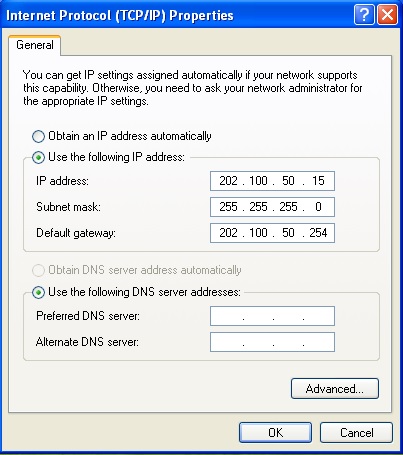
**Gambar 5.5 Konfigurasi IP Address Komputer Server**

Gambar di atas menampilkan konfigurasi yang dilakukan pada komputer sebagai server.

****

**Gambar 5.6 Konfigurasi IP Address Komputer Client 1**

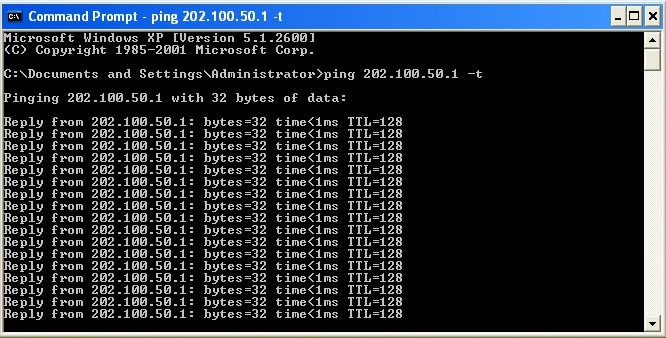
Gambar di atas menampilkan konfigurasi yang dilakukan pada komputer sebagai client 1.

****

**Gambar 5.7 Konfigurasi IP Address Komputer Client 2**

Gambar di atas menampilkan konfigurasi yang dilakukan pada komputer sebagai client 1.

1. Kemudian lakukan tes koneksi dari client ke server.

****

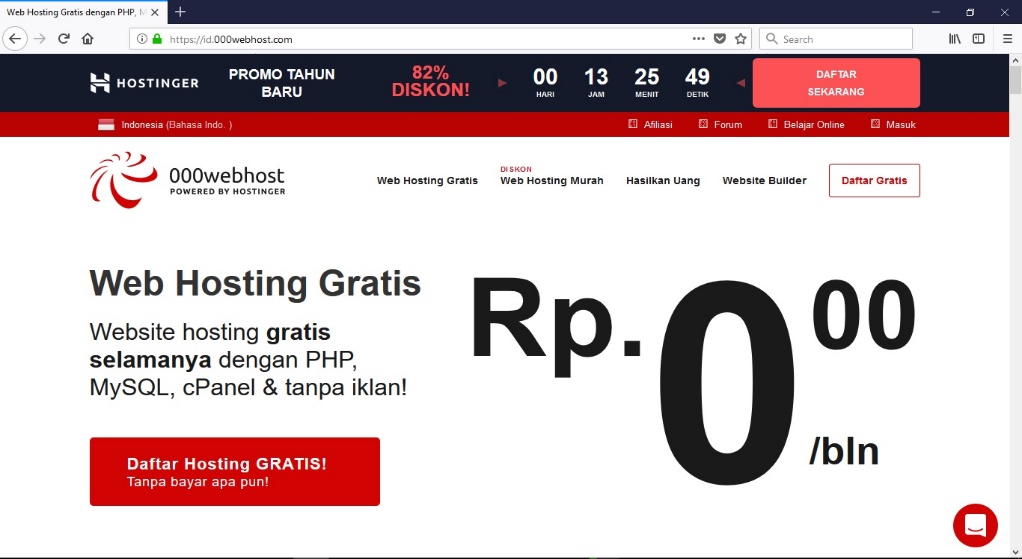
**Gambar 5.8 Proses Ping dari Client ke Server**

Gambar di atas menampilkan hasil tes koneksi dari dari client ke server.

* + 1. **Pengujian *Online***

Pada tahap pengujian ini akan dilakukan *hosting* *website* atau mendaftarkan *website* yang telah dibangun ke internet agar *website* bisa diakses oleh *user* secara *online* melalui *web browser*. Berikut ini tahap-tahap dalam melakukan *hosting* :

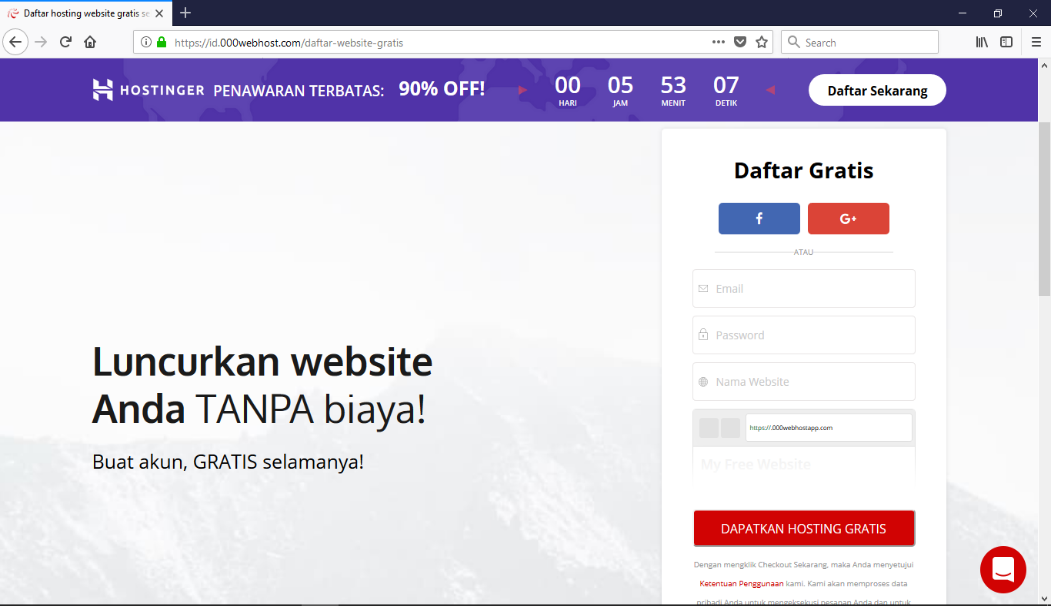
1. Langkah pertama yang dilakukan adalah dengan mengakses situs hosting yakni <https://id.000webhost.com/>

****

**Gambar 5.9 Langkah Pertama Proses Hosting**

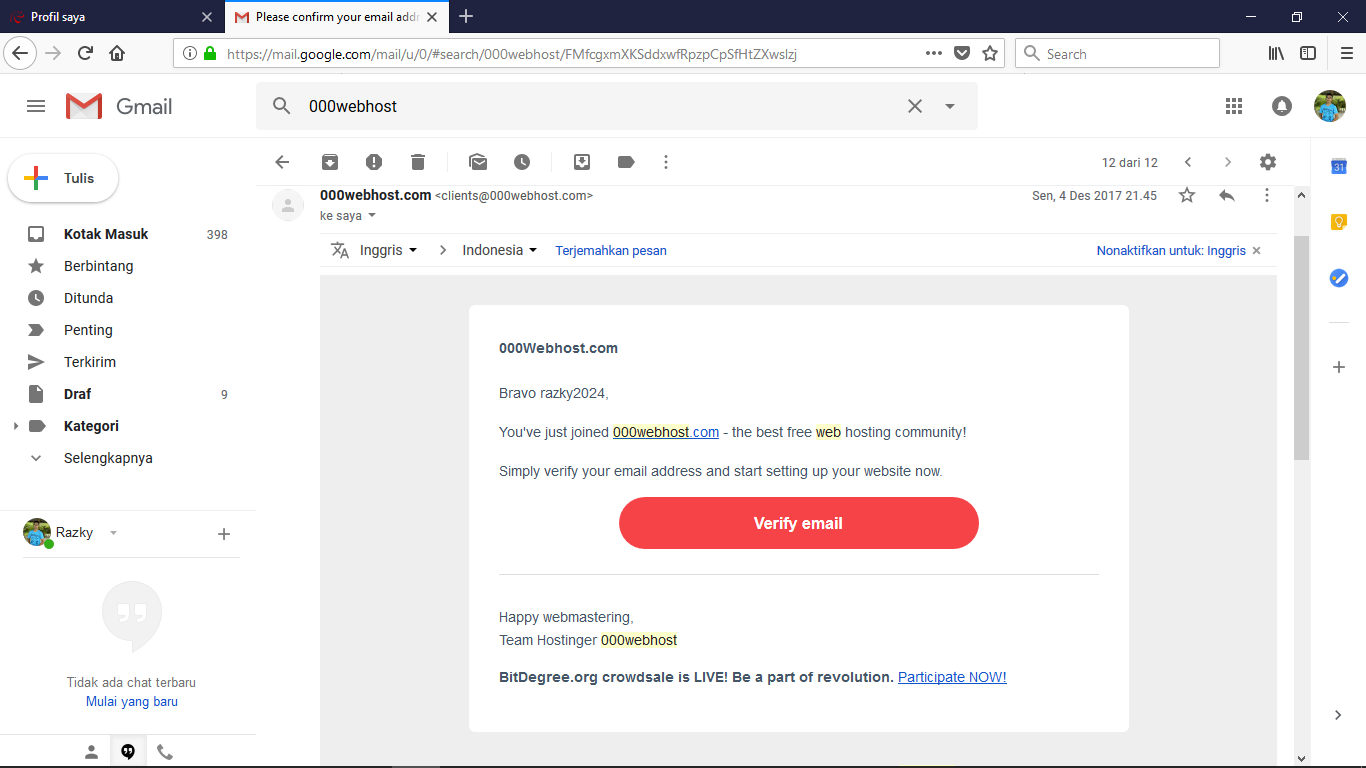
Gambar di atas menunjukan tahap pertama saat mengakses halaman web tempat akan di *hosting*.

1. Langkah kedua klik tombol “daftar gratis” bewarna merah untuk mendaftarkan diri di *website* tersebut



**Gambar 5.10 Langkah Kedua Proses Hosting**

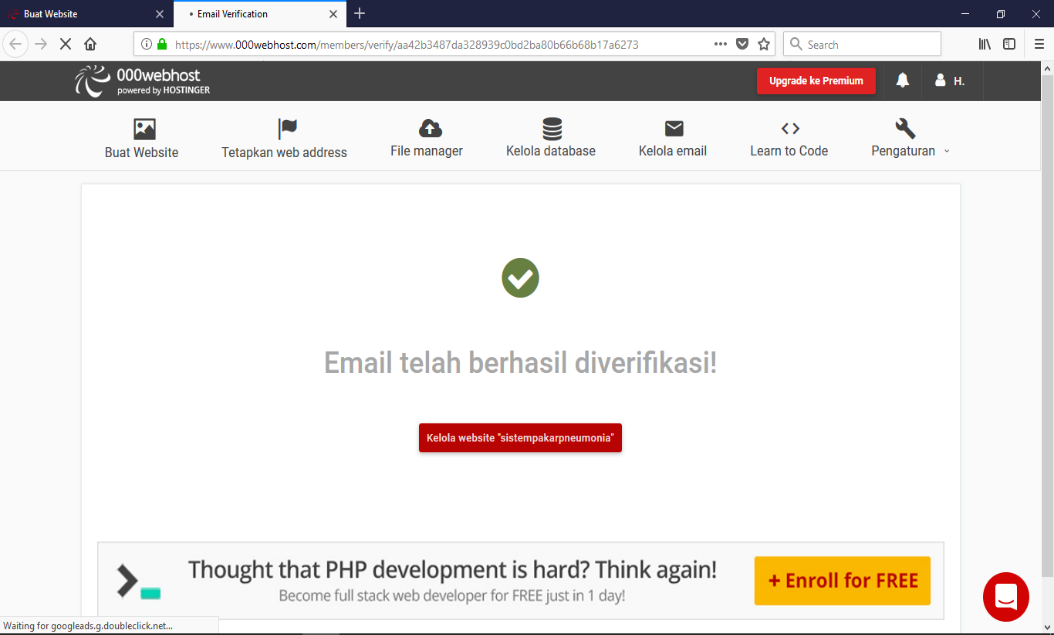
Gambar di atas menampilkan halaman untuk melakukan pendaftaran akun pada *website* tersebut.

1. Langkah ketiga, loginlah ke email dan periksa pesan masuk dari 000webhost yang berisi verifikasi email pada webhost

**Gambar 5.11 Langkah Ketiga Proses Hosting**

Gambar di atas menampilkan email konfirmasi yang dikirim oleh *website* *hosting* tntuk verifikasi email.

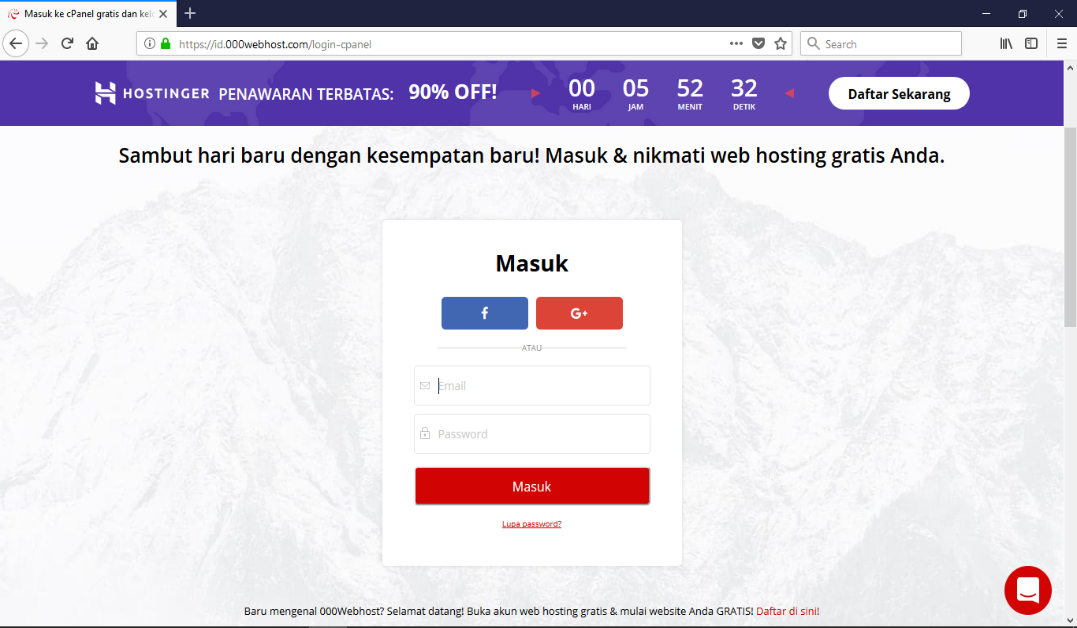
1. Langkah keempat, verifikasi akun anda untuk mengaktifkan akun 000webhost



**Gambar 5.12 Langkah Keempat Proses Hosting**

Gambar di atas menampilkan email telah berhasil diverifikasi.

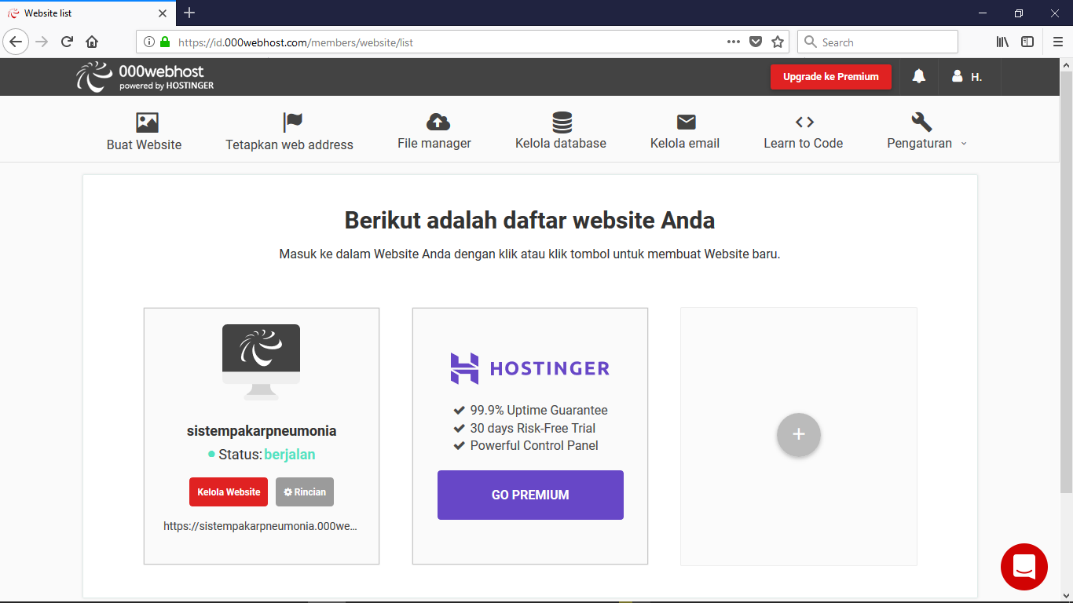
1. Langkah kelima, kembali lagi ke webhost dan melakukan login kedalam akun 000webhost



**Gambar 5.13 Langkah Kelima Proses Hosting**

Gambar di atas menampilkan halaman login ke dalam akun yang telah didaftarkan.

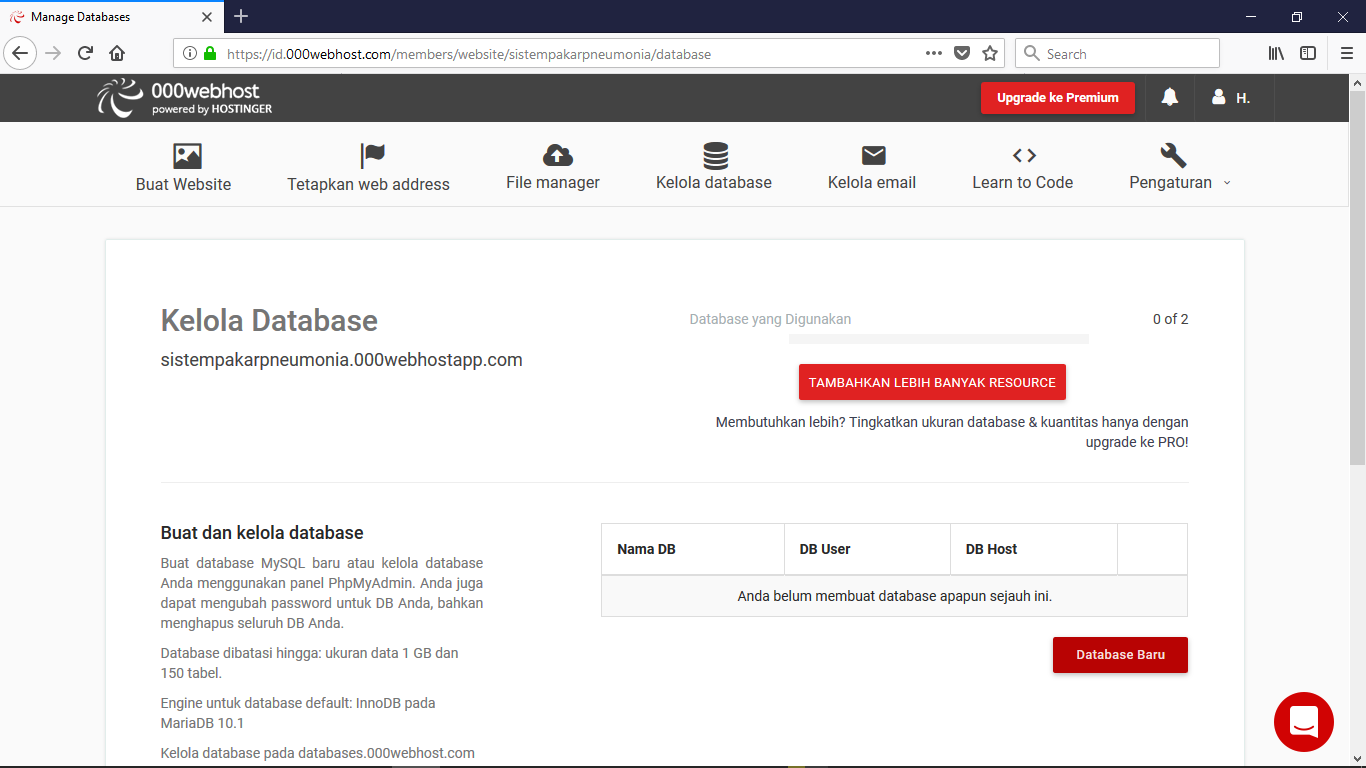
1. Langkah keenam, pilih *website* yang akan di kelola pada beranda



**Gambar 5.14 Langkah Keenam Proses Hosting**

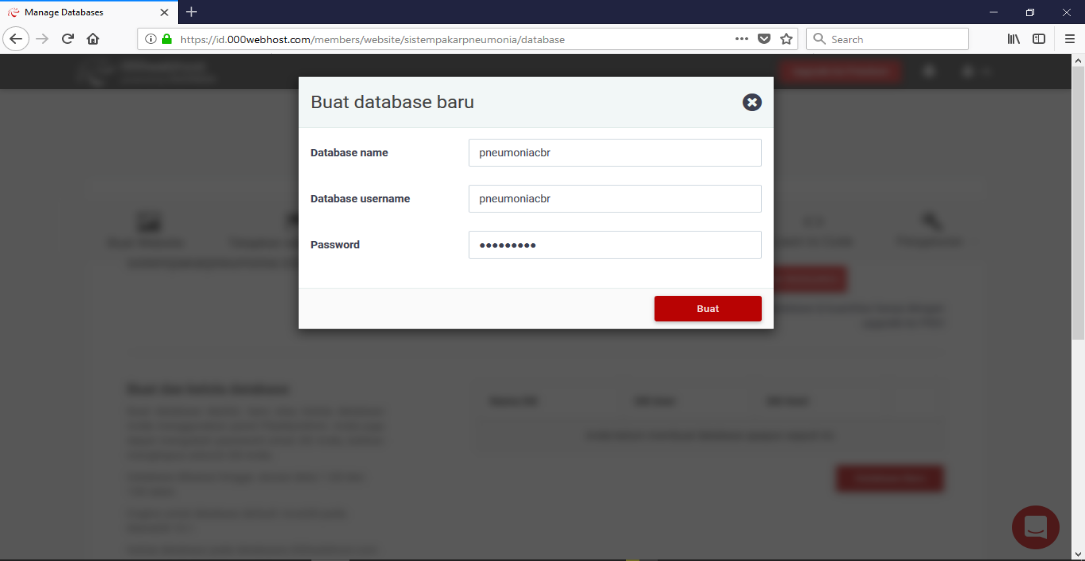
Gambar di atas menampilkan halaman untuk melakukan pengelolaan website dengan klik “Kelola Website”.

1. Langkah ketujuh, mengelola database yang akan digunakan pada *website* yang akan di *hosting*



**Gambar 5.15 Langkah Ketujuh Proses Hosting**

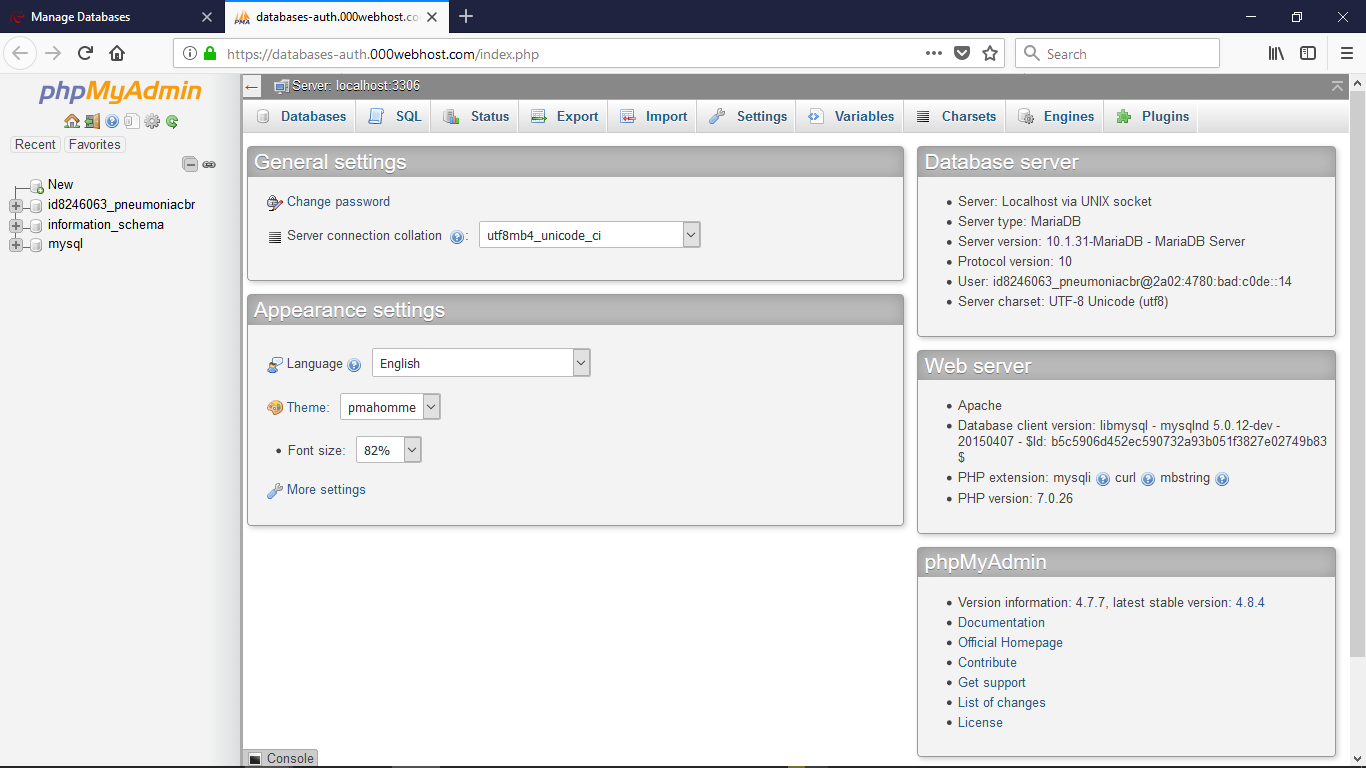
Gambar di atas menampilkan halaman untuk pengelolaan database

1. Langkah kedelapan, klik tombol database baru dan lakukan pebuatan database

**Gambar 5.16 Langkah Kedelapan Proses Hosting**

Gambar di atas menampilkan halaman untuk pembuatan database baru dengan mengisi nama database, database username dan password. Kemudian klik “buat”.

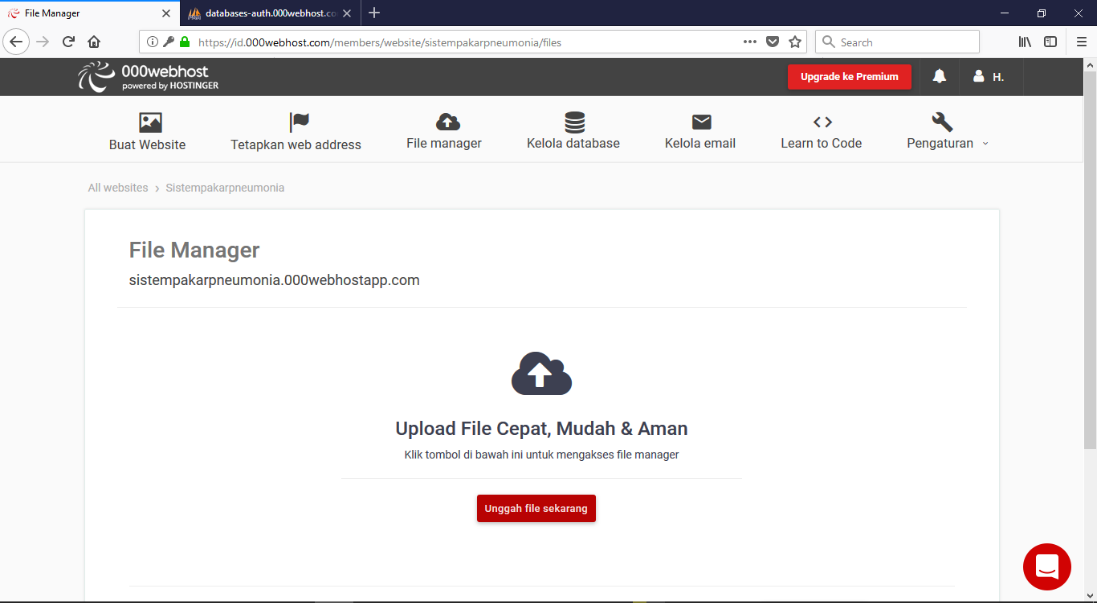
1. Langkah kesembilan, setelah database berhasil dibuat, masukan database yang akan menjadi database *web* dengan mengklik menu import pada halaman kelola database, lalu masukan file yang akan di *upload* kedalam database.



**Gambar 5.17 Langkah Kesembilan Proses Hosting**

Gambar di atas merupakan tahapan untuk memasukan database ke dalam *website*.

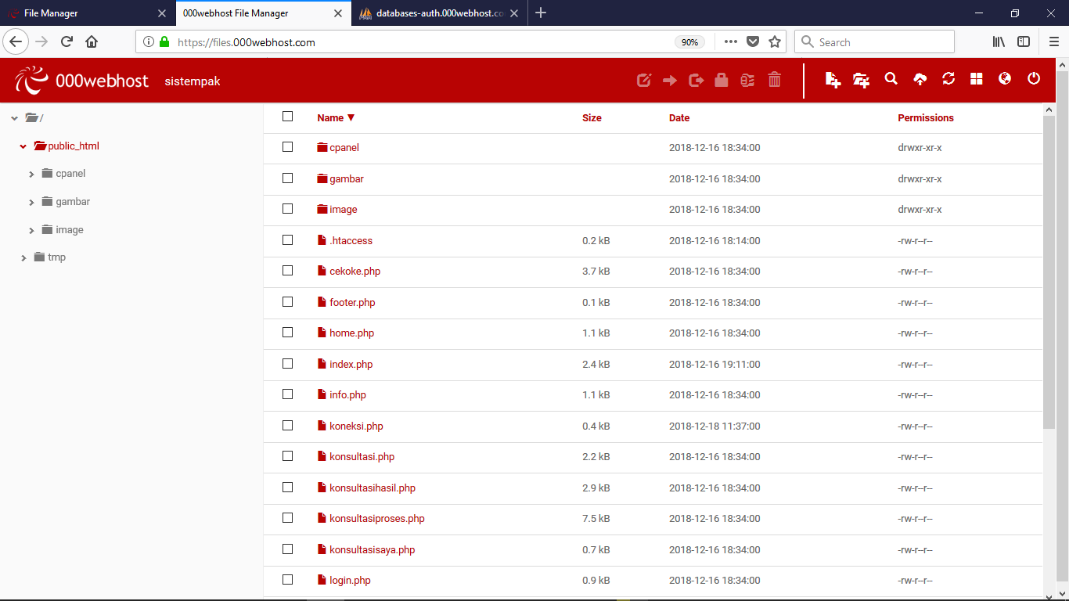
1. Langkah kesepuluh, tekan url yang ada di beranda website yang akan mengarahkan kepada website yang sudah dibuat.



**Gambar 5.18 Langkah Kesepuluh Proses Hosting**

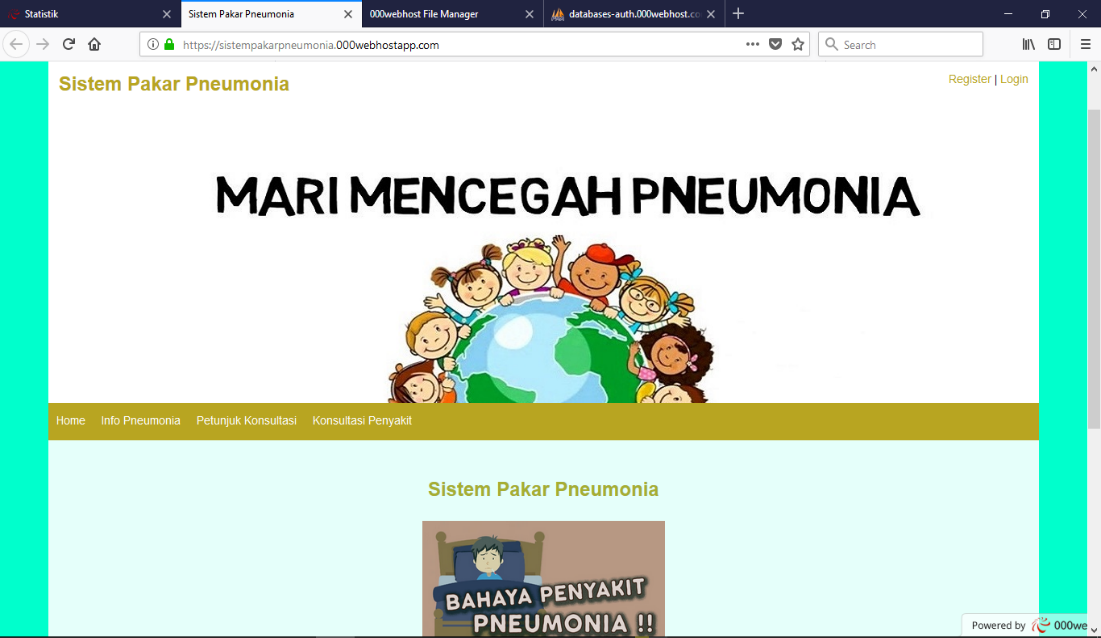
Gambar di atas menampilan halaman menu file manager pada *website* untuk memasukkan file-file *website*.

1. Langkah kesebelas, masukkan semua file-file *web* yang akan di *hosting*



**Gambar 5.19 Langkah Kesebelas Proses Hosting**

Gambar di atas menampilkan halaman upload file *website*.

1. Setelah file selesai di *upload*, langkah selanjutnya klik url yang ada di beranda *website* yang mengarah kepada website yang sudah dibuat.

**Gambar 5.20 Langkah Keduabelas Proses Hosting**

Gambar di atas menampilkan *website* yang telah di *hosting*.

* + 1. **Pengujian Aplikasi**

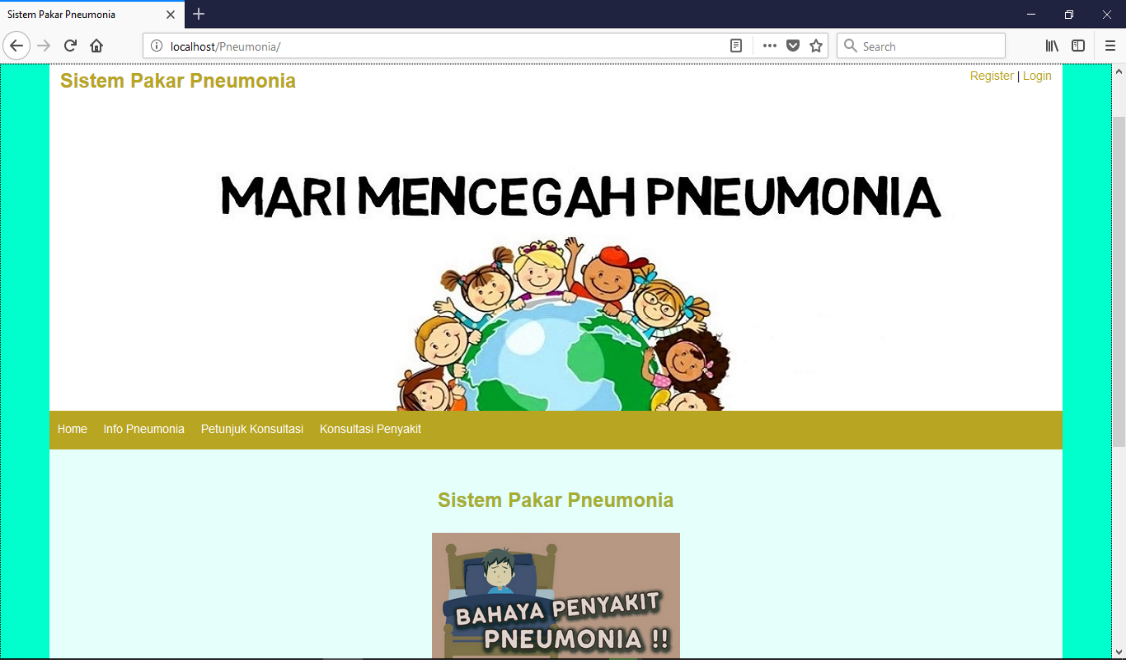
Pada tahap ini dilakukan pengujian aplikasi menggunakan teknik pengujian yang akan menguji fungsionalitas dari menu-menu yang disediakan dalam sistem. Hasil dari pengujian fungsionalitas aplikasi dapat dilihat pada tabel dibawah ini :

**Tabel 5.1 Pengujian Aplikasi**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| 1. | Beranda | Tampil halaman sekilas tentang sistem pakar. | Berhasil |
| 2. | Melihat Penyakit *Pneumonia* pada Anak | Tampil halaman infomasi penyakit *pneumonia* pada anak. | Berhasil |
| 3. | Register | Tampil form yang harus diisikan user apabila ingin menjadi *member*. | Berhasil |
| 4. | Login | Setelah register tampil form login agar *user* dapat masuk ke dalam sistem dan bisa melakukan konsultasi. | Berhasil |
| 5. | Melihat Petunjuk Penggunaan | Tampil halaman petunjuk penggunaan sistem. | Berhasil |
| 6. | Melakukan Konsultasi | Tampil halaman konsultasi yang akan dilakukan *member.* | Berhasil |
| 7. | Melihat daftar hasil konsultasi | Tampil halaman melihat seluruh hasil konsultasi member | Behasil |
| 8. | Logout | Tampil halaman awal sistem. | Berhasil |
| 9. | Beranda admin | Tampil halaman awal sistem untuk admin. | Berhasil |
| 10. | Mengelola Daftar Member | Tampil halaman kelola daftar member untuk admin. | Berhasil |
| 11. | Mengelola Daftar Gejala | Tampil halaman kelola daftar gejala untuk admin. | Berhasil |
| 12. | Mengelola Data Penyakit | Tampil halaman kelola daftar penyakit untuk admin. | Berhasil |
| 13. | Mengelola Daftar Relasi | Tampil halaman kelola daftar relasi untuk admin. | Berhasil |
| 14. | Melihat Rekap Konsultasi | Tampil halaman melihat rekap konsultasi untuk admin. | Berhasil |

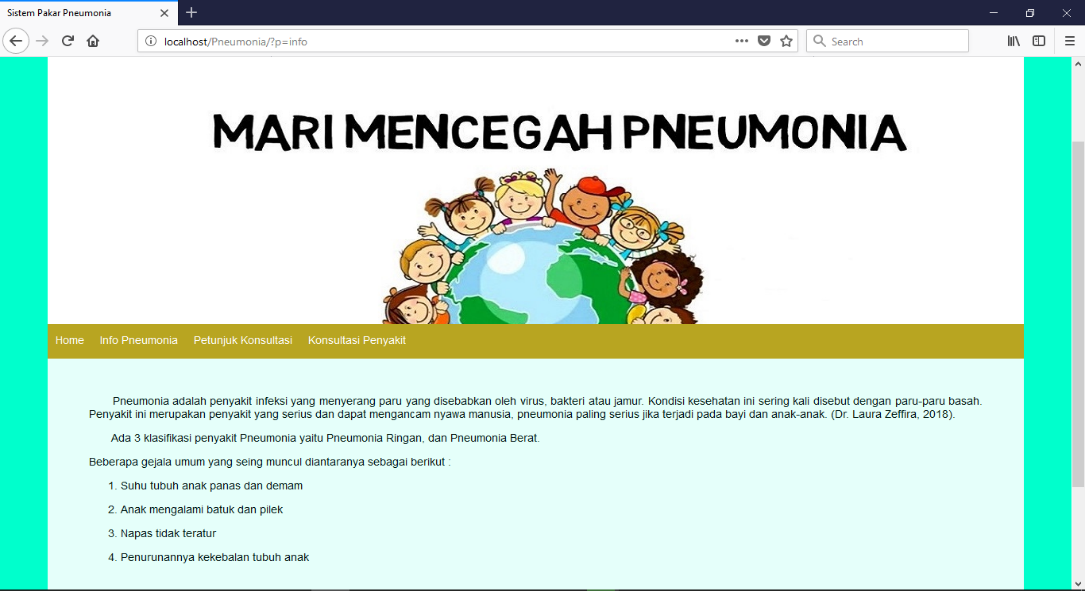
* + 1. **Pengujian *Interface***

Setelah pengujian aplikasi selesai, pada tahap selanjutnya dilakukan pengujian *interface* yang merupakan tampilan sistem yang sudah dibuat

1. **Tampilan Halaman Home**

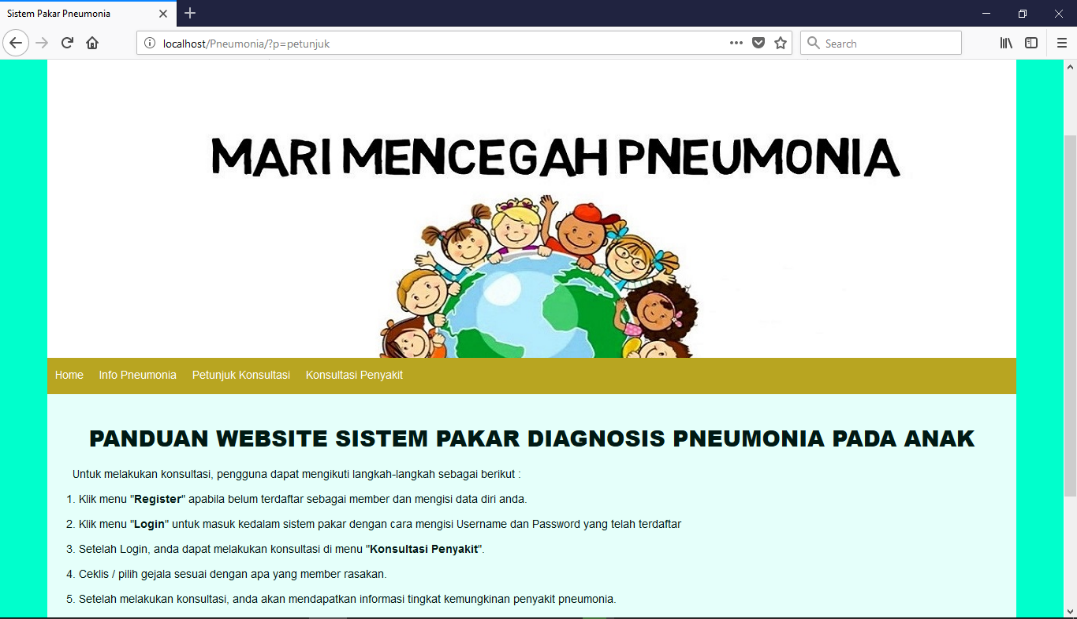
**Gambar 5.21 Tampilan Halaman Home**

Gambar di atas merupakan tampilan dari menu home. Halaman ini berisikan penjelasan sekilas tentang sistem pakar.

1. **Tampilan Halaman Tentang Penyakit *Pneumonia* pada Anak**

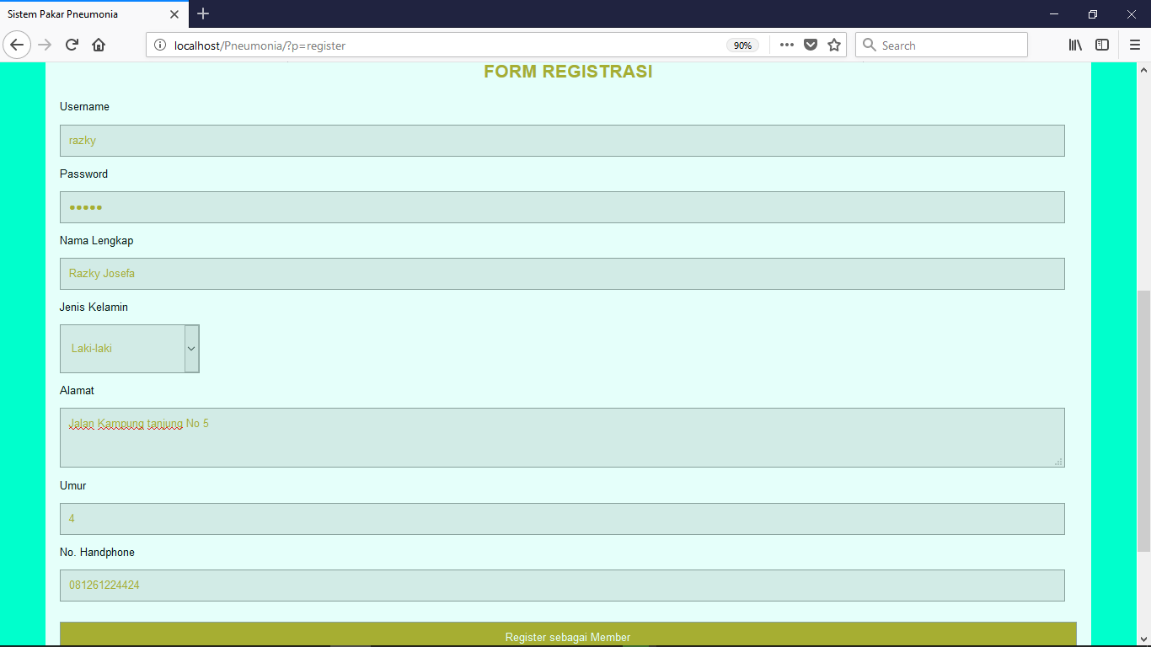
**Gambar 5.22 Tampilan Halaman Tentang Penyakit *Pneumonia* pada Anak**

Gambar di atas merupakan tampilan halaman penyakit *pneumonia* yang berisi penjelasan tentang penyakit *pneumonia* beserta gejala-gejala yang ada.

1. **Tampilan Petunjuk Konsultasi**

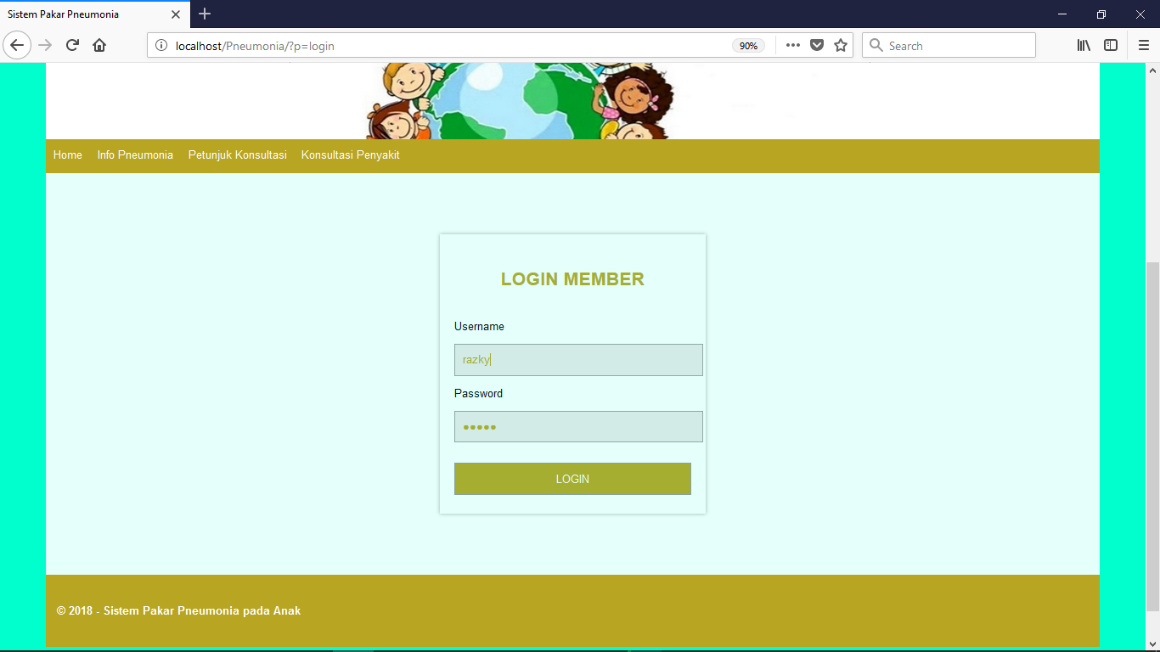
**Gambar 5.23 Tampilan Halaman Petunjuk Konsultasi**

Gambar di atas merupakan tampilan halaman berisikan informasi petunjuk penggunaan sistem pakar penyakit *pneumonia* pada anak.

1. **Tampilan Halaman Registrasi**

**Gambar 5.24 Tampilan Halaman Registrasi Member**

Gambar di atas merupakan tampilan halaman registrasi. Halaman ini menampilkan form data diri yang akan diisi oleh calon member sistem pakar.

1. **Tampilan Halaman Login**

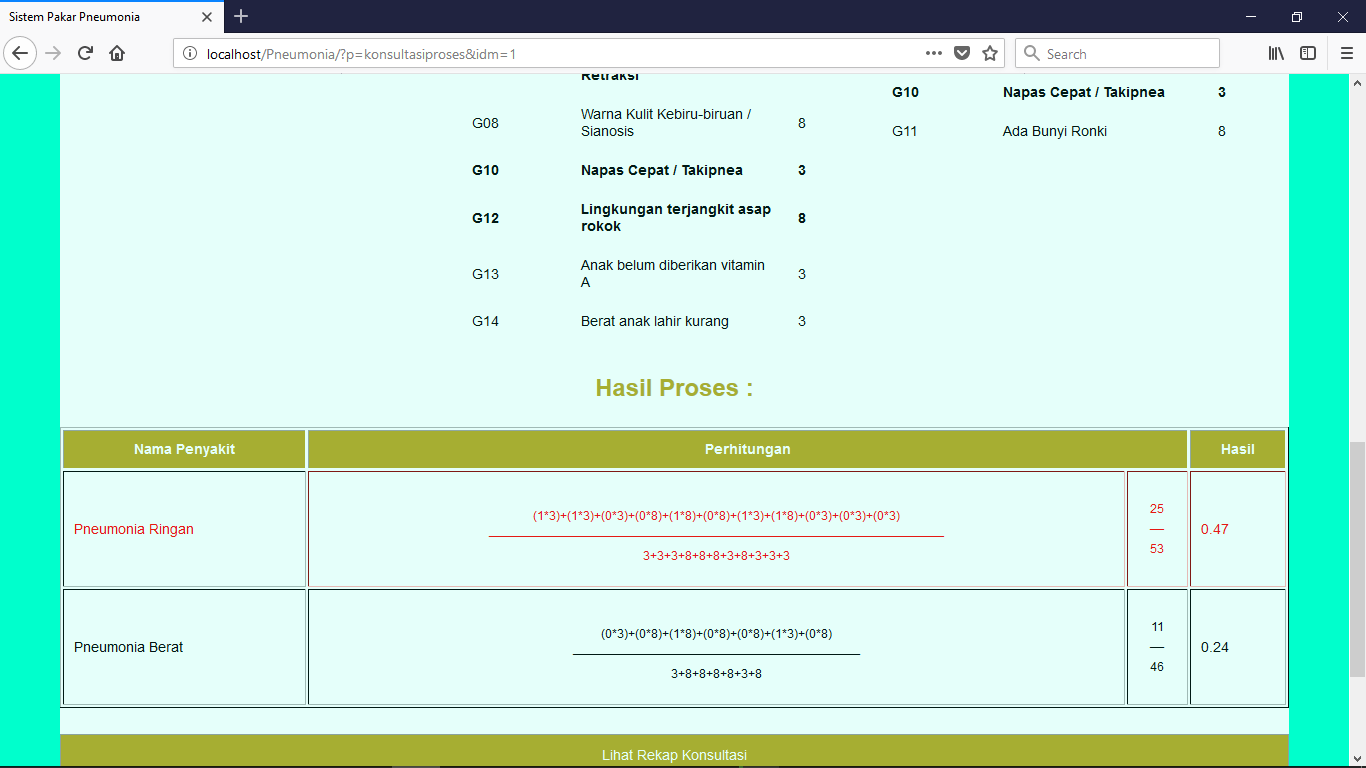
**Gambar 5.25 Tampilan Halaman Registrasi Member**

Gambar di atas merupakan tampilan halaman login yang terdapat form username dan password yang harus diisi oleh member untuk melakuan login.

1. **Tampilan Halaman Konsultasi**

**Gambar 5.26 Tampilan Halaman Konsultasi *Member***

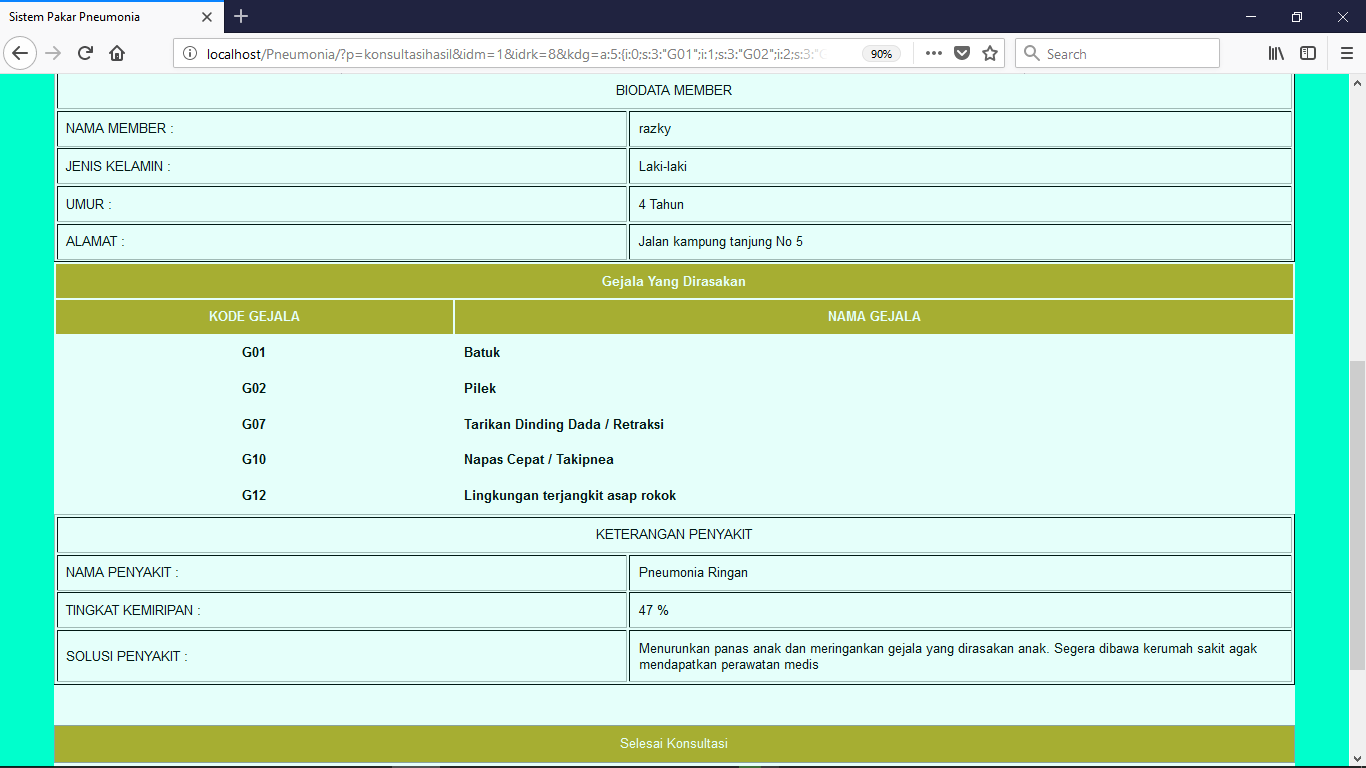
Gambar di atas merupakan tampilan halaman konsultasi dimana member yang ingin melakukan konsultasi akan memilih gejala sesuai dengan yang dirasakan.

1. **Tampilan Proses Perhitungan**

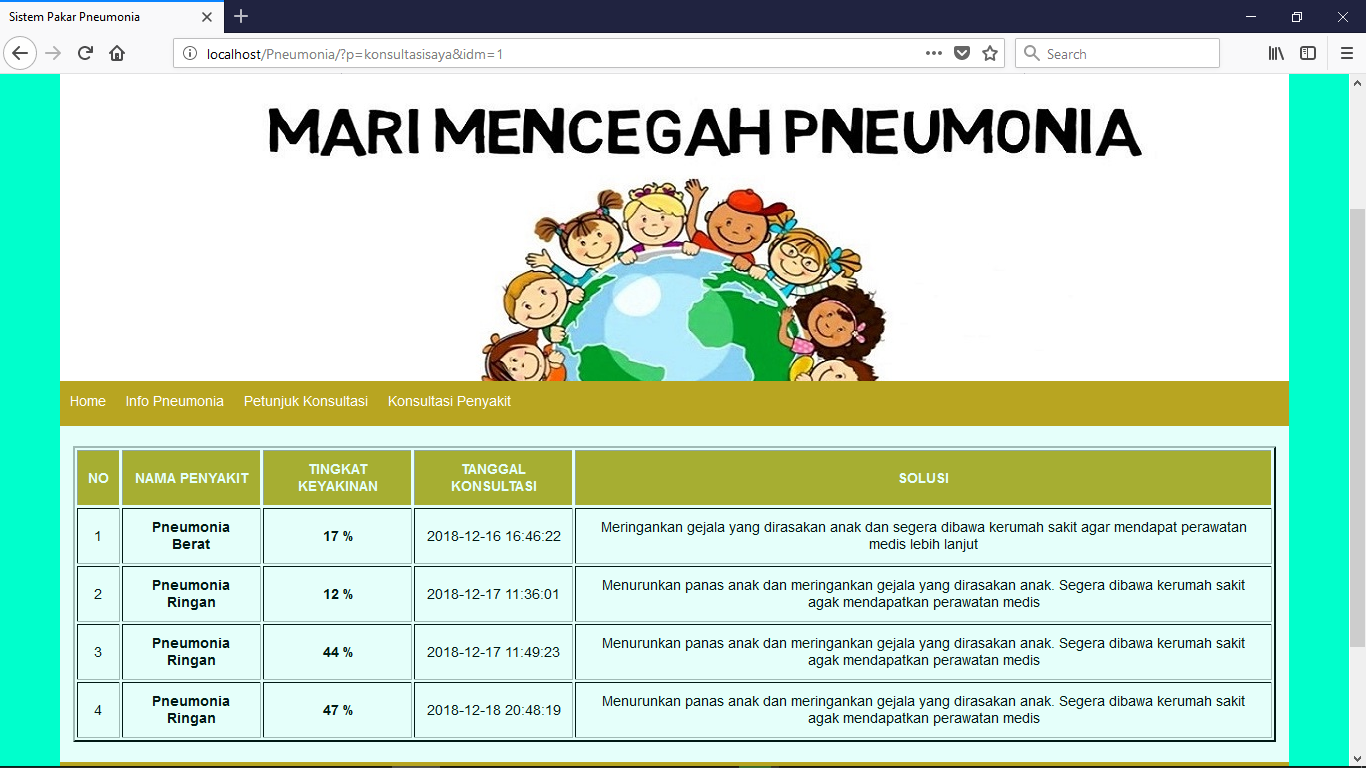
**Gambar 5.27 Tampilan Halaman Proses Perhitungan**

Gambar di atas merupakan tampilan halaman proses perhitungan. Halaman ini berisikan hasil perhitungan tingkat kemiripan dengan gejala yang dirasakan oleh member.

1. **Tampilan Hasil Konsultasi**

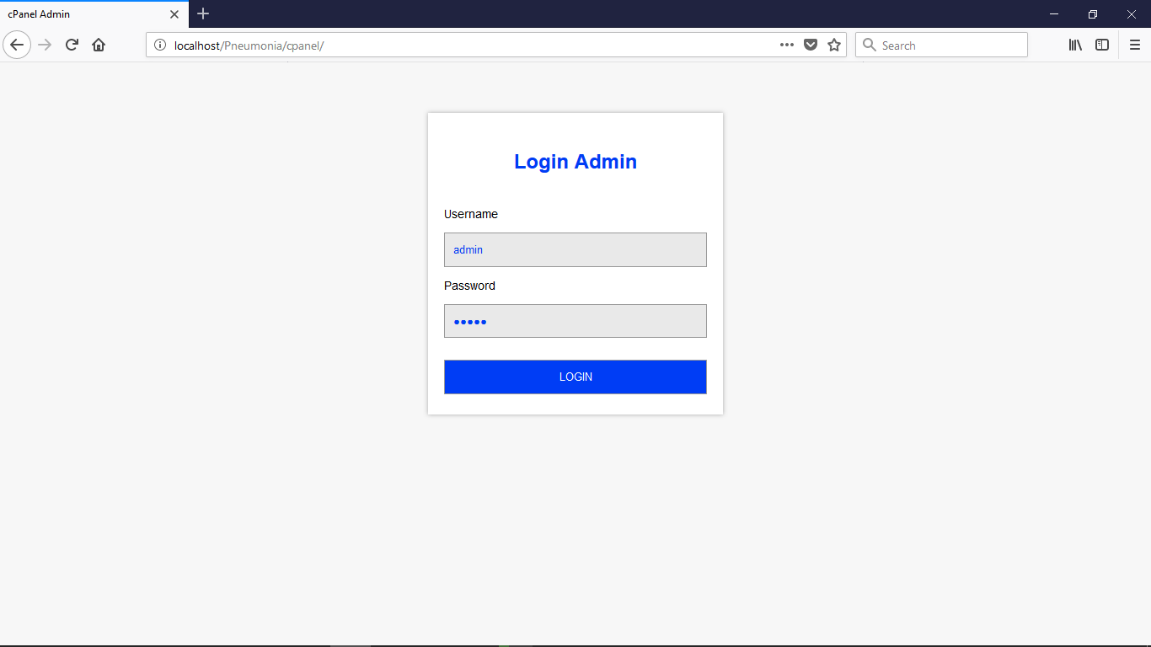
**Gambar 5.28 Tampilan Halaman Hasil Konsultasi**

Gambar di atas merupakan tampilan halaman hasil konsultasi member. Halaman ini berisikan hasil konsultasi dan keterangan penyakit.

1. **Tampilan Daftar Hasil Konsultasi**

**Gambar 5.29 Tampilan Halaman Daftar Hasil Konsultasi**

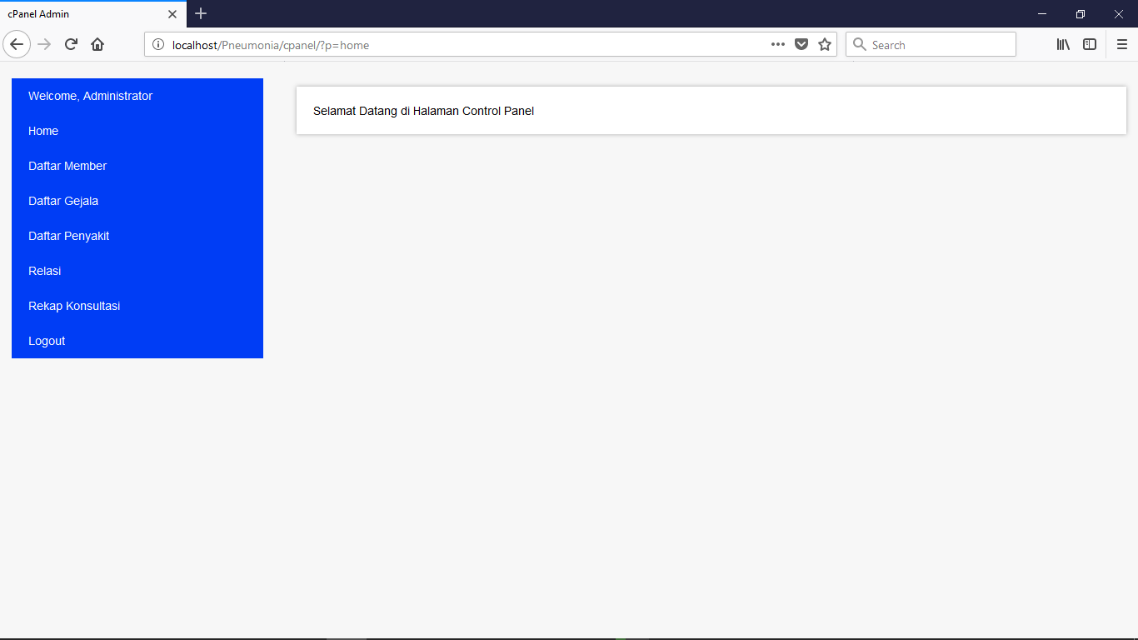
Gambar di atas merupakan tampilan halaman daftar hasil dari konsultasi yang dilakukan oleh member.

1. **Tampilan Login Admin**

**Gambar 5.30 Tampilan Halaman Login Admin**

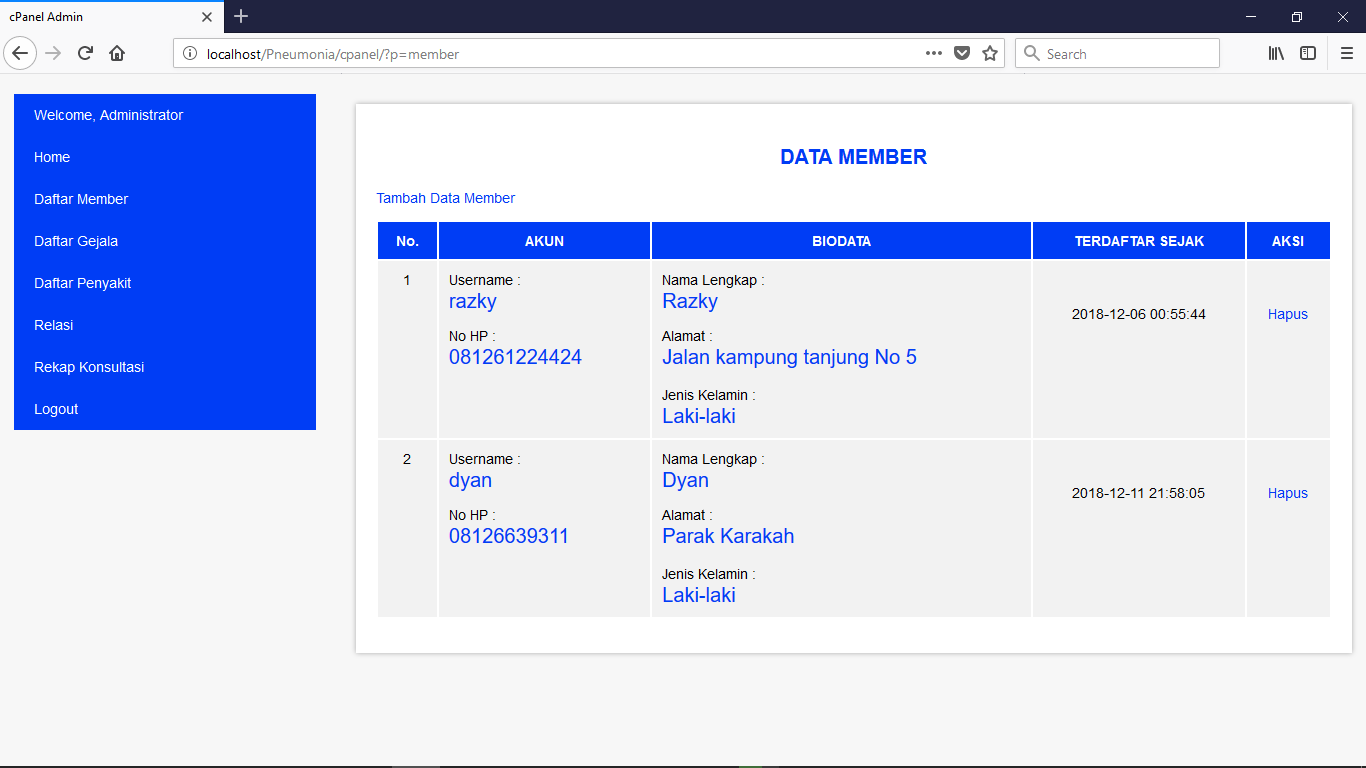
Gambar di atas merupakan tampilan halaman untuk login ke dalam halaman admin.

1. **Tampilan Home Admin**



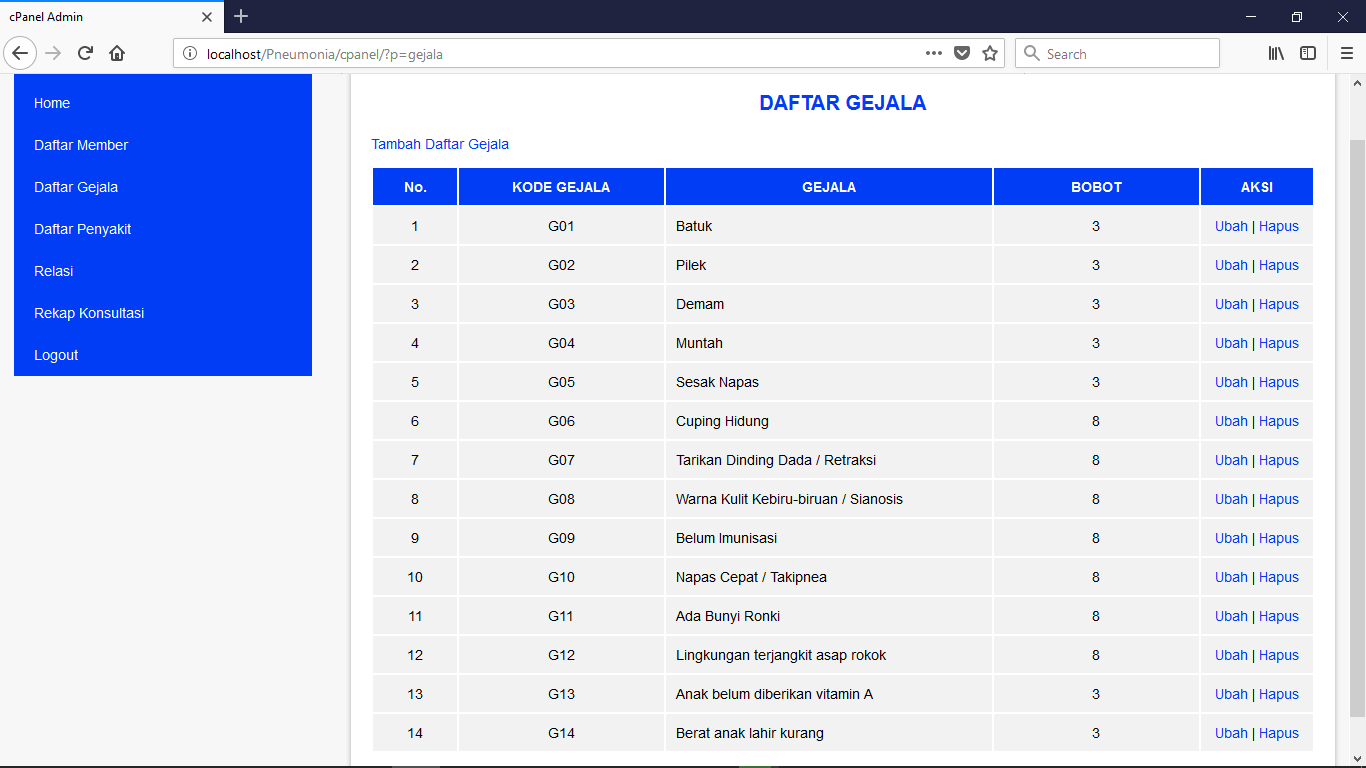
**Gambar 5.31 Tampilan Halaman Home Admin**

Gambar di atas merupakan tampilan halaman home admin setelah selesai melakukan login.

1. **Tampilan Daftar *Member***

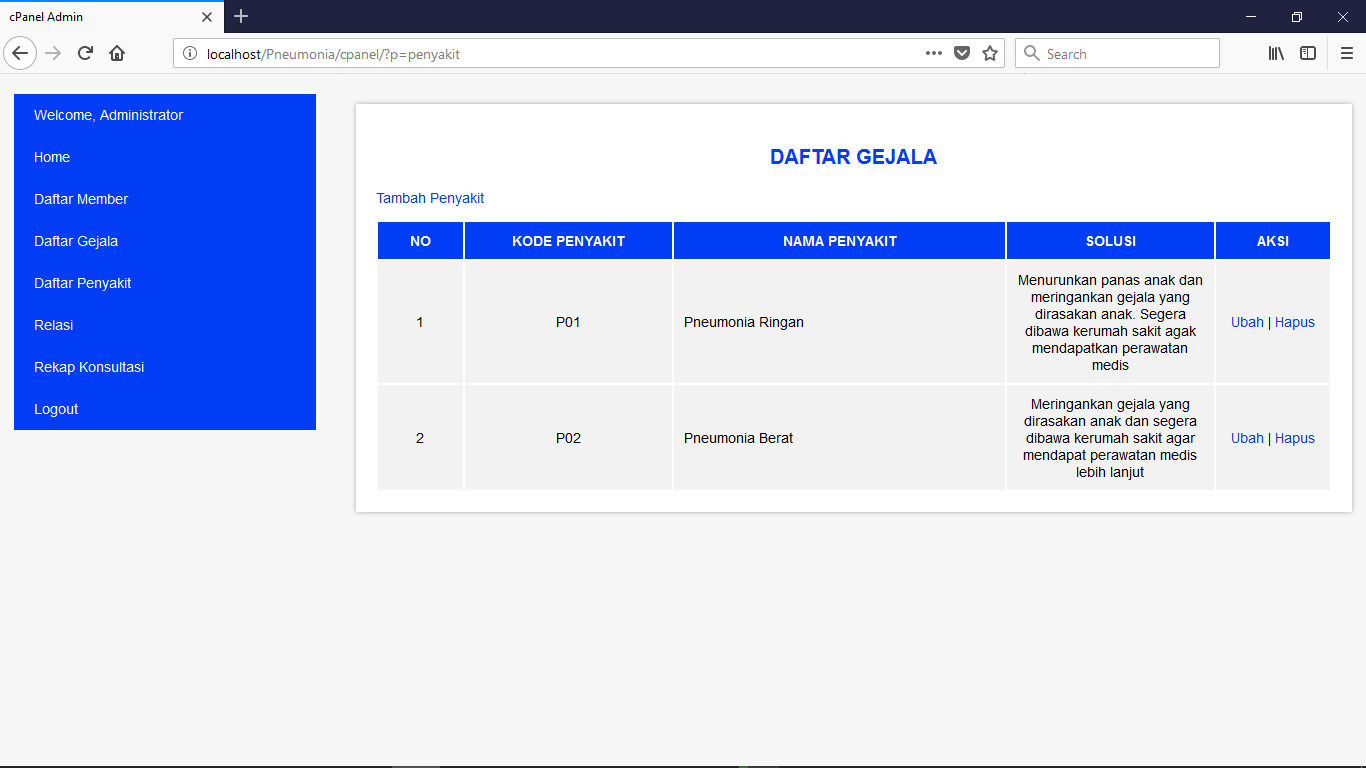
**Gambar 5.32 Tampilan Halaman Daftar *Member***

Gambar di atas merupakan tampilan untuk admin saat melakukan kelola daftar member yang ada, dimana admin dapat melakukan tambah dan hapus member yang ada.

1. **Tampilan Daftar Gejala**

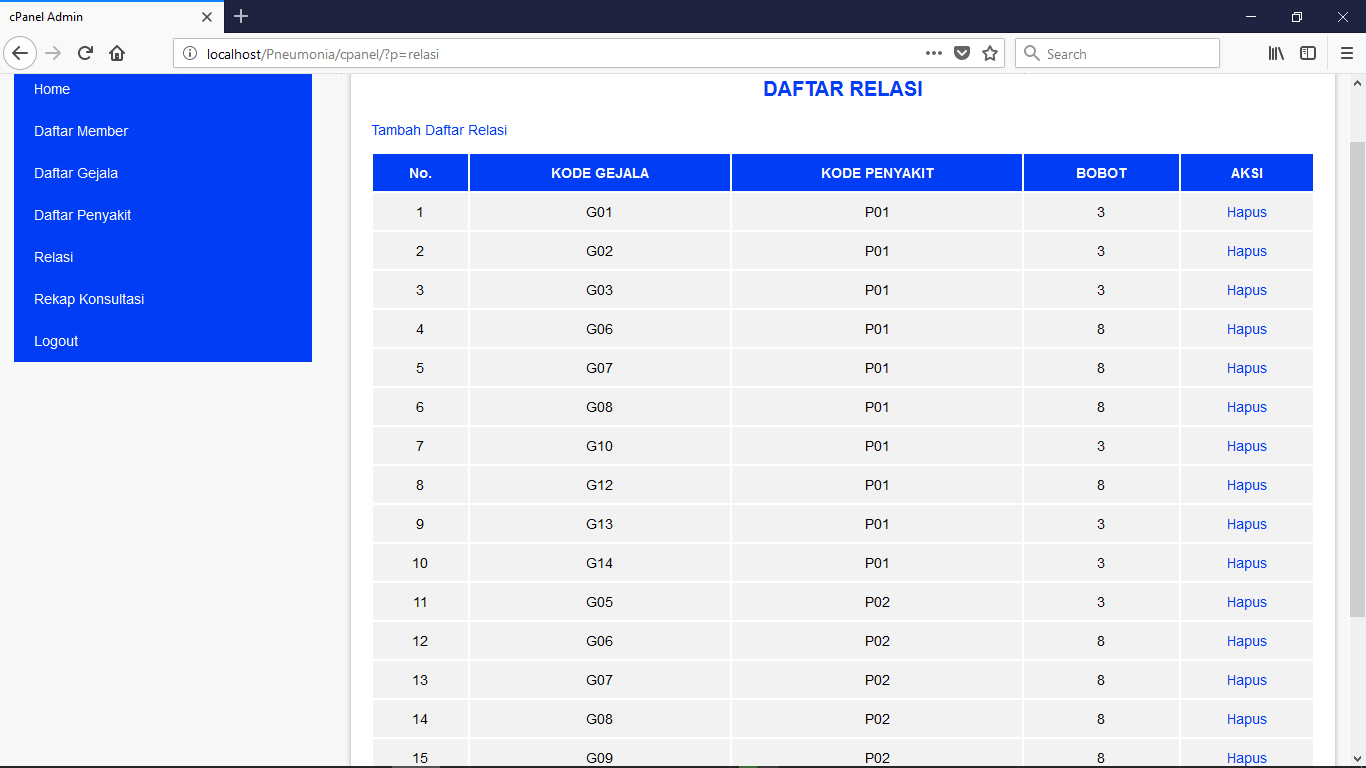
**Gambar 5.33 Tampilan Halaman Daftar Gejala**

Gambar di atas merupakan tampilan untuk admin saat melakukan kelola gejala yang ada, dimana admin dapat melakukan tambah, edit, dan hapus gejala yang ada.

1. **Tampilan Daftar Penyakit**

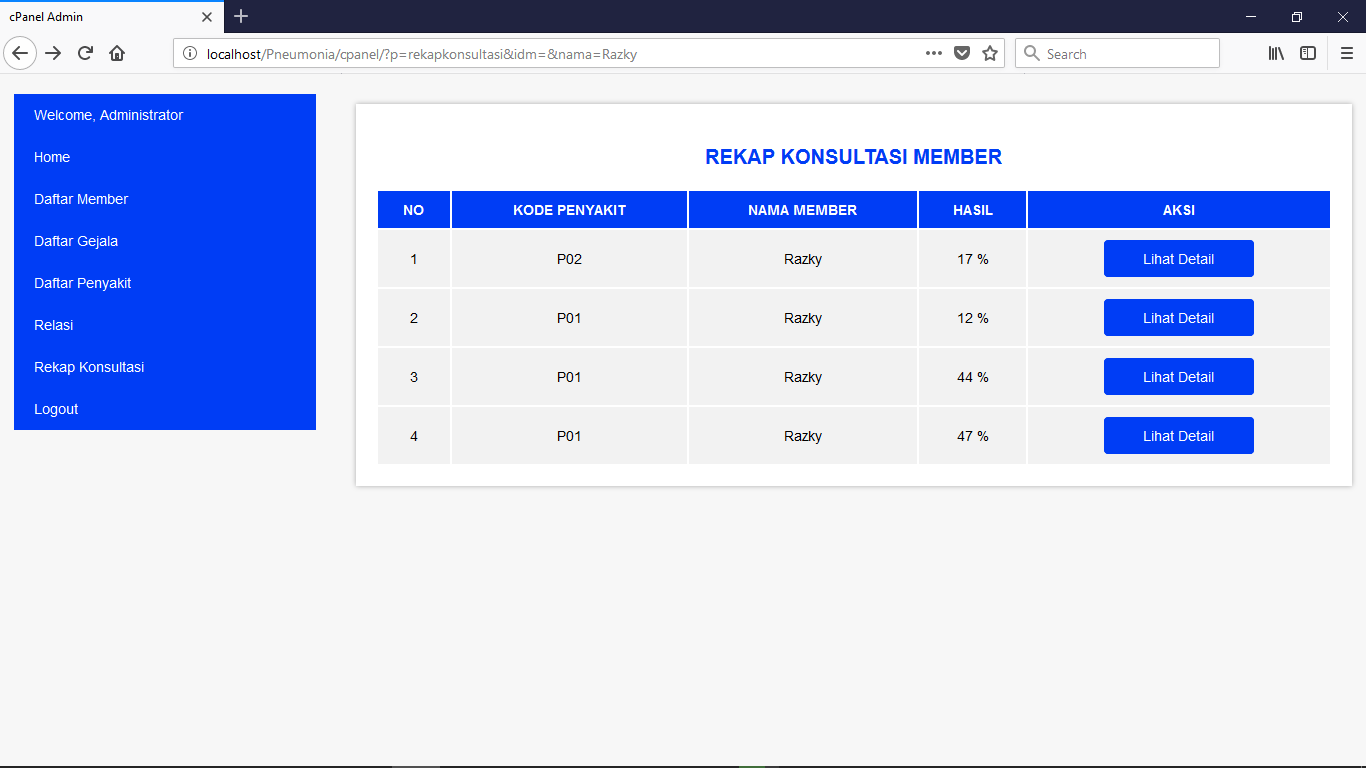
**Gambar 5.34 Tampilan Halaman Daftar Penyakit**

Gambar di atas merupakan tampilan untuk admin saat melakukan kelola penyakit yang ada, dimana admin dapat melakukan tambah, edit, dan hapus penyakit yang ada.

1. **Tampilan Relasi**

**Gambar 5.35 Tampilan Halaman Data Relasi**

Gambar di atas merupakan tampilan untuk admin saat melakukan kelola relasi yang ada, dimana admin dapat melakukan tambah dan hapus relasi yang ada.

1. **Tampilan Rekap Konsultasi**

**Gambar 5.36 Tampilan Halaman Rekap Konsultasi**

Gambar di atas merupakan tampilan dari rekap konsultasi yang dapat dilihat oleh admin, pada halaman ini admin hanya dapat melihat riwayat konsultasi member.

**BAB VI**

**PENUTUP**

* 1. **Kesimpulan**

Berdasarkan analisa yang telah dilakukan pada bab-bab yang ada. Maka dapat diambil kesimpulan sebagai berikut :

1. Dengan menggunakan sistem pakar ini bisa mempermudah orangtua dalam mengetahui tentang informasi penyakit *pneumonia* pada anak. Hal ini dibuktikan dengan orangtua yang mempunyai anak memiliki pengetahuan baru tentang penyakit *pneumonia* pada anak.
2. Dengan menggunakan sistem pakar ini bisa membantu orangtua dalam mengetahui gejala penyakit *pneumonia* pada anak. Hal ini dibuktikan dengan memasukkan gejala yang ada, maka dapat menghasilkan hasil diagnosa penyakit *pneumonia* pada anak.
3. Dengan adanya aplikasi sistem pakar ini orangtua dapat mengetahui solusi berupa pencegahan dan penanganan lebih awal pada anak yang terkena penyakit *pneumonia*. Hal ini dibuktikan dengan adanya pencegahan dan penanganan awal setelah mendapatkan hasil diagnosa.
   1. **Keterbatasan Sistem**

Dalam perancangan sistem pakar ini terdapat beberapa keterbasan, diantaranya sebagai berikut :

1. Aplikasi sistem pakar ini sudah bisa memberikan informasi, gejala dan cara untuk penanganan secara cepat terhadap penyakit *pneumonia* pada anak.
2. Aplikasi sistem pakar ini dapat digunakan oleh orangtua untuk melakukan konsultasi penyakit *pneumonia* pada anak.
3. Aplikasi sistem pakar ini sudah dapat mengirimkan hasil konsultasi dan dapat dilihat oleh dokter sebelum orangtua datang menemui dokter.
4. Aplikasi sistem pakar ini hanya bisa menggunakan metode case based reasoning.
   1. **Saran**

Dari kesimpulan yang telah diambil, sistem pakar ini masih memiliki beberapa kekurangan yang diharapkan agar program sistem pakar ini dapat berjalan maksimal dan hasil yang lebih akurat tentunya dibutuhkan pengembangan lebih lanjut. Beberapa hal yang perlu dikembangkan sebagai berikut :

1. Diharapkan program sistem pakar ini dikembangkan lebih lanjut dengan menambahkan gejala-gejala penyakit *pneumonia* pada anak, pengobatan dan penanganan yang lebih optimal.
2. Diharapkan penelitian ini bisa dikembangkan agar ruang lingkup terhadap konsultasi penyakit saluran pernapasan secara umum, tidak sekedar penyakit *pneumonia* dan dilakukan dengan metode yang berbeda.

**DAFTAR PUSTAKA**

Akmal, Faza, dan Sri Winiarti. "Sistem Pakar Untuk Mendiagnosa Penyakit Lambung dengan Implementasi Metode CBR (Case-Based Reasoning) Berbasis Web." *JSTIE (Jurnal Sarjana Teknik Informatika) (E-Journal)* 2.1 (2014): 119-129.

Budiharto, Widodo, dan Derwin Suhartono. “Artificial Intelligence Konsep dan Penerapan.” Yogyakarta : Andi (2014).

Efni, Yulia, Rizanda Machmud, and Dian Pertiwi. "Faktor Risiko yang Berhubungan dengan Kejadian Pneumonia pada Balita di Kelurahan Air Tawar Barat Padang." *Jurnal Kesehatan Andalas* 5.2 (2016).

Gulo, Amonius Asmin Hardi Saputra, dan Muhammad Syahrizal. "PERANCANGAN APLIKASI SISTEM PAKAR MENDIAGNOSA PENYAKIT HEMOFILIA PADA MANUSIA MENERAPKAN METODE CASE BASED REASONING." *Pelita Informatika: Informasi dan Informatika* 17.1 (2018).

Hartati, Sri dan Sari Ismawanti. “Sistem Pakar dan Pengembangannya.” Yogyakarta : Graha Ilmu (2008).

Kusuma, Diki Andita, dan Chairani Chairani. "Rancang Bangun Sistem Pakar Pendiagnosa Penyakit Paru-Paru Menggunakan Metode Case Based Reasoning." *Jurnal Infotel* 6.2 (2014): 57-62.

Mandala, Eka Praja Wiyata. "Web Programing Project 1: epwm forum." Yogyakarta: *CV. ANDI OFFSET* (2015).

Minarni, Indra Warman, dan Wenda Handayani. "Case-Based Reasoning (CBR) Pada Sistem Pakar Identifikasi Hama dan Penyakit Tanaman Singkong dalam Usaha Meningkatkan Produktivitas Tanaman Pangan." *Jurnal Teknoif* 5.1 (2017).

Monita, Osharinanda, Finny Fitry Yani, dan Yuniar Lestari. "Profil pasien pneumonia komunitas di bagian anak RSUP DR. M. Djamil Padang Sumatera Barat." *Jurnal Kesehatan Andalas* 4.1 (2015).

Mujilahwati, Siti. "Diagnosa penyakit tanaman hias menggunakan metode certainty factor berbasis WEB." *Universitas Islam Lamongan* (2014).

Nurnajiah, Mia, Rusdi Rusdi, dan Desmawati Desmawati. "Hubungan Status Gizi dengan Derajat Pneumonia pada Balita di RS. Dr. M. Djamil Padang." *Jurnal Kesehatan Andalas* 5.1 (2016).

Putra, Teri Ade. "PERANCANGAN SISTEM PAKAR UNTUK MENDIAGNOSA KELAINAN SEKS MENGGUNAKAN METODE CERTAINTY FACTOR BERBASIS WEB." *PSIKOLOGI 165* 11.2 (2018).

Rahajoe, Nastiti N, Bambang Supriyatno, dan Darmawan Budi Setyanto. “Buku Ajar Respirologi Anak” Jakarta : Badan Penerbit IDAI (2008).

Rahman, Fakhrul, Eka Praja Wiyata Mandala, Teri Ade Putra. "PERANCANGAN APLIKASI SISTEM PAKAR DENGAN MENGGUNAKAN METODE CERTAINTY FACTOR UNTUK MENENTUKAN JENIS GANGGUAN DISLEKSIA BERBASIS WEB." *Jurnal INKOFAR* 1.1 (2017).

Rismawan, Tedy dan Sri Hartati. "Case-Based Reasoning untuk Diagnosa Penyakit THT (Telinga Hidung dan Tenggorokan)." *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)* 6.2 (2012).

Sovia, Rini, Aulia Fitrul Hadi, dan Ani Yuliana. "SISTEM PAKAR MENDIAGNOSA PENYAKIT HIPERTENSI MENGGUNAKAN CASE-BASED REASONING (CBR)." *INFOTEKNO* 1.1 (2018).

Sovia, Rini, dan Jimmy Febio. "Membangun Aplikasi E-Library Menggunakan Html, Php Script, Dan Mysql Database." *Jurnal MEDIA PROCESSOR* 6.2 (2017).

S, Rosa A, M. Shalahudin. “*Modul Pembelajaran Rekayasa Perangkat Lunak*.”Bandung: Informatika (2013).

Simarmata, Janner. “*Rekayasa Perangkat Lunak*.” Yogyakarta: Andi (2010).

Sutojo, T , dkk. “*Kecerdasan Buatan*.” Yogyakarta: Andi (2011).

Putri, Tiara Eka, Desi Andreswari, and Rusdi Efendi. "Implementasi Metode CBR (Case Based Reasoning) Dalam Pemilihan Pestisida Terhadap Hama Padi Sawah Menggunakan Algoritma K-Nearest Neighbor (KNN) (Studi Kasus Kabupaten Seluma)." *Rekursif: Jurnal Informatika* 4.1 (2016).

Utomo, Dito Putro, dan Surya Darma Nasution. "SISTEM PAKAR MENDETEKSI KERUSAKAN TONER DENGAN MENGGUNAKAN METODE CASE BASED-REASONING." *JURIKOM (Jurnal Riset Komputer)* 3.5 (2016).

Yuhandri, Yuhandri. "Diagnosa Penyakit Osteoporosis Menggunakan Metode Certainty Factor." *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 2.1 (2018): 422-429.