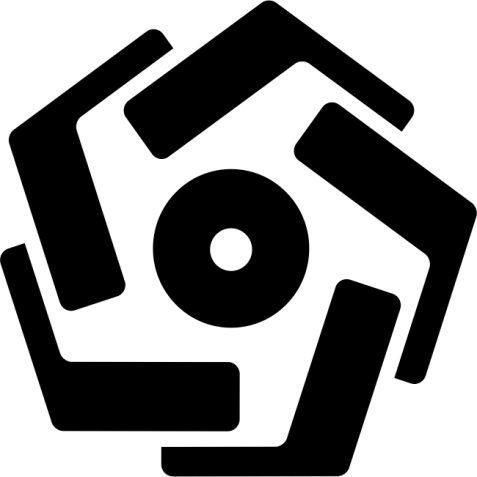
**PERANCANGAN APLIKASI SISTEM PAKAR METODE CERTAINTY FACTOR UNTUK PENYAKIT PARU-PARU BERBASIS ANDROID**

**SKRIPSI**



disusun oleh

**Annisa Alfi Fauziana**

**13.11.7390**

**PROGRAM SARJANA**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**AMIKOM YOGYAKARTA**

**YOGYAKARTA**

**2016**

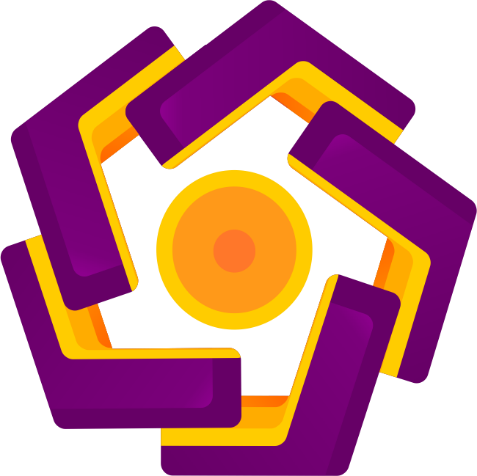
**PERANCANGAN APLIKASI SISTEM PAKAR METODE CERTAINTY FACTOR UNTUK PENYAKIT PARU-PARU BERBASIS ANDROID**

**SKRIPSI**

untuk memenuhi sebagian persyaratan

mencapai derajat Sarjana S1

pada jurusan Teknik Informatika



disusun oleh

**Annisa Alfi Fauziana**

**13.11.7390**

**PROGRAM SARJANA**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**AMIKOM YOGYAKARTA**

**YOGYAKARTA**

**2016 2016**

**PERSETUJUAN**

**SKRIPSI**

**PERANCANGAN APLIKASI SISTEM PAKAR METODE CERTAINTY FACTOR UNTUK PENYAKIT PARU-PARU BERBASIS ANDROID**

yang dipersiapkan dan disusun oleh

Annisa Alfi Fauziana

13.11.7390

telah disetujuai oleh Dosen Pembimbing Skripsi

pada tanggal 24 November 2016

**Dosen Pembimbing,**

**Heri Sismoro, M.Kom**

**NIK. 190302057**

# **PERNYATAAN**

**PENGESAHAN**

**SKRIPSI**

**PERANCANGAN APLIKASI SISTEM PAKAR METODE CERTAINTY FACTOR UNTUK PENYAKIT PARU-PARU BERBASIS ANDROID**

yang dipersiapkan dan disusun oleh

Annisa Alfi Fauziana

13.11.7390

telah dipertahankan di depan Dewan Penguji

pada tanggal 20 Oktober 2016

**Susunan Dewan Penguji**

**Nama Penguji Tanda Tangan**

**Tony Hidayat, M.Kom \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NIK. 190302182**

**Mei Parwanto Kurniawanm M.Kom \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NIK. 190302187**

**Joko Dwi Santoso, M.Kom \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NIK. 190302181**

Skripsi ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana Komputer

Tanggal 28 November 2016

**KETUA STMIK AMIKOM YOGYAKARTA**

**Prof. Dr. M. Suyanto, M.M.**

**NIK. 190302001**

Saya yang bertandatangan dibawah ini menyatakan bahwa, skripsi ini merupakan karya saya sendiri (ASLI), dan isi dalam skripsi ini tidak terdapat karya yang pernah diajukan oleh orang lain untuk memperoleh gelar akademis di suatu institusi pendidikan tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis dan/atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Segala sesuatu yang terkait dengan naskah dan karya yang telah dibuat adalah menjadi tanggungjawab saya pribadi.

Yogyakarta, 28 November 2016

Meterai

Rp. 6.000

Annisa Alfi Fauziana

NIM. 13.11.7390

**MOTTO**

"Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain), dan hanya kepada Tuhanmulah engkau berharap."

**(Qur'an Surat Asy-Syarh [94] Ayat 7-8)**

”Barang siapa yang menghendaki kehidupan dunia maka wajib baginya memiliki ilmu, dan barang siapa yang menghendaki kehidupan Akhirat, maka wajib baginya memiliki ilmu, dan barang siapa menghendaki keduanya

maka wajib baginya memiliki ilmu”.

**(Hadits Riwayat Turmudzi)**

“Man Jadda Wajada”

**(Filsafat Arab)**

“Gemah Ripah Lohjinawi, Murah Kang Sarwo Tinuku, Thukul Kang

Sarwo Tinandur, Toto Tentrem Kerto Raharjo.”

**(Ilmu Kearifan Jawa)**

“Iman tanpa ilmu bagaikan lentera di tangan bayi, namun ilmu

tanpa iman bagaikan lentera ditangan pencuri.”

**(Haji Abdul Malik Karim Amrullah)**

“Our greatest weakness lies in giving up. The most certain way to

succeed is always to try one more time.”

**(Thomas Alva Edison)**

“Saya sedikit beruntung, dan lebih banyak bekerja keras.”

**(Penulis)**

**PERSEMBAHAN**

Puji syukur penulis panjatkan kehadirat Allah SWT, atas segala limpahan rahmat dan ridho-Nya yang telah memberikan kesehatan, kelancaran, keteguhan dan membekali anugerah ilmu sehingga penulis dapat menyelesaikan skripsi ini. Skripsi ini dipersembahkan untuk:

1. Allah SWT yang telah memberikan kesehatan, kemudahan dan kelancaran dalam penyusunan dan pembuatan skripsi ini.
2. Kedua orang tua tercinta, Bapak Suparjana dan Ibu Suwati, yang telah menjadi orangtua terhebat, yang selalu memberikan kasih sayang yang tak ada batasnya, yang tak pernah lelah sedikit pun untuk mendoakan dan memberikan segala bentuk dukungan maupun nasehat pada penulis sehingga mampu menyelesaikan skripsi ini.
3. Adikku tersayang Zahra Sella Natta, yang selalu mendoakan dan mendukung penulis, yang telah memberikan semangat dan keceriaan dalam keluarga.
4. Seorang pria yang sabar, pengertian, Ahmad Yusuf Affandi yang selalu mendampingi dari awal hingga akhir pengerjaan skripsi, membantu dalam pengerjaan skripsi, yang selalu mendukung setiap keputusan yang penulis pilih, dan memberikan semangat serta doa.
5. Bapak Heri Sismoro, M.Kom selaku dosen pembimbing yang senantiasa membimbing dengan penuh kesabaran dan selalu memberikan solusi agar skripsi ini dapat terselesaikan dengan baik.
6. Bapak Tony Hidayat, M.Kom , Bapak Mei P Kurniawan, M.Kom , dan Bapak Joko Dwi Santoso M.Kom yang telah memberikan saran saat ujian pendadaran.
7. Ibu Hartatik, M.Cs yang telah memberikan dukungan, memberikan saran, sehingga semakin memantapkan niat untuk mempelajari tentang sistem pakar, semoga ibu banyak mendapatkan keberkahan dan dilancarkan di segala urusan.
8. Bapak Tri Susanto M.Kom yang banyak memberikan motivasi saat mengasisteni.
9. Bapak dan Ibu Dosen STMIK Amikom Yogyakarta yang telah memberikan banyak materi, sehingga dapat dijadikan sebagai ilmu-ilmu yang menunjang penelitian ini.
10. Mas Albar dan Mas Agung Nur Hidayat yang telah membantuk membimbing proses penyelesaian skripsi.
11. Milawati Hartono, Lisa Dinda Yunita, Ka Mohammad Santosa Mulyo Diningrat, Ka Toto Indriyatmoko, Ka Ali Maksum, dan Ka Farid yang selalu memberi semangat, memberi dukungan setiap keputusan yang diambil, yang selalu memberikan keceriaan dimana saja.
12. Keluarga besar 13-S1 TI-09, teman-teman yang selalu menemani dari awal semester yang telah berjuang bersama dalam proses perkuliahan, semoga kebersamaan di masa depan tetap terjaga.
13. Teman-teman kotrakan dota, yang telah memberikan tempat dan keceriaan selama pengerjaan skripsi.
14. Mukhlis Adi Irsyadi dan Muhammad Kholil Aziz Ashari yang telah mengajarkan tentang database.
15. Mba Fatika yang sudah mau datang jauh-jauh untuk hadir dan memberikan dukungan saat pendadaran
16. Annisa Arti Jayanti yang telah memberikan pinjaman kemeja putih saat ujian pendadaran dan Atika yang telah menemani dan memberi semangat saat mengerjakan skripsi.
17. Teman-teman angkatan 2013 yang sempat menjadi rekan dalam berbagai kesempatan, semoga tetap menjadi rekan yang baik saat bertemu di dunia pekerjaan.
18. Teman-teman Asisten Praktikum yang pernah menjadi rekan di Laboraturium Komputer, ridwan, najib, kawe.
19. Teman-teman Asisten Praktikum Hardware Software 1 dan 2.
20. Teman-teman yang sudah menyempatkan untuk hadir dan mendukung selama ujian pendadaran berlangsung, Qolbun, Mona, Husnia, Ade, Ibnu, Galih.
21. Meka Saima Perdani yang berada di Bengkulu dan Lathifah Alphiba Dalianti yang berada di Purwokerto yang sedang menempuh skripsi juga semoga lekas selesai dan semoga sukses selalu dan semoga kita selalu dilindungi oleg rahmat dan karunia-Nya.
22. Dan kepada semua teman-teman yang memberikan do'a dan dukungan, dari awal sampai akhir perjuangan pelaksanaan penelitian, sehingga dapat sampai pada di puncak keberhasilan, semoga mendapatkan balasan terbaik dari Allah Ta'ala.

**KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas limpahan rahmat, hidayah serta inayah\_Nya penulis masih diberi kesempatan dan kemudahan untuk menyelesaikan skripsi ini.

Skripsi ini disusun dalam rangka memenuhi salah satu syarat kelulusan perguruan tinggi Program Studi Strata-1 Teknik Informatika di Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta dan meraih gelar S.Kom. selain itu skripsi ini juga bertujuan agar pembaca dapt menambah pengetahuan tentang sistem pakar dan pengaplikasiannya untuk mendiagnosa penyakit paru-paru.

Penulis juga mengucapkan terima kasih yang setulus-tulusnya kepada:

1. Selaku Ketua STMIK AMIKOM Yogyakarta.
2. Bapak Sudarmawan, MT selaku Ketua Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta.
3. Bapak Heri Sismoro, M.Kom selaku Dosen Pembimbing yang telah memberikan bimbingan dengan penuh kesabaran.
4. Bapak Tony Hidayat, M.Kom , Bapak Mei P Kurniawan, M.Kom , dan Bapak Joko Dwi Santoso M.Kom selaku Dosen Penguji yang telah menguji skripsi.
5. Segenap dosen dan staf STMIK AMIKOM Yogyakarta yang telah memberikan banyak ilmu dan pengalaman.
6. Kedua Orangtua yang tak pernah lelah mendoakan dan memberikan dukungan.
7. Sahabat serta rekan-rekan 13-S1TI-09 yang beberikan banyak dukungan dan berbagi pengalaman.
8. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah banyak membantu segingga skripsi ini dapat terselesaikan.

Penulis juga memohon maaf apabila dalam penyusunan skripsi ini masih banyak kekurangan dan masih jauh dari kata sempurna. Penulis dengan hati terbuka menerima kritik dan saran dari para pembaca.

Semoga skripsi ini dapat menambah pengetahuan dan memberikan manfaat bagi para pembacanya maupun diri penulis sendiri serta dapat dipergunakan sebagaimana mestinya.

|  |
| --- |
| Yogyakarta, 25 Oktober 2016 |
|  |
| Annisa Alfi Fauziana |

**DAFTAR ISI**

**JUDUL** i

**PERSETUJUAN** ii

**PENGESAHAN** iii

**PERNYATAAN KEASLIAN** iv

**MOTTO** v

**PERSEMBAHAN** vi

**KATA PENGANTAR** viii

**DAFTAR ISI** ix

**DAFTAR TABEL** xii

**DAFTAR GAMBAR** xiii

**BAB I PENDAHULUAN** 1

* 1. Latar Belakang 1
  2. Rumusan Masalah 2
  3. Batasan Masalah 3
  4. Maksud dan Tujuan Penelitian 4
  5. Metode Penelitian 4
     1. Metode Pengumpulan Data 4
     2. Metode Analisis 5
     3. Metode Perancangan 5
     4. Metode Pengembangan 6
     5. Metode *Testing* 6
     6. Metode Implementasi 6
  6. Sistematika Penulisan 7

**BAB II LANDASAN TEORI** 9

* + 1. Paru-Paru 11
    2. Kecerdasan Buatan 19
    3. Sistem Pakar 21
    4. *Certainty* *Factor* 28
    5. Analisis dan Perancangan Sistem 30
    6. Konsep Dasar Android 39
    7. Konsep Basis Data 44
    8. ERD (*Entity* *Relationship* *Diagram*) 45
    9. UML (*Unified* *Modelling* *Language*) 49
    10. Bahasa Pemrograman 52
    11. Perangkat Lunak yang Digunakan 54

**BAB III ANALISIS DAN PERANCANGAN** 56

* + 1. Gambaran Umum Aplikasi 56
    2. Analisis Masalah 57
    3. Analisis SWOT 57
  1. Solusi yang Dapat Diterapkan 59
  2. Solusi Yang Dipilih 59
  3. Analisis Kebutuhan 60
     1. Analisis Kebutuhan Fungsional 60
     2. Analisis Kebutuhan Non-Fungsional 61
  4. Analisis Kelayakan 64
     1. Analisis Kelayakan Teknologi 64
     2. Analisis Kelayakan Hukum 64
     3. Analisis Kelayakan Operasional 64
     4. Analisis Kelayakan Ekonomi 65
  5. Analisis Pengetahuan 66
     1. Daftar Penyakit 66
     2. Daftar Gejala 67
     3. Nilai CF *Rule* untuk Gejala dan Penyakit / CF(H,E) 69
     4. Nilai CF *User* untuk Gejala dan Penyakit / CF(E,e) 70
     5. Rekomendasi Pakar 71
  6. Perancangan Sistem 73
     1. Perancangan UML 73
     2. Perancangan Basis Data Aplikasi Mobile 81
     3. Perancangan *User* *Interface* 82

**BAB IV IMPLEMENTASI DAN PEMBAHASAN** 87

* 1. Implementasi 87
     1. Manual Instalasi 87
     2. Implementasi *Interface* 90
  2. Uji Coba Sistem dan Program 97
     1. *Black*-*box* *Testing* 99
     2. *White*-*box* *Testing* 99
     3. Kesalahan Kode Program (*Syntax* *Error*) 99
     4. Kesalahan Proses (*Run* *Time* *Error*) 100
     5. Kesalahan Logika 100
  3. Pengujian Hasil Diagnosa 100
  4. Implementasi Kode Program 103

**BAB V PENUTUP** 105

5.1 Kesimpulan 105

5.2 Saran 105

**DAFTAR PUSTAKA** 107

**LAMPIRAN**

**DAFTAR TABEL**

Tabel 2.1 Perbandingan Jurnal Terkait 10

Tabel 2.2 Simbol *Use* *Case* *Diagram* 48

Tabel 2.3 Simbol *Class* *Diagram* 50

Tabel 2.4 Simbol *Sequence* *Diagram* 51

Tabel 2.5 Simbol *Activity* *Diagram* 52

Tabel 3.1 Strategi SWOT 58

Tabel 3.2 Jenis Penyakit Paru-Paru 66

Tabel 3.3 Gejala Penyakit Paru-Paru 67

Tabel 3.4 Relasi Kode Gejala dan Kode Penyakit 68

Tabel 3.5 Nilai CF Rule 69

Tabel 3.6 Nilai CF User 70

Tabel 3.7 Rekomendasi Pakar 71

Tabel 3.8 Rancangan Struktur Tabel Gejala 118

Tabel 3.9 Rancangan Struktur Tabel Penyakit 118

Tabel 3.10 Rancangan Struktur Tabel CF Rule 119

Tabel 3.11 Rancangan Struktur Tabel CF Pakar 119

Tabel 4.1 *Black*-*box* *Testing* Menu Utama 98

Tabel 4.2 *Black*-*box* *Testing* Menu Mulai Diagnosa 98

Tabel 4.3 *Black*-*box* *Testing* Menu Info Penyakit 98

Tabel 4.4 Pengujian Hasil Diagnosa 100

**DAFTAR GAMBAR**

Gambar 2.1 Arsitektur Sistem Pakar 26

Gambar 2.2 Arsitektur Android 40

Gambar 2.3 Logo Android 4.0 (*Ice* *Cream* *Sandwich*/ICS) 41

Gambar 2.4 Logo Android 4.1 / 4.2 / 4.3 (*Jelly* *Bean*) 42

Gambar 2.5 Logo Android 4.4 (*Kit* *Kat*) 43

Gambar 2.6 Logo android 5.0 (*Lollipop*) 44

Gambar 2.7 Contoh Penggambaran ERD 46

Gambar 2.8 Simbol Dasar Notasi Diagram E-R Dasar 47

Gambar 2.9 Derajat Relasi Notasi Diagram E-R Dasar 47

Gambar 2.10 Kardinalitas Relasi Notasi Diagram E-R Dasar 47

Gambar 3.1 *Use* *Case* *Diagram* Aplikasi Android 74

Gambar 3.2 *Activity* *Diagram* *Splash* *Screen* 75

Gambar 3.3 *Activity* *Diagram* Menu Mulai Diagnosa 76

Gambar 3.4 *Activity* *Diagram* Menu Info Penyakit 77

Gambar 3.5 *Activity* *Diagram* Menu Tentang 78

Gambar 3.6 *Class* *Diagram* 79

Gambar 3.7 *Sequence* *Diagram* *Splash* *Screen* 79

Gambar 3.8 *Sequence* *Diagram* Menu Mulai Diagnosa 80

Gambar 3.9 *Sequence* *Diagram* Menu Info Penyakit 80

Gambar 3.10 *Sequence* *Diagram* Menu Tentang 81

Gambar 3.11 Rancangan Relasi Antar Tabel Aplikasi Android 81

Gambar 3.12 Desain *Interface* *Splash* *Screen* 83

Gambar 3.13 Desain *Interface* Menu Utama 83

Gambar 3.14 Desain *Interface* Mulai Diagnosa 84

Gambar 3.15 Desain *Interface* Hasil Diagnosa 85

Gambar 3.16 Desain *Interface* Menu Info Penyakit 85

Gambar 3.17 Desain *Interface* Info Detail Penyakit 86

Gambar 3.18 Desain *Interface* Menu Tentang 86

Gambar 4.1 Manual Instalasi 88

Gambar 4.2 Manual Instalasi Kedua 88

Gambar 4.3 Manual Instalasi Ketiga 89

Gambar 4.4 Manual Instalasi Keempat 89

Gambar 4.5 *Interface* *Splash* *Screen* 90

Gambar 4.6 *Interface* Menu Utama 91

Gambar 4.7 *Interface* Menu Mulai Diagnosa 92

Gambar 4.8 *Interface* Hasil Diagnosa 93

Gambar 4.9 *Interface* Menu Info Penyakit 94

Gambar 4.10 *Interface* Detail Penyakit 95

Gambar 4.11 *Interface* Menu Tentang 96

Gambar 4.12 *Interface* Menu Keluar 97

Gambar 4.13 *White*-*box* *Testing* Aplikasi Mobile 99

**DAFTAR LAMPIRAN**

Lampiran 1 Surat Pernyataan Wawancara Dokter Sebagai Pakar 1

Lampiran 2 Perhitungan Manual Nilai *Certainty* *Factor* 2

Lampiran 3 Rancangan Struktur Tabel 9

Lampiran 4 Implementasi Kode Program Aplikasi *Mobile* 11

**INTISARI**

Sistem pakar adalah cabang kecerdasan buatan yang menggunakan pengetahuan/ *knowledge* khusus untuk memecahkan masalah pada level *human* *expert*/pakar. Salah satu penerapan sistem pakar dalam bidang kesehatan adalah untuk melakukan diagnosa penyakit pada paru-paru. Pada makalah ini dilakukan perancangan dan pembuatan sistem pakar yang digunakan untuk membantu menentukan diagnosa suatu penyakit yang diawali dari gejala utama penyakit pada paru-paru untuk menentukan saran atau solusi pengobatan kepada pengguna.

Masalah ketidakpastian pengetahuan dalam sistem pakar ini diatasi dengan menggunakan metode probabilitas Bayesian. Proses penentuan diagnosa dalam sistem pakar ini diawali dengan sesi konsultasi, dimana sistem akan mengajukan pertanyaan-pertanyaan yang relevan kepada petani sesuai gejala utama penyakit tanaman buah nanas.

Hasil akhir dari penelitian ini adalah sebuah sistem pakar untuk melakukan diagnosa penyakit pada tanaman buah nanas beserta nilai probabilitas dari penyakit hasil diagnosa, yang menunjukkan tingkat kepercayaan sistem terhadap penyakit tersebut dan saran atau solusi pengobatan kepada para petani buah nanas.

**Kata-kunci :** Sistem pakar, Metode *Certainty Factor*, diagnosa, paru-paru

***ABSTRACT***

*An expert system is a branch of artificial intelligence that uses knowledge / specialized knowledge to solve problems on a human level expert / specialist. One application of expert systems in agriculture is for diagnosing diseases in plants. In this paper carried out the design and manufacture of an expert system that is used to help diagnose a disease that begins from the main symptoms of the disease in pineapple plants to determine any advice or treatment solutions to farmers pineapple.*

*The uncertainty of knowledge in an expert system is overcome by using the method of Bayesian probability. The process of determining diagnoses in this expert system begins with the consultation session, where the system will ask the questions that are relevant to the farmers of the main symptoms of the disease pineapple plants.*

*The end result of this research is an expert system for diagnosing diseases of the pineapple plants showing the probability of disease diagnosis, which indicates the level of confidence the system against the disease and treatment suggestions or solutions to farmers pineapple.*

***Keywords****: Expert system, Certainty Factor Method, diagnoses, lungs.*

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Polusi udara di bumi yang semakin hari semakin tercemar mempunyai dampak yang negatif, terutama terhadap kesehatan manusia. Polusi udara yang terjadi pada saat ini menyebabkan munculnya berbagai macam jenis penyakit, terutama pada paru-paru manusia. Lemahnya daya tahan tubuh manusia, membuat manusia rentan terhadap penyakit. Padatnya aktifitas yang dilakukan manusia di luar rumah menyebabkan manusia mudah terkena penyakit khususnya pada paru-paru. Terlebih lagi kualitas udara yang semakin menurun sangat berbahaya bagi kesehatan.

Kurangnya pelayanan kesehatan dan keterbatasan ekonomi masyarakat untuk konsultasi langsung ke dokter menjadi penyebab utama keterlambatan penanganan kesehatan. Kebutuhan informasi yang cepat dari dokter spesialis paru-paru sangatlah dibutuhkan, akan tetapi masyarakat tidak mampu berkonsultasi dengan dokter spesialis dikarenakan biaya yang cukup mahal. Oleh karena itulah, banyak penderita paru-paru yang tidak dapat mendapatkan pelayanan medis dengan cepat dan tepat.

Gejala penyakit yang muncul tiba-tiba harus ditangani dengan cepat dan tepat. Terdapat permasalahan yang dapat menyebabkan tingginya tingkat kematian, yaitu terlambatnya penanganan pasien, sehingga penyakit makin memburuk. Penangan yang tepat dan cepat dapat ditangani melalui alat bantu yang dapat mengetahui penyakit dari gejala-gejala yang diderita atau dirasakan dan dapat memberikan solusi penanganan awal dalam mengatasi penyakit yang di derita.

Perkembangan teknologi saat ini dapat dimanfaatkan untuk mendekteksi gejala-gejala secara dini penyakit paru-paru dengan memanfaatkan konsep sistem pakar. Dengan menggunakan sistem pakar untuk mendiagnosis penyakit paru-paru, masyarakat dapat mengenali gejala-gejala penyakit paru-paru dengan lebih cepat, mudah, murah, dan dapat diakses kapan pun dimana pun karena sistem pakar ini berbasis android.

Berdasarkan permasalahan yang muncul, dalam penelitian ini dibuat suatu sistem pakar dengan judul **“PERANCANGAN APLIKASI SISTEM PAKAR METODE CERTAINTY FACTOR UNTUK PENYAKIT PARU-PARU BERBASIS ANDROID”** yang dapat membantu pasien atau *user* dalam mendiagnosa gejala penyakit paru-paru yang diderita. Adapun *output* yang akan dihasilkan dari sistem ini adalah jenis penyakit dan cara mengatasi penyakit yang diderita.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang yang diangkat, maka dibuat rumusan masalah yang akan diselesaikan dalam penelitian ini adalah “Bagaimana merancang suatu aplikasi sistem pakar yang dapat mendiagnosa penyakit paru-paru dengan menggunakan metode *certainty factor yang* di aplikasikan pada android?”

* 1. **Batasan Masalah**

Untuk membatasi pembatasan agar terfokuskan pada aspek yang diangkat sebagai pembahasan utama dalam penelitian ini, maka perlu dibuat adanya batasan-batasan masalah, antara lain:

1. Pembuatan sistem pakar ini hanya mendiagnosa penyakit Asma, Bronkiolitis, dan TB (*Tuberculosis*) dengan mencocokan gejala yang ada dan mendapatkan hasil dari pencocokan yang sudah dilakukan.
2. Menggunakan metode *Certainty Factor* untuk perhitungan dan kepastiannya.
3. Terdapat 23 macam gejala penyakit pada aplikasi sistem pakar ini.
4. Referensi tentang penyakit paru-paru diperoleh dari literatur/buku medis, internet, dan pakar/dokter.
5. Hasil akhir diagnosa menampilkan definisi penyakit, gejala penyakit, penyebab, pencegahan, tindakan umum, pengobatan.
6. Hasil diagnosa bersifat deteksi awal dan masih dibutuhkan pemeriksaan lebih lanjut oleh dokter untuk memastikan penyakit secara tepat.
7. Sistem pakar yang dibuat berbasis android.
8. Database yang digunakan adalah SQLite.
9. Dijalankan pada Android versi 4.0 *Ice Cream Sandwich* sampai Android versi 5.0 *Lollipop.*
   1. **Maksud dan Tujuan Penelitian**

Tujuan dari pembuatan aplikasi sistem pakar ini adalah sebagai berikut:

1. Merancang dan membangun aplikasi sistem pakar yang dapat membantu peran seorang pakar/dokter dalam mengetahui penyakit paru-paru berdasarkan gejala-gejala yang dirasakan dengan cepat dan tepat berbasis android dengan menggunaan metode *certainty factor*.
2. Memudahkan *user* dalam mendiagnosa penyakit paru-paru yang diderita.
   1. **Metode Penelitian**

Dalam penelitian ini data-data yang digunakan didapat dari beberapa metode, antara lain:

* + 1. **Metode Pengumpulan Data**

Dalam penelitian ini data-data yang digunakan di dapat dari beberapa metode, antara lain:

1. Metode Wawancara

Dalam metode ini, pengumpulan data yang dilakukan adalah mewawancarai dokter.

1. Metode Observasi

Dalam metode ini, dilakukan observasi pada pederita penyakit asma, bronkiolitis, dan TB (*tuberculosis*) untuk pengumpulan data-data yang akan dibutuhkan oleh sistem.

1. Metode Studi Literatur

Pengumpulan data dalam metode ini dilakukan dengan cara mempelajari dan memahami berbagai literatur seperti buku, jurnal ilmiah, situs-situs internet dan berbagai bahan ain yang berkaitan dengan topik penelitian.

* + 1. **Metode Analisis**

Merupakan tahapan dalam menganalisis atau mendefinisikan permasalahan yang akan dibangun. Adapun metode analisis yang digunakan adalah sebagai berikut:

1. Analisis SWOT yang terdiri dari *Strength* (Kekuatan), *Weakness* (Kelemahan), *Opportunity* (Kesempatan), dan *Threats* (Ancaman).
2. Analisis kebutuhan sistem yang terdiri dari analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.
3. Analisis kelayakan sistem yang terdiri dari analisis kelayakan teknologi, analisis kelayakan operasional, analisis kelayakan ekonomi, dan analisis kelayakan hukum.
   * 1. **Metode Perancangan**

Merupakan tahapan dalam merancang proses yang terjadi pada sistem, serta relasi yang terdapat dalam *Database*. Perancangan UML (*Unified* *Modeling* *Languange*) untuk memvisualisasikan proses yang terjadi pada sistem dan merancang *interface*, untuk membuat tampilan sistem bagi *user*.

* + 1. **Metode Pengembangan**

Metode pengembangan yang digunakan dalam penelitian ini adalah metode SDLC. Metode SDLC merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya, inti dari metode SDLC adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear, dimulai dari analisis kebutuhan sistem, perancangan sistem, pembuatan sistem, pengujian sistem, dan pemeliharaan sistem.

* + 1. **Metode Testing**

Merupakan tahapan untuk menguji coba sistem aplikasi. Ada dua jenis pengujian yang dilakukan yaitu sebagai berikut:

1. Pengujian *White box testing*, yaitu pengujian per modul.
2. Pengujian *Black box testing*, yaitu pengujian secara terintegrasi.
   * 1. **Metode Implementasi**

Tahap implementasi merupakan tahap dimana sistem telah melewati proses pengujian dan dinyatakan bekerja sesuai fungsinya dan layak digunakan oleh pengguna.

* 1. **Sistematika Penulisan**

Sistematika penulisan bertujuan untuk mempermudah dalam penulisan laporan skripsi. Adapun sitematika penulisan pada laporan perancangan aplikasi sitem pakar ini adalah sebagai berikut:

|  |  |
| --- | --- |
| **BAB I** | **PENDAHULUAN** |
|  | Bab ini berisi merupakan pengantar pokok permasalahan dan gambaran penelitian secara keseluruhan, adapun hal-hal yang dibahas adalah latar belakang, rumusan masalah, batasan masalah, maksud dan tujuan penelitian, metode penelitian, dan sistematika penulisan. |
| **BAB II** | **LANDASAN TEORI** |
|  | Bab ini berisi tentang konsep dasar serta teori-teori yang berkaitan dengan topik penelitian dari sumber pustaka dan refrensi yang menjadi landasan dasar dalam perancangan, analisis kebutuhan sampai implementasi dan pengujian sistem. |
| **BAB III** | **ANALISIS DAN PERANCANGAN SISTEM** |
|  | Bab ini berisi tentang analisis kebutuhan dalam membangun aplikasi ini, analisis sistem yang sedang berjalan pada aplikasi ini sesuai dengan metode pembangunan perangkat lunak yang digunakan. Selain itu terdapat juga perancangan antar muka untuk aplikasi yang akan dibangun sesuai dengan analisis yang telah dibuat. |
| **BAB IV** | **IMPLEMENTASI DAN PERANCANGAN** |
|  | Bab ini berisi tahapan implementasi dan pengujian yang merupakan tahap yang dilakukan dalam mengimplementasikan dari hasil penelitian, analisis dan perancangan yang telah diidentifikasikan untuk mengimplementasikan dan menguji aplikasi. |
| **BAB V** | **PENUTUP** |
|  | Bab ini berisi tentang kesimpulan yang diambil dari hasil penulisan dan saran yang dapat menjadi masukan bagi perusahaan dan penelitian selanjutnya. |

**BAB II**

**LANDASAN TEORI**

* 1. **Tinjauan Pustaka**

Andri Saputra (2011), adalah mahasiswa Jurusan Teknik Informatika, STMIK PalComTech Palembang dalam jurnalnya yang berjudul *Sistem Pakar Identifikasi Penyakit Paru-Paru Pada Manusia Menggunakan Pemrograman Visual Basic 6.0*. Dalam jurnalnya yang ditulis oleh Andri Saputra tersebut membahas tentang sistem pakar untuk diagnosa penyakit paru-paru yang diterapkan di Dinas Kesehatan Rumah Sakit Khusus Paru-Paru Palembang. Aplikasi sistem pakar ini akan digunakan pada puskesmas, serta rumah sakit yang membutuhkan. Dalam jurnal tersebut berisi tentang perancangan sistem pakar dengan menggunakan metode *forward chaining*, tetapi dalam aplikasi sistem pakar ini hanya bisa digunakan pada komputer. Hasil akhir dari jurnal tersebut berupa aplikasi sistem pakar untuk mendiagnosis penyakit paru-paru berbasis desktop.

Geovani, Dadang Syarif SS, & Mardhiah Fadhli (2012), adalah mahasiswa Sistem Informasi dan Teknik Informatika, Politeknik Caltex Riau dalam jurnalnya yang berjudul *Sistem Pakar Diagnosa Penyakit Paru Anak Berbasis J2ME*. Dalam jurnalnya yang ditulis oleh Geovani, Dadang Syarif dan Mardhiah Fadhli tersebut berisi tentang perancangan sistem pakar dengan menggunakan metode *forward chaining* dan metode *Depth First Search*, tetapi aplikasi ini hanya dapat digunakan pada anak-anak dan belum bisa digunakan di android. Hasil akhir dari jurnal tersebut berupa aplikasi sistem pakar untuk mendiagnosis penyakit pada anak berbasis J2ME.

Firmansyah Putra adalah mahasiswa jurusan teknik informatika UIN dalam skripsinya yang berjudul “Perancangan Sistem Pakar Identifikasi Penyakit Paru-Paru Menggunakan Metode Forward Chaining”. Dalam skripsi yang ditulis oleh Firmansyah Putra tersebut membahas tentang sistem pakar untuk diagnosa penyakit paru-paru dalam skripsi tersebut berisi tentang perancangan sistem pakar dengan menggunakan metode forward chaining tetapi info penyakit dan gejala yang terdapat dalam sistem pakar masih sangat terbatas. Hasil akhir dari skripsi tersebut berupa aplikasi sistem pakar untuk mendiagnosis penyakit paru-paru berbasis website.

**Tabel 2.1 Perbandingan Jurnal Terkait**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Judul** | **Peneliti, Tahun** | **Hasil** | **Pembeda** |
| 1. | Sistem Pakar Identifikasi Penyakit Paru-Paru Pada Manusia Menggunakan Pemrograman Visual Basic 6.0 | Andri Saputra (2011) | Sistem pakar untuk mendiagnosis penyakit paru-paru berbasis desktop yang bisa digunakan di rumah sakit atau puskesmas | Memberikan informasi tentang penyakit paru-paru menggunakan metode forward chaining dan menggunakan pemrograman visual basic 6.0 |
| **No** | **Judul** | **Peneliti, Tahun** | **Hasil** | **Pembeda** |
| 2. | Sitem Pakar Diagnosa Penyakit Paru pada Anak Berbasis J2ME | Miranda Geovani, Dadang Syarif SS,S.Si,M.Sc, dan Mardhiah Fadhli, S.T (2012) | Sistem pakar untuk mendiagnosa penyakit paru pada anak-anak | Memberikan hasil informasi tentang dan diagnosis penyakit paru pada anak menggunakan metode forward chaining dan metode depth first search. |
| 3. | Perancangan Sistem Pakar Identifikasi Penyakit Paru-Paru Menggunakan Metode Forward Chaining | Firmansyah Putra (2011) | Sistem Pakar identifikasi penyakit paru-paru dan pernafasan berbasis website. | Memberikan hasil infomasi penyakit paru-paru dan pernafasan menggunakan metode forward chaining. |

* 1. **Dasar Teori**
     1. **Paru-Paru**
        1. **Pengertian Paru-Paru**

Paru-paru sangat penting bagi tubuh manusia, sebab salah satu fungsi paru-paru adalah memasukkan oksigen dan mengeluarkan karbondioksida ketika tubuh menghirup udara. [1]

* + - 1. **Jenis Penyakit Paru-Paru**

Untuk lebih jelasnya mengenai gejala dan pengobatan jenis penyakit diatas akan dibahas pada sub bab dibawah ini.

1. **Asma (*Asthma*)**

Asma adalah penyempitan bronkus yang bersifat reversibel yang terjadi oleh karena bronkus yang hiperaktif mengalami kontaminasi dengan antigen. Autopsi yang dilakukan pada penderita yang dilakukan pada penderita yang mati karena asma yang menjadi masalah pokok adalah bukan saja bronkospasme dari otot, akan tetapi juga adanya edema dan penuhnya mukus di intraluminal dari bronkus yang menyebabkan jalan napas menjadi tersumbat. Dari pemeriksaan mukus diperlihatkan banyak eosinifil, sementara itu limfosit T dan sel epitel telah mengalami kerusakan. [2]

* + 1. **Gejala**

Adapun gejala-gejala dari asma, adalah: [3]

1. Dada sesak
2. Kesulitan bernapas
3. Terdengar suara (mengi) sewaktu mengeluarkan napas
4. Batuk, terutama malam hari, karena udara dingin
5. Terdapat sedikit dahak
6. Napas pendek dan sering lebih nyaman jika duduk
7. Otot pada leher menegang

Gejala lanjut yang berat :

1. Kulit kebiru-biruan
2. Kelelahan
3. Napas berbunyi
4. Sulit berbicara
5. Gangguan kesadaran, termasuk gelisah dan kebingungan
   * 1. **Penyebab**

Penyempitan saluran pernapasan (bronkus dan bronkiolus), diikuti pembengkakan saluran dan bertambah banyaknya secret paru-paru (riak). Hal ini menyumbat saluran udara ke paru. Perubahan ini disebabkan oleh: [3]

1. Allergen, seperti serbuk bunga, debu, jamur dan beberapa makanan.
2. Infeksi paru seperti bronchitis.
3. Iritasi udara seperti asap rokok dan bau-bauan.
4. Stress
   * 1. **Pencegahan** [3]
5. Hindari allergen dan polutan udara yang sudah diketahui.
6. Gunakan obat untuk pencegahan secara teratur – jangan berhenti walaupun sudah sembuh.
   * 1. **Tindakan** [3]
7. Hindari allergen dan iritan di rumah maupun di kantor jika mungkin.
8. Bawalah selalu obat bersama anda. Mintalah pada dokter obat untuk keadaan gawat.
9. Duduk tegak disaat serangan datang.
10. Berlatihlah menarik napas panjang setiap pagi untuk mengurangi penimbunan secret paru.
    * 1. **Pengobatan**

Dokter akan memberikan: [3]

1. Ekspektoran untuk mencairkan dahak
2. Bronkodilator untuk melebarkan saluran pernapasan.
3. Kortison intravena (hanya pada keadaan gawat darurat) untuk mengurangi raksi alergi tubuh.
4. Kortison dengan cara disemprot ke hidung, yang ternyata memiliki efek samping yang lebih kecil jika dibandingkan dengan yang dimakan.
5. Sodium kromoglikat juga disemprot. Ini merupakan obat pencegahan.
6. **TBC (Tuberculosis)**

Tuberkulosis paru adalah penyakit yang disebabkan oleh Mycobacterium tuberculosis, yakni kuman aerob yang dapat hidup terutama di paru atau di berbagai organ tubuh yang lainnya yang mempunyai kandungan lemak yang tinggi pada membran selnya sehingga menyebabkan bakteri ini menjadi tahan terhadap asam dan pertumbuhan dari kumannya berlangsung dengan lambat. Bakteri ini tidak tahan terhadap ultraviolet, karena itu penularannya terutama terjadi pada malam hari. [4]

1. **Gejala**

Adapun gejala-gejala dari asma, adalah: [5]

1. Demam suhu rendah (sumer-sumer)
2. Penurunan berat badan
3. Berkeringat yang banyak, terutama di malam hari

Tahap lanjut:

1. Batuk dengan dahak yang lama-lama bercampur dengan darah, kuning, kental atau putih
2. Nyeri dada
3. Sesak napas
4. **Penyebab**

Infeksi oleh kuman, yang bernama mycobacterium tuberculosis. Kuman ini ditularkan melalui percikan dahak dan udara dari penderita ke orang lain. [5]

1. **Pencegahan**

Infeksi oleh kuman, yang bernama mycobacterium tuberculosis. Kuman ini ditularkan melalui percikan dahak dan udara dari penderita ke orang lain. [5]

1. **Tindahakan** [5]
2. Membawa penderita teberkulosis ke rumah sakit yang terdapat pelayanan DOTS.
3. Pengobatan “Obat Anti Tuberculosis” (OAT). Kuman akan meluas setelah pengobatan OAT dilakukan.
4. Kadangkala anda perlu mengecek dahak pada pagi hari selama 3 hari, untuk melihat apakah TBC masih aktif.
5. **Pengobatan**

Menggunakan Obat Anti TBC (OAT) dengan memerlukan waktu yang cukup lama yaitu 6-8 bulan yang harus dilakukan secara terus menerus dan tuntas hingga sembuh. Obat Anti TBC diantaranya yaitu Isoniazid, Rifampisin, Pyrazinamide, Streptomycin, dan Ethambutol. OAT pada umunya memberatkan ginjal dan hati. Kondisi ini sangat memberatkan penderita TBC, karena fungsi detoksikasi dan fungsi pembuangan racun mengalami keterlambatan atau hambatan, sehingga kerja OAT untuk membunuh bakteri tbc terhambat akibat memburuknya kondisi penderita.

Tahapan pengobatan TBC secara medis terbagi menjadi tiga kategori dan pada umunya, pasien melakukan pengobatan pada kategori 1 dimana akan dilakukan 2 tahap atau 2 fase yaitu pada fase intensif penderita harus minum obat empat jenis oat selama paling tidak 2 bulan untuk membantu mengubah terjadinya infeksi yang tidak aktif dan tidak akan menular lagi. [4]

1. **Bronkiolitis**

Bronkiolitis adalah infeksi saluran respiratorik bawah yang disebabkan virus, biasanya lebih berat pada bayi muda, terjadi epidemik setiap tahun dan ditandai dengan obstruksi saluran pernapasan dan wheezing. Penyebab paling sering adalah Respiratory syncytial virus. Infeksi bakteri sekunder bisa terjadi dan biasa terjadi pada keadaan tertentu. Penatalaksanaan bronkiolitis, yang disertai dengan napas cepat atau tanda lain distres pernapasan, sama dengan pneumonia. Episode wheezing bisa terjadi beberapa bulan setelah serangan bronkiolitis, namun akhirnya akan berhenti. [6]

1. **Gejala**

Adapun gejala-gejala dari Bronkiolitis, adalah: [7]

1. Kesulitan bernapas
2. Biasanya didahului dengan batuk ringan
3. Mengi (bisa ada bisa tidak)
4. Pernapasan yang cepat dan dangka (60-80 kali per menit)
5. Retraksi dada (gerak gergaji)
6. Demam Kulit atau kuku biru (kasus berat)
7. **Penyebab**

Infeksi virus atau bakteri, atau kombinasi dari keduanya. Beberapa anak muda mengalami gangguan ini setelah terkena demam. Bronkiolitis menular dan sering menjadi epidemic. [7]

1. **Pencegahan** [7]
2. Pasang pengatur kelembapan di ruang tidur anak. Pasang alat ini setiap malam selama dan sesudah infeksi pernapasan yang mengenai anak yang rentan terhadap bronkiolitis.
3. Amati dan hindari setiap aktivitas yang kelihatannya dapat memicu serangan pada anak, seperti bermain di udara malam yang dingin.
4. Kurangi kontak antara anak dengan orang lain, khususnya anak lain untuk menghindari tertular demam.
5. **Tindakan**

Pasang pengatur kelembapan pada ruang anak setinggi mungkin, khususnya pengatur kelembapan dingin. Bila anda tidak mempunyai pengatur suhu, isi bak mandi dengan air panas atau dingin dan tutup pintu serta jendela untuk mendapatkan ruang dengan kelembapan tinggi. Biarkan anak di dalam ruang tersebut selama 20 menit beberapa kali sehari, khusunya sebelum tidur. Bila anak terbangun di malam hari akibat mengi atau sesak napas, ulangi proses tersebut. [7]

1. **Pengobatan**

Dokter akan memberikan antibiotik untuk melawan infeksi bakteri. [7]

* + 1. **Kecerdasan Buatan (*Artificial Intelligence*)** 
       1. **Pengertian Kecerdasan Buatan**

*Artificial Intelligence* (AI) atau kecerdasan buatan dapat didefinisikan sebagai cabang sains komputer yang mempelajari otomatis tingkah laku cerdas (*intelligent*). Kecerdasan buatan harus didasarkan pada prinsip-prinsip teoretikal dan terapan yang menyangkut struktur data yang digunakan dalam representasi pengetahuan (*knowledge representation*), algoritma yang diperlukan dalam penerapan pengetahuan itu, serta teknik-teknik bahasa dan pemrograman yang dipakau dalam implementasinya.

Kecerdasan buatan (AI) sebagai sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia. Ada tiga tujuan kecerdasan buatan, yaitu: [8]

1. Membuat komputer lebih cerdas
2. Mengerti kecerdasan
3. Membuat mesin lebih berguna
   * + 1. **Bidang penelitian dalam Kecerdasan Buatan**

Ada beberapa bidang penelitian yang berkaitan dengan kecerdasan buatan, diantaranya: [9]

1. *Formal task* (matematika, *games*)
2. *Mundane task* (*perception*, *robotics*, *natural language common sense*, *reasoning*)
3. *Expert task* (*financial analysis*, *medical diagnotic*, *engineering scientific analysis*, dan lain-lain)
   * + 1. **Konsep Kecerdasan Buatan**

Ada beberapa konsep yang harus dipahami dalam kecerdasan buatan, diantaranya: [9]

1. Turing Tesk – Metode Pengujian Kecerdasan.
2. Pemprosesan Simbolik.
3. Konsep *Heuristic*.
4. Penarikan Kesimpulan (*Inferencing*).
5. Pencocokan Pola (*Patern Matching*).
   * + 1. **Perbandingan Kecerdasan Buatan dengan Kecerdasan Alamiah**

Kelebihan kecerdasan buatan dibandingkan kecerdasan alamiah: [9]

1. Lebih permanen.
2. Memberikan kemudahan dalam duplikasi dan penyebaran.
3. Relatif lebih murah dari kecerdasan alamiah.
4. Konsisten dan teliti.
5. Dapat didokumentasi.
6. Dapat mengerjakan task dengan lebih cepat dan lebih baik dibanding manusia.

Kelebihan kecerdasan alamiah dibanding kecerdasan buatan: [9]

1. Bersifat lebih kreatif.
2. Dapat melakukan proses pembelajaran secara langsung, sementara AI harus mendapatkan masukan berupa simbol dan representasi- representasi.
3. Menggunakan fokus yang luas sebagai referensi untuk pengambilan keputusan. Sebaliknya AI menggunakan fokus yang sempit.
   * 1. **Sistem Pakar**
        1. **Pengertian Sistem Pakar**

Sistem pakar adalah salah satu cabang AI yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. [10]

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecah yang diamaksud antara lain: pembuatan keputusan (*decicion* *making*), oemaduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar. [11]

* + - 1. **Pemakaian Sistem Pakar**

Sistem pakar dapat digunakan oleh: [11]

1. Orang awam yang bukan pakar untuk meningkatkan kemampuan mereka dalam memecahkan masalah.
2. Pakar sebagai asisten yang berpengetahuan.
3. Memperbanyak atau menyebarkan sumber pengetahuan yang semakin langka.

Sistem pakar merupakan program yang dapat menggantikan keberadaan seorang pakar. Alasan mendasar mengapa *Expert System* dikembangkan untuk menggantikan seorang pakar: [11]

1. Dapat menyediakan kepakaran setiap waktu dan diberbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menghadirkan atau menggunakan jasa seorang pakar memerlukan biaya mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*hostile environtment*).
   * + 1. **Ciri-ciri Sistem Pakar**

Ciri-ciri dari sistem pakar adalah sebagai berikut: [11]

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikan dengan cara yang dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Outputnya bersifat nasihat atau anjuran.
7. Output tergantung dari dialog dengan *user*.
8. *Knowledge* *base* dan *inference* *engine* terpisah.
   * + 1. **Keuntungan Pemakaian Sistem Pakar**

Banyak keuntungan yang diberikan oleh sistem pakar, diantaranya: [11]

1. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
3. Meningkatkan *output* dan produktifitas. *Expert* *System* dapat bekerja lebih cepat dari manusia. Keuntungan ini berarti mengurangi jumlah pekerja yang dibutuhkan, dan akhirnya akan mereduksi biaya.
4. Meningkatkan kualitas.
5. *Expert* *System* menyediakan nasihat yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena ES dapat melatih pekerja yang tidak berpengalaman.
7. Handal (*realibility*).
8. ES tidak dapat lelah atau bosan. Juga konsisten dalam memberi jawaban dan atau selalu memberikan perhatian penuh.
9. Memiliki kemampuan untuk memecahkan masalah yang kompleks.
10. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai dimana saja. Merupakan arsip yang terpercaya dari sebuah keahlian sehingga user seolah-olah berkonsultasi langsung dengan sang pakar meskipun mungkin sangpakar sudah pensiun.
    * + 1. **Arsitektur Sistem Pakar**

Menurut Martin dan Oxman (1988), sistem pakar memiliki beberapa komponen utama, yaitu antarmuka pengguna (*user interface*), basis data sistem pakar (*expert* *system* *database*), fasilitas akuisisi pengetahuan (*knowledge* *acquisition* *facility*), dan mekanisme inferensi (*inference* *mechanism*). Selain itu ada satu komponen yang hanya ada pada beberapa sistem pakar, yaitu fasilitas penjelasan (*explanation* *facility*).

Antarmuka pengguna adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. Basis data sistem pakar berisi pengetahuan setingkat pakar pada subjek tertentu. Berisi pengetahuan yang dibutuhkan untuk memahami, merumuskan, dan menyelesaikan masalah. Basis data ini terdiri atas 2 elemen dasar: [11]

1. Fakta, sesuai masalah dalam teori terkait.
2. Heuristik khusus atau *rules*, yang langsung menggunakan pengetahuan untuk menyelesaikan masalah khusus.

Pengetahuan ini dapat bersalah dari pakar, jurnal, majalah, dan sumber pengetahuan lain. Fasilitas akuisisi pengetahuan merupakan perangkat lunak yang menyediakan failitas dialog antara pakar dengan sistem. Fasilitas akuisisi ini digunakan untuk memasukkan fakta-fakta dan kaidah-kaidah sesuai dengan perkembangan ilmu. Meliputi proses pengumpulan, pemindahan, dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber terdokumentasi ke program computer, yang bertujuan untuk memperbaiki dan atau mengembangkan basis pengetahuan (*knowledge base*).

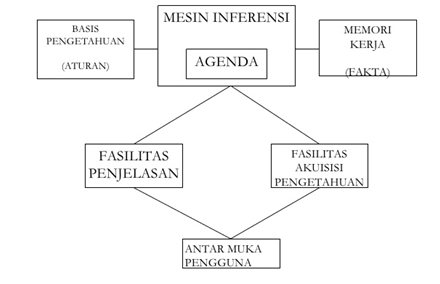
Mekanisme inferensi merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan yang ada untuk menghasilkan suatu kesimpulan atau hasil akhir. Dalam komponen ini dilakukan permodelan proses berfikir manusia.

Fasilitas penjelasan berguna dalam memberikan penjelasan kepada pengguna mengapa computer meminta suatu informasi tertentu dari pengguna dan dasar apa yang digunakan komputer sehingga dapat menyimpulkan suatu kondisi.

Ada 4 tipe penjelasan yang digunakan dalam sistem pakar, yaitu (Schnupp, 1989): [11]

1. Penjelasan mengenai jejak aturan yang menunjukan status konsultasi.
2. Penjelasan mengenai bagaimana sebuah keputusan diperoleh.
3. Penjelasan mengapa sistem menanyakan suatu pertanyaan.
4. Penjelasan mengapa sistem tidak memberikan keputusan seperti yang dikehendaki pengguna.

Arsitektur dasar dari sistem pakar dapat dilihat pada Gambar 2.1 (Giarratano dan Riley, 1994)



**Gambar 2.1** Arsitektur Sistem Pakar

Memori kerja dalam arsitektur sistem pakar merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi, berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi. [11]

* + - 1. **Orang Yang Terlibat Dalam Sistem Pakar**

Untuk memahami perancangan sistem pakar, perlu dipahami mengenai siapa saja yang berinteraksi dengan sistem. Mereka adalah: [11]

1. Pakar (*domain expert*): seorang ahli yang dapat menyelesaikan masalah yang sedang diusahakan untuk dipecahkan oleh sistem.
2. Pembangunan pengetahuan (*knowledge engineer*): seseorang yang menerjemahkan pengetahuan seorang pakar dalam bentuk deklaratif sehingga dapat digunakan oleh sistem pakar.
3. Pengguna (*user*): seseorang yang berkonsultasi dengan sistem untuk mendapatkan saran yang disediakan oleh pakar.
4. Pembangunan sistem (*system engineer*): seorang yang membuat antarmuka pengguna, merancang bentuk basis pengetahuan secara deklaratif dan mengimplementasikan mesin inferensi.
   * + 1. **Kategori Masalah Sistem Pakar**

Masalah-masalah yang dapat diselesaikan dengan sistem pakar, diantaranya: [11]

1. Interpretasi: membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
2. Prediksi: memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.
3. Diagnosis: menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati.
4. Desain: menentukan konfigurasi komponen-komponen sistem yang
5. Pengendalian: mengatur tingkah laku suatu environment yang kompleks.
6. *Selection*: mengidengtifikasi pilihan terbaik dari sekumpulan (list) kemungkinan.
7. *Simulation*: permodelan interaksi antara komponen-komponen sistem.
8. *Monitoring*: membandingkan hasil pengamatan dengan kondisi yang diharapkan.
   * 1. ***Certainty Factor* (Faktor Kepastian)**

Faktor kepastian (*certainty* *factor*) diperkenakan oleh Shorthliffe Buchanam dalam pembuatan MYCIN (Wesley, 1984). *Certainty* *factor* merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. *Certainty* *factor* didefinisikan sebagai berikut (Giarattano dan Riley, 1994): [12]

**CF(H,E) = MB(H,E) – MD(H,E)**

Keterangan

|  |  |  |
| --- | --- | --- |
| CF(H,E) | : | *Certainty factor* dari hipotesis H yang dipengaruhi oleh gejala (evidence) E. Besarnya CF berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak. |
| MB(H,E) | : | Ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh gejala E. |
| MD(H,E) | : | Ukuran kenaikan ketidakpercayaan (*measure of increased* *disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E. |

* + - 1. **Bentuk *Certainty Factor***

1. CF Sequensial

Bentuk dasar rumus *certainty factor* sebuah aturan jika E maka H. [12]

Keterangan

|  |  |  |
| --- | --- | --- |
| CF(E,e) | : | *Certainty factor evidence* E yang dipengaruhi oleh *evidence e* |
| CF(H,E) | : | *Certainty factor* hipotesis dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika CF(E,e) = 1 |
| CF(H,e) | : | *Certainty factor* hipotesis yang dipengaruhi oleh *evidence* e |

Jika semua *evidence* pada *antecedent* diketahui dengan pasti, maka rumusnya ditunjmmukkan dibawah ini. [12]

CF Sequensial diperoleh dari hasil perhitungan CF paralel dari semua premis dalam satu aturan dengan CF aturan yang diberikan oleh pakar. Adapun rumus untuk melakukan perhitungan untuk CF Sequensial. [12]

Keterangan

CF(x,y) : CF Paralel

CF(x) : CF Sequensial dari semua premis

CF(y) : CF Pakar

1. CF Gabungan

CF Gabungan merupakan CF akhir dari sebuah calon konsklusi. CF ini dipengaruhi oleh CF Paralel dari aturan yang menghasilkan konsklusi tersebut. CF gabungan diperlukan jika suatu konsklusi diperoleh dari beberapa aturan sekaligus. CF akhir dari satu aturan dengan aturan yang lain digabungkan untuk mendapatkan nilai CF akhir bagi calon konsklusi tersebut. [12]

* + 1. **Analisis dan Perancangan Sistem** 
       1. **Konsep Dasar Analisis Sistem**

Analisis sistem didefinisikan sebagai bagaimana memahami dan menspesifikasi dengan detail apa yang harus dilakukan oleh sistem. Sementara sistem desain diartikan sebagai menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Dengan demikian, analisis dan desain sistem informasi (ANSI) bisa didefinisikan sebagai: Proses organisasional kompleks di mana sistem informasi berbasis komputer diimplementasikan.

Kesuksesan suatu sistem informasi tergantung pada analisis dan perancangan yang baik. Tahapan analisis akan menentukan masalah apa yang harus diselesaikan pada organisasi atau perusahaan. Kesalahan dalam tahap ini akan mengakibatkan masalah tetap ada walaupun sistem informasi telah diimplementasikan. Sementara tahapan desain akan sangat menentukan seperti apa sistem akan berfungsi. Walaupun pada tahapan analisis masalah utama sudah terpetakan dengan benar, kesalahan desain akan mengakibatkan kegagalan penyelesaian masalah oleh sistem komputer. Dengan demikian, dua langkah ini adalah langkah yang sangat menentukan untuk pengembangan sistem. [13]

* + - 1. **Metodelogi Pengembangan Sistem**

Beberapa ahli membagi proses-proses pengembangan sistem ke dalam sejumlah urutan yang berbeda-beda. Tetapi semuanya akan mengacu pada proses-proses standar berikut: analisis, desain, implementasi, dan pemeliharaan.

Pada perkembangannya, proses-proses standar tadi dituangkan dalam satu metode yang dikenal dengan nama Systems Development Life Cycle (SDLC) yang merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain. SDLC meliputi fase-fase sebagai berikut: identifikasi dan seleksi proyek, inisiasi dan perencanaan proyek, analisis, desain (logikal dan fisikal), implementasi, dan pemeliharaan. [13]

* + - 1. **Identifikasi dan Seleksi Sistem**

Merupakan langkah pertama dalam SDLC keseluruhan informasi yang dibutuhkan oleh sistem: identifikasi, analisis, prioritas, dan susun ulang.

Aktivitas yang biasa dilakukan pada tahap ini meliputi mewawancarai pakar, merangkum pengetahuan yang didapatkan, dan mengestimasi cakupan sistem dan mendokumentasikan hasilnya. Tahapan ini akan menghasilkan laporan kelayakan yang berisi definisi masalah dan rangkuman tujuan yang ingin dicapai dari proyek yang dipilih. [13]

* + - 1. **Tahapan Analisis**

Tahapan analisis adalah tahapan di mana sistem yang sedang berjalan dipelajari dan sistem pangganti diusulkan. Dalam tahapan ini dideskripsikan sistem yang sedang berjalan, masalah, dan kesempatan didefinisikan, dan rekomendasi umum untuk bagaimana memperbaiki, meningkatkan atau mengganti sistem yang sedang berjalan diusulkan. Tujuan utama dari fase analis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis (business need) dan persyaratan proses dari sistem baru. Ada beberapa aktivitas dalam fase ini: [13]

1. **Pengumpulan Informasi**

Langkah awal pada tahapan analisis adalah mengumpulkan informasi tentang bagaimana proses-proses bisnis yang ada pada sistem lama berjalan. Kemudian ditentukan pada titik-titik mana saja proses bisnis yang mengalami masalah yang bisa diselesaikan dengan sistem informasi. Kelemahan-kelamahan dari sistem lama diidentifikasi dan diperbaiki dengan sistem baru.

1. **Mendefinisikan Sistem Requirement**

Dari informasi kelemahan sistem yang didapat, analis sistem kemudian mendefinisikan apa saja sebenarnya yang dibutuhkan oleh sistem lama untuk mengantisipasi masalah. Inilah yang disebut sebagai system requirement (kebutuhan sistem). Seringkali kebutuhan ini akan mengubah total keseluruhan proses bisnis pada sistem lama, tetapi kadang-kadang hanya perubahan penambahan beberapa prosedur tertentu.

1. **Memprioritaskan Kebutuhan**

Dalam beberapa kasus, kebutuhan yang diperoleh sangat lengkap dan rumit. Ketersediaan waktu dan sumber daya lain untuk menyelesaikan keseluruhan requirement bisa saja tidak mencukupi. Pada kondisi seperti ini maka analis akan memprioritaskan kebutuhan-kebutuhan yang dianggap kritis untuk diprioritaskan.

1. **Menyusun dan Evaluasi Alternatif**

Satu hal yang tidak boleh dilupakan analis adalah rencana kedua. Setelah menyusun dan memprioritaskan kebutuhan, analis harus menyiapkan alternatif jika seandainya susunan kebutuhan nantinya akan ditolak oleh klien.

1. **Analisis SWOT**

SWOT adalah metode perencanaan strategis yang digunakan untuk mengevaluasi kekuatan (*strengths*), kelemahan (*weakness*), peluang (*opportunities*), dan ancaman (*threats*) dalam suatu proyek atau suatu spekulasi bisnis. Keempat faktor itulah yang membentuk akronim SWOT (*Strengths*, *Weakness*, *Opportunities*, dan *Threats*). Proses ini melibatkan penentuan tujuan yang spesifik dari spekulasi bisnis atau proyek dan mengidentifikasi faktor internal dan eksternal yang mendukung dan yang tidak dalam mencapai tujuan tersebut.

Analisis SWOT dapat diterapkan dengan cara menganalisis dan memilah berbagai hal yang mempengaruhi keempat faktornya, dimana aplikasinya adalah bagaimana kekuatan (*Stengths*) mampu mengambil keuntungan (*Opportunities*) yang ada, bagaimana cara mengatasi kelemahan (*Weakness*) yang ada, selanjutnya bagaimana kekuatan (*Strengths*) mampu menghadapi ancaman (*Threats*) yang ada, dan terakhir adalah bagaimana cara mengatasi kelemahan (*Weakness*) yang mampu membuat ancaman (*Threats*) menjadi nyata atau menciptakan sebuah ancaman baru. [14]

1. Kekuatan (*Strenghts*)

Kekuatan adalah sumber daya, keterampilan, atau keunggulan-keunggulan lain yang berhubungan dengan para pesaing perusahaan dan kebutuhan pasar yang dapat dilayani oleh perusahaan yang diharapkan dapat dilayani. Kekuatan adalah kompetisi khusus yang memberikan keunggulan kompetitif bagi perusahaan dipasar.

1. Kelemahan (*Weaknesses*)

Kelemahan adalah keterbatasan atau kekurangan dalam sumber daya, keterampilan, dan kapabilitas yang secara efektif menghambat kinerja perusahaan. Keterbatasan tersebut dapat berupa fasilitas, sumber daya keuangan, kemampuan menajemen dan keterampilan pemasaran dapat merupakan sumber dari kelemahan perusahaan.

1. Peluang (*Opportunities*)

Peluang adalah situasi penting yang menguntungkan dalam lingkungan perusahaan. Kecenderungan-kecenderungan penting merupakan salah satu sumber peluang.

1. Ancaman (*Threats*)

Ancaman adalah situasi penting yang tidak menguntungkan dalam lingkungan perusahaan. Ancaman merupakan pengganggu utama bagi posisi sekarang atau yang diinginkan perusahaan

1. **Analisis Kebutuhan**

Tujuan dari fase analisis adalah memahami dengan sebenar-benarnya kebutuhan dari sistem baru dan mengembangkan sebuah sistem yang mewadahi kebutuhan tersebut, atau memutuskan bahwa sebenarnya pengembangan sistem baru tidak dibutuhkan. Penentuan kebutuhan sistem merupakan langkah yang paling krusial dalam tahapan SDLC. Kebutuhan sistem bisa diartikan sebagai berikut: pernyataan tentang apa yang harus dikerjakan oleh sistem, dan pernyataan tentang karakteristik yang harus dimiliki sistem.

Untuk mempermudah analis sistem dalam menentukan keseluruhan kebutuhan secara lengkap, maka analis membagi kebutuhan sistem ke dalam dua jenis. Jenis pertama adalah kebutuhan fungsional (functional requirement). Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem. Jenis kedua adalah kebutuhan nonfungsional (nonfunctional requirements). Kebutuhan ini adalah tipe kebutuhan yang berisi properti perilaku yang dimiliki oleh sistem, meliputi: operasional, kinerja, keamanan, dan politik dan budaya. [13]

1. **Analisis Kelayakan**

Dokumen yang dihasilkan dari tahapan-tahapan sebelumnya dikumpulkan menjadi suatu proposal pendahuluan proyek. Untuk memastikan usulan tersebut bisa diteruskan menjadi proyek yang menguntungkan maka proyek harus dievaluasi kelayakannya dari berbagai segi kelayakan, di antaranya: kelayakan teknis, kelayakan operasional, kelayakan ekonomi, dan kelayakan hukum. [13]

* + - 1. **Tahapan Desain**

Tahapan desain adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang riil. Tahapan desain sistem dapat dibagi menjadi 2 tahap, yaitu desain logis (logical design) dan tahapan desain fisik (physical design). Adapun perbedaan dari keduanya dapat dijelaskan sebagai berikut: [13]

1. **Desain Logis**

Desain logis adalah bagian dari fase desain dalam SDLC di mana semua fitur-fitur fungsional dari sistem dipilih dari tahapan analisis dideskripsikan terpisah dari platform komputer yang nanti digunakan. Hasil tahapan ini adalah: deskripsi fungsional mengenai data dan proses yang ada dalam sistem baru, dan deskripsi yang detail dari spesifikasi sistem meliputi: input (data apa saja yang menjadi input), output (informasi apa saja yang menjadi output), dan process (prosedur apa saja yang harus dieksekusi untuk mengubah input menjadi output).

Tahapan desain logis biasanya menghasilkan beberapa dokumen, diantaranya dokumen model data, dokumen model proses, rancangan tabel, hierarki antar modul, sampai desain antar muka dari sistem yang akan dibuat. [13]

1. **Desain Fisik**

Pada bagian bagian ini, spesifikasi logis diubah ke dalam detail teknologi dimana pemrograman dan pengembangan sistem bisa diselesaikan. Pada tahapan inilah aktivitas coding dilakukan. Adapun output dari sistem ini adalah: deskripsi teknis mengenai pilihan teknologi perangkat lunak dan perangkat keras yang digunakan, dan deskripsi yang detail dari spesifikasi sistem meliputi: modul-modul program, file-file, sistem jaringan, dan sistem perangkat lunak.

Pada tahap desain, ada beberapa aktivitas utama yang dilakukan, yaitu: merancang dan mengintegrasikan jaringan, merancang arsitektur aplikasi, mendesain antar muka pengguna, mendesain sistem antar muka, mendesain dan mengintegrasikan database, membuat prototype untuk detail dari desain, mendesain dan mengintegrasikan kendali sistem. [13]

* + - 1. **Implementasi**

Pada tahapan implementasi ini terdapat beberapa hal yang perlu dilakukan, yaitu: [13]

1. *Testing*, yaitu menguji hasil kode program yang telah dihasilkan dari tahapan desain fisik. Tujuan pengujian ada dua. Dari sisi pengembangan sistem, harus dijamin kode program yang dibuat bebas dari kesalahan sintaks maupun logika. Dari sisi pengguna, program yang dihasilkan harus mampu menyelesaikan masalah yang ada pada klien dan sistem baru harus mudah dijalankan dan dipahami oleh pengguna akhir.
2. Instalasi. Setelah program lulus ujicoba, maka perangkat lunak dan perangkat keras akan diinstal pada organisasi atau perusahaan klien dan secara resmi mulai digunakan untuk menggantikan sistem lama.

Output dari tahapan ini adalah: source code yang error free, prosedur pelatihan, dan buku panduan. [13]

* + - 1. **Pemeliharaan**

Langkah terakhir dari SDLC di mana pada tahapan ini sistem secara sistematis diperbaiki dan ditingkatkan. Hasil dari tahapan ini adalah versi baru dari perangkat lunak yang telah dibuat. Perbaikan yang dilakukan tingkatannya bisa sangat variatif, mulai dari memperbaiki program yang crash hingga berfungsi kembali sampai pada penambahan modul-modul program yang baru sebagai jawaban atas perubahan kebutuhan pengguna. [13]

* + 1. **Konsep Dasar Android**
       1. **Pengertian Android**

Google menginginkan agar Android bersifat terbuka dan gratis oleh karena itu hampir setiap kode program Android diluncurkan berdasarkan lisensi *open* *source* Apache yang berarti bahwa semua orang yang ingin menggunakan Android dapat men-*download* penuh *source* *code* nya. Disamping itu produsen perangkat keras juga dapat menambahkan ekstensinya sendiri kedalam Android sesuai kebutuhan produk mereka. Model pengembangannya yang sederhana membuat Android menarik bagi vendor perangkat keras (contoh: Samsung). [15]

* + - 1. **Arsitektur Android**

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut:

**Gambar 2.1 Arsitektur Android**

1. **Android *Run Time***

*Layer* yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasu Linux, Davlik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. Di dalam Android Runtime dibagi menjadi dua bagian yaitu:

1. *Core* *Libraries*: Aplikasi Android dibangun dalam bahasa java, sementara Davlik sebagai virtual mesinnya bukan *Virtual* *Machine* *Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java yang ditangani oleh *Core* *Libraries*.
2. *Davlik* *Virtual* *Machine*: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel melakukan threading dan manajemen tingkat rendah.
   * + 1. **Versi Android**

Pada penghujung tahun diperkirakan hampir semua vendor selular di dunia menggunakan android sebagai operating system. Adapun versi-versi android yang pernah dirilis adalah sebagai berikut: [15]

1. **Android 4.0 (*Ice Cream Sandwich / ICS*)**

Tombol virtual di Bar Sistem membiarkan pengguna menavigasi langsung ke kembali, *home* dan *Apps* terbaru. *Multitasking* dibuat lebih visual pake android versi 4.0 ini, lalu *resizeable widget* juga sudah ada pada adroid versi 4.0 ini, dll. Yang membuat versi 4.0 ini nampak jauh berbeda dari pendahulunya.



**Gambar 2.2 Logo android 4.0 (*Ice Cream Sandwich / ICS*)**

1. **Android 4.1 / 4.2 / 4.3 (*Jelly Bean*)**

Android 4.1 *Jelly Bean* baru saja resmi diluncurkan, dan sudah mulai bergulir pada pertengahan Juli 2012. Mari kita lihat fitur yang di dukung oleh sistem operasi ini. [16]

1. *Google now*

Merupakan sebuah fitur baru pada sistem Android yang terbaik. Sistem ini dapat membantu anda menyimpulkan dan mengatur apa yang anda lakukan pada titik yang lebih cepat dan lebih efektif.

1. *Project Butter*

Dibalik namanya yang aneh, terdapat ide yang begitu brilian. Tujuan *Project Butter* ini adalah untuk membuat Android berjalan pada solid 60fps lengkap dengan *triple buffering* dan *force vsync.*

1. *Andoid Beam*

*Android Beam* adalah sebuah NFC file sharing interface yang berfungsi untuk berbagi file dengan ponsel yang mendukung NFC hanya dengan ketukan. Setelah terjadi ketukan, informasi akan langsung ditransfer ke perangkat penerima.



**Gambar 2.3 Logo android 4.1 / 4.2 / 4.3 (*Jelly Bean*)**

1. **Android 4.4 (*Kit Kat*)**

Seperti yang sudah dirumorkan sebelumnya, Google akhirnya menyatukan semua pesan keluar dan masuk pada Android 4.4 *Kit Kat* melalui aplikasi *Hangouts*. Tambahan lainnya pada pengelolaan pesan adalah fitur emoji yang kini hadir langsung di aplikasi *keyboard* bawaan Android.

Pada bagian produktivitas, Google menambahkan dukungan untuk memungkinkan pengguna mencetak dokumen dengan printer langsung dari ponselnya. Saat ini Google mendukung printer-printer yang bisa tersambung dengan *Google Cloud Print, HP ePrint*, serta printer-printer yang memiliki aplikasi khusus di *Google Play*. Selain urusan cetak mencetak, urusan mengolah dokumen juga didukung penuh dengan aplikasi *Quick Office* yang kini disematkan dalam paket sistem operasi Android 4.4 *Kit Kat*.



**Gambar 2.4 Logo android 4.4 (*Kit Kat*)**

1. **Android 5.0 (*Lollipop*)**

Lebih banyak waktu bermain, sedikit waktu pengisian Listrik untuk jangka panjang dengan fitur penghemat baterai yang membuat perangkat anda bertahan hingga 90 menit. Dan sekarang lebih mudah untuk mengatur penggunaan kekuatan *smartphone*, dengan melihat perkiraan waktu yang tersisa sebelum perlu mengisi ulang, dan ketika pengisian terjadi, tidak memerlukan banyak waktu sampai penuh dan siap untuk dibawa pergi.

Lebih banyak cara untuk mengamankan perangkat, dengan menjamin perangkat aman dengan modus tamu pengguna. Atau dengan membuat beberapa *account* pengguna untuk memungkinkan teman-teman bisa masuk pada perangkat *smartphone*. Dalam kedua kasus, tidak ada yang bisa mengakses barang-barang pribadi dari pemilik *smartphone*. Bisa juga menggunakan *Android Smart Lock* untuk mengamankan ponsel atau tablet dengan memasangkannya dengan perangkat terpercaya seperti saat anda menggunakan mobil.



**Gambar 2.5 Logo android 5.0 (*Lollipop*)**

* + 1. **Konsep Basis Data**

Menurut kusrini, basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan degnan nilai (angka, deretan karakter, atau simbol). [17]

Menurut Fatahansyah, basis data dapat didefinisikan dalam sejumlah sudut pandang seperti: [18]

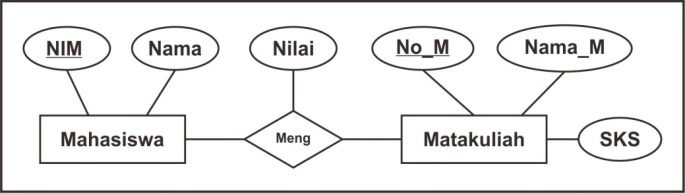
1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan.
   * 1. **ERD (*Entity Relationship Diagram*)**

Model data diagram hubungan entitas (*Entity Relationship Diagram*/ERD) dibuat berdasarkan anggapan bahwa dunia nyata terdiri atas koleksi objek-objek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu. Entitas adalah “sesuatu” atau “objek” pada dunia nyata yang dapat dibedakan satu dengan yang lainnya, yang bermanfaat bagi aplikasi yang sedang kita kembangkan.

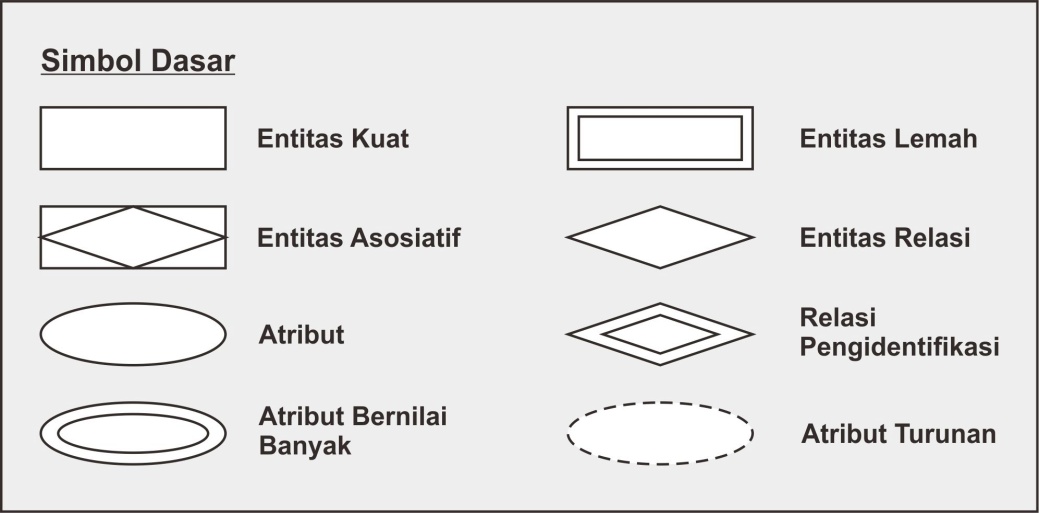
Entitasdalam basis data dideskripsikan berdasarkan atributnya. Sebagai contoh, nomor rekening membedakan suatu rekening seseorang yang menyimpan uangnya di suatu bank tertentu, dan nomor-nomor rekening tersebut merupakan atribut dari entitas rekening yang bersangkutan. Dalam hal ini, nomor rekening secara unik membedakan sebuah rekening dengan rekening yang lainnya.

Hubungan menjelaskan kaitan antara beberapa entitas. Sebagai contoh, mahasiswa memiliki orangtua, memiliki menjelaskan hubungan tertentu antara mahasiswa dengan orangtuanya. Dalam hal ini, himpunan semua entitas dengan tipe yang sama dan himpunan semua hubungan antar entitas dirujuk sebagai himpunan entitas (*entity set*), dan himpunan relasi (*relationship set*).

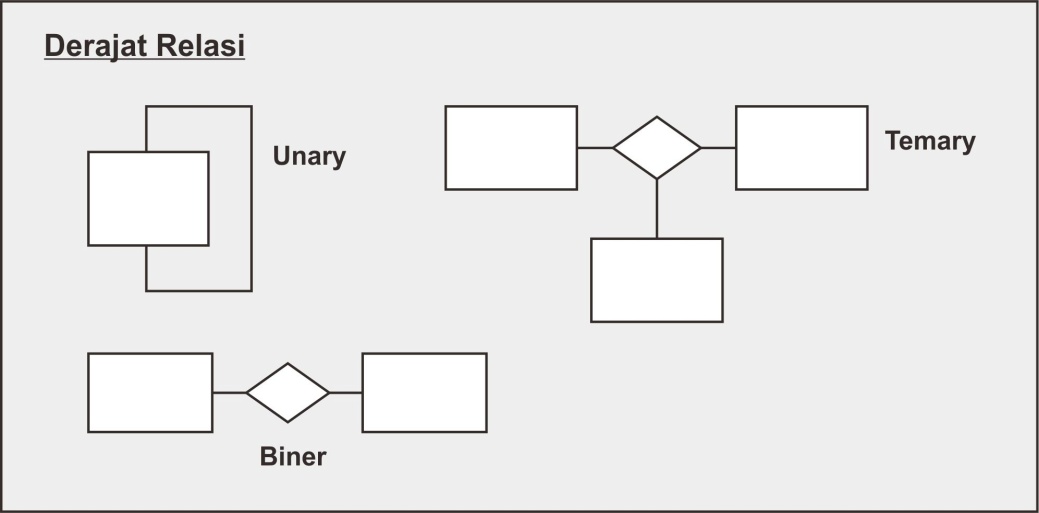
Secara skematik, basis data dapat dideskripsikan secara grafis dengan ERD yang memiliki komponen-komponen utama sebagai berikut, empat persegi panjang yang menggambarkan himpunan entitas, elips yang menggambarkan atribut, jajar genjang yang menggambarkan relasi/hubungan antar entitas, dan garis yang menyatukan atribut-atribut pada entitas tertentu serta menyatukan entitas-entitas dalam suatu relasi tertentu. [19]



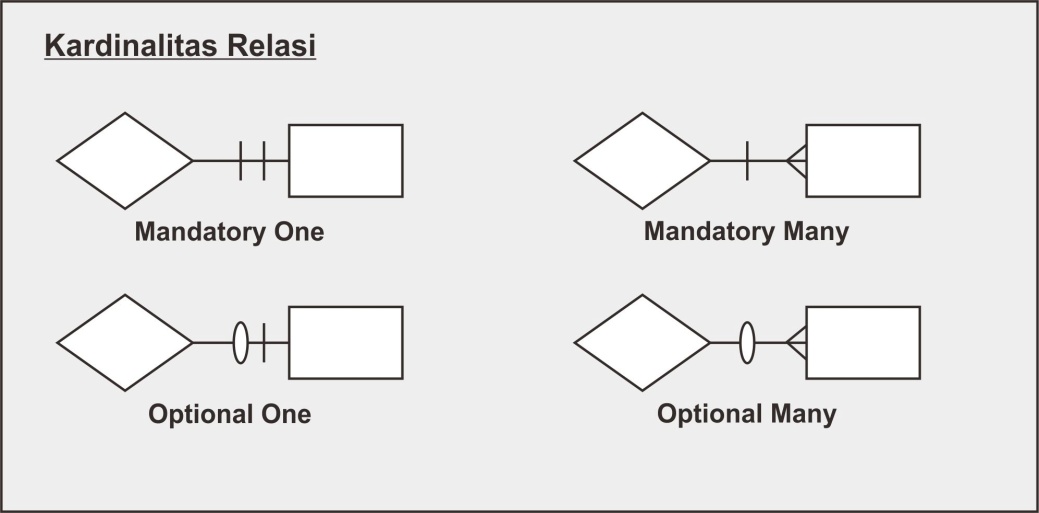
**Gambar 2.7 Contoh Penggambaran ERD**



**Gambar 2.8 Simbol Dasar Notasi Diagram E-R Dasar**



**Gambar 2.9 Derajat Relasi Notasi Diagram E-R Dasar**



**Gambar 2.10 Kardinalitas Relasi Notasi Diagram E-R Dasar**

* + 1. ***Unified Modeling Language* (UML)**

UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat anaisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [20]

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

*Use Case Diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *Use Case Diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *Use Case Diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem. [20]

**Tabel 2.2 Simbol *Use Case* Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

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

*Class diagram* adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class Diagram* menggambarkan struktur dan deskripsi *Classs, package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. [20]

**Tabel 2.3 Simbol *Class Diagram***

|  |  |  |
| --- | --- | --- |
| **Gambar** | **Nama** | **Keterangan** |
|  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
|  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari dua objek. |
|  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
|  | *Collaboration* | Deskripsi dari aturan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
|  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
|  | *Depedencey* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

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

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. [20]

**Tabel 2.4 Simbol *Sequence Diagram***

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

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

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendiskripsikan aktivitas yang dibentuk dalam suatu opersasi segingga dapat juga digunakan untuk aktifitas lainnya. Pembuatan *Activity Diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity Diagram*  juga digunakan untuk menggambarkan interaksi antara beberapa *use case*. [20]

**Tabel 2.5 Simbol *Activity Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Activity* | Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Activity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

* + 1. **Bahasa Pemrograman**

Bahasa pemrograman, atau sering diistilahkan juga dengan [bahasa komputer](https://id.wikipedia.org/wiki/Bahasa_komputer) atau [bahasa pemrograman komputer](https://id.wikipedia.org/w/index.php?title=Bahasa_pemrograman_komputer&action=edit&redlink=1), adalah instruksi standar untuk memerintah [komputer](https://id.wikipedia.org/wiki/Komputer). Bahasa pemrograman ini merupakan suatu himpunan dari aturan [sintaks](https://id.wikipedia.org/wiki/Sintaks) dan [semantik](https://id.wikipedia.org/wiki/Semantik) yang dipakai untuk mendefinisikan [program komputer](https://id.wikipedia.org/wiki/Program_komputer).

* + - 1. ***Java***

Sejarah *java* berawal pada tahun 1991 ketika perusahaan *Sun Microsystem* memulai *Green Project*, yakni proyek penelitian untuk *device intelligent customer electronic.* Bahasa tersebut haruslah bersifat *multiplatform,* tidak tergantung kepada *vendor* yang memanufactur chip tersebut.

Dalam penelitiannya, *Project Green* berhasil membuat *prototype* semacam PDA (*Personal Data Assistance*) yang dapat berkomunikasi antara satu dengan yang lain dan diberi nama *Star* *7*. Setelah berjalan beberapa lama, James Gosling, salah seorang anggota tim, merasa kurang puas dengan beberapa karakteristik dari kedua bahasa tersebut kemudian diberi nama Oak. Belakangan Oak beralih nama menjadi *Java*. [21]

* + - 1. **Konsep OOP (*Object Oriented Programming*)**

Pemrograman berorientasi objek berarti sebuah teknik pemrograman yang dalam proses pengembangannya menggunakan terminologi objek, dimana setiap objek memiliki atribut beserta dengan fungsi yang dapat saling berinteraksi satu dengan yang lain seperti halnya objek. Objek sendiri adalah bentuk nyata dari sebuah class dan class merupakan kumpulan dari atribut dan *method*/fungsi.

Menurut Sakur (2010) Pemrograman Berorientasi Objek memiliki tiga pilar utama, yaitu: [22]

1. *Encapsulation* (Pengkapsulan)

*Encapsulation* atau pengkapsulan merupakan sebuah teknik yang digunakan untuk melakukan penyembunyian informasi yang tidak diperlukan oleh *user.* Data yang disembunyikan berupa atribut dan fungsi yang saling berhubungan membentuk serangkaian struktur data.

1. *Inheritance* (Turunan)

*Inheritance* atau turunan merupakan salah satu pilar dari OOP yang merepresentasikan permodelan turunan dari dunia nyata, dimana setiap tipe data baru dapat merupakan turunan dari tipe data induknya atau *superclass* sehingga kelas turunan akan mewarisi seluruh sifat-sifat induknya.

1. *Polymorphism*

Jika ditinjau dari arti kata *poly,* berarti banyak dan *morphism* merupakan bentuk atau model (*form*)*.* Sehingga *polymorphism* diartikan sebagai model atau *form* yang dapat digunakan pada objek yang beragam. Objek-objek yang berbeda yang berasal dari induk yang sama dan dapat memiliki fungsi yang beragama.

* + 1. **Perangkat Lunak yang Digunakan**

Perangkat lunak adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud.

* + - 1. **Android SDK (*Software Development Kit*)**

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di *release* untuk Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk memulai pengembangan windows, maupun versi linux, arena SDK Android sifatnya gratis serta bebas di distribusikan. [21]

* + - 1. **Android Studio**

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated* *Development* *Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android. [22]

**BAB III**

**ANALISIS DAN PERANCANGAN SISTEM**

* 1. **Gambaran Umum Aplikasi**

Pada bagian ini akan membahas tentang analisis dan perancangan sistem pakar pendiagnosa penyakit paru-paru berbasis android. Agar menghasilkan sistem yang sesuai dengan kebutuhan maka harus dilakukan analisis dan perancangan yang baik dan benar. Analisis dan perancangan sistem ini dilakukan dengan metode UML (*Unified Modelling Language*) sebagai alat bantu pemodelan sistem.

Pada metode UML, *Use* *Case* digunakan untuk mendeskripsikan kebutuhan sistem dan bagaimana user mampu berinteraksi dengan sistem. Sedangkan *Activity diagram* menggambarkan rangkaian aliran dari aktivitas *user* (pengguna) dan sistem. *Class diagram* digunakan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Diagram* *sequence* digunakan untuk menggambarkan kelakuan dinamis sistem yang terjadi diantara objek atau entitas.

Aplikasi sistem pakar ini menggunakan metode *certainty factor*. Metode *certainty factor* digunakan untuk memberikan tahapan proses dengan memberikan alternatif gejala, maka akan diberikan hasil kemungkinan penyakit yang diderita.

* 1. **Analisis Masalah**

Penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

* + 1. **Analisis SWOT**

Dalam menganalisa system, penulis menggunakann analisis SWOT dikarenakan belum ada aplikasi lain sebagai pembanding. SWOT adalah perangkat umum yang didesain dan digunakan sebagai langkah awal dalam proses pengambilan keputusan dan sebagai langkah awal dalam proses terapan. SWOT adalah metode perencanaan strategis yang digunakan untuk mengevaluasi kekuatan (*strenght*), kelemahan (*weakness*), peluang (*opportunities*), dan ancaman (*threats*) dalam suatu proyek atau suatu spekulasi bisnis.

* + - 1. **Analisis Kekuatan (*Strenght*)**

1. Banyak masyarakat menggunakan smarphone berbasis android.
2. Tingkat kesibukan masyarakat yang tinggi dan mementingkan kepraktisan, sehingga aplikasi ini memudahkan user untuk mendiagnosa penyakit paru-paru.
3. Aplikasi ini dapat diakses dimana saja dan kapan saja.
4. Sistem menggunakan *certainty factor* sebagai dasar penyelesaian masalah yang akurat.
5. Proses aplikasi dapat digunakan secara *offline*.
   * + 1. **Analisis Kelemahan (*Weaknesses*)**
6. Tampilan aplikasi sistem pakar diagnosa penyakit paru-paru pada manusia masih sangat sederhana.
7. Program ini masih berbasis *offline*.
8. Data tentang penyakit paru-paru masih terbatas.
9. Aplikasi sistem pakar hanya dapat digunakan pada sistem operasi android
   * + 1. **Analisis Peluang (*Opportunitties*)**
10. Pasien lebih mudah dalam mendiagnosis penyakit paru-paru.
11. Biaya yang dikeluarkan lebih minim.
    * + 1. **Analisis Ancaman (*Threats*)**
12. Pengguna smartphone beralih ke sistem operasi lain.
13. Persaingan dengan aplikasi sistem pakar yang sejenis.

**Tabel 3.1 Tabel Strategi SWOT**

|  |  |  |
| --- | --- | --- |
| **Internal**  **Eksternal** | **Strength(S)** | **Weakness (W)** |
| **Opportunities (O)** | **Strategi S-O** | **Strategi W-O** |
| Dengan menggunakan Sistem Operasi Android, dan dapat digunakan secara *offline* maka dapat dimungkinkan untuk mendapatkan jumlah pengguna yang banyak, serta dapat digunakan dalam kondisi tanpa signal internet sekalipun. | Dengan semakin berkembangnya Sistem Operasi Android yang terus melakukan *update*, maka pengguna dan *developer* harus terus melakukan *update* terhadap aplikasi dan Sistem dalam *smartphone*. |
| **Threats (T)** | **Strategi S-T** | **Strategi W-T** |
| Dengan adanya kemungkinan bermunculan aplikasi sejenis, harus melakukan penambahan fitur atau melakukan pembaharuan terhadap sistem Android dan Web. | Meningkatkan kualitas aplikasi sehingga dapat bersaing dengan aplikasi lain yang sejenis. |

* 1. **Solusi yang Dapat diterapkan**

1. Memudahkan pengguna atau pasien dalam mengetahui penyakit paru-paru yang diderita.
2. Memberikan hasil diagnosa penyakit paru-paru secara cepat dan tepat.
3. Memberikan informasi tentang penyakit yang diderita.
4. Memberikan informasi tentang gejala yang diderita.
   1. **Solusi yang Dipilih**

Berdasarkan analisis masalah yang telah dibuatm dipilih solusi yang daoat menyelesaikan permasaahan tersebut, yaitu dengan menggunakan platfrom berbasis mobile untuk pembuatan sistem pakar. Platfrom ini dipilih karena penggunaannya dapat dilakukan dengan mudah, yaitu melalui *mobile* yang dapat sudah banyak penggunanya. Untuk mengatasi ketidakpastian data sistem ini dibangun dengan menggunakan algoritma atau metode *certaity* *factor* untuk mendapatkan solusi terbaik.

* 1. **Analisis Kebutuhan**

Analisis kebutuhan dari sistem dimaksud untuk mengetahui apa saja yang dibutuhkan oleh sistem yang akan digunakan dalam rangka mengganti atau memperbaiki kelemahan-kelemahan yang ditemukan dari sistem yang lama agar kinerja keseluruhan dari sistem menjadi lebih baik.

* + 1. **Analisis Kebutuhan Fungsional**

Kebutuhan fungsional merupakan fungsi-fungsi utama yang harus dapat dijalankan oleh sistem pakar. Adapun sistem pakar diagnosa penyakit paru-paru ini harus mampu memenuhi kebutuhan-kebutuhan seperti:

* + - 1. **Analisis Kebutuhan Fungsional Admin**

1. Sistem harus dapat melakukan pengelolaan data gejala.
2. Admin dapat melihat data gejala.
3. Admin dapat melakukan penambahan data gejala.
4. Admin dapat melakukan *update* data gejala.
5. Admin dapat menghapus data gejala.
6. Sistem harus dapat melakukan pengelolaan data penyakit.
7. Admin dapat melihat data penyakit.
8. Admin dapat melakukan penambahan data penyakit.
9. Admin dapat melakukan *update* data penyakit.
10. Admin dapat menghapus data penyakit.
11. Sistem harus dapat melakukan pengelolaan data *rule*.
12. Admin dapat melihat data rule.
13. Admin dapat melakukan penambahan data *rule*.
14. Admin dapat melakukan update *rule*.
15. Admin dapat menghapus data *rule*.
    * + 1. **Analisis Kebutuhan Fungsional Pengguna**
16. Sistem harus dapat menampilkan saran pencegahan penyakit.
17. Pengguna dapat melihat saran pencegahan penyakit pada paru-paru setelah hasil diagnosa penyakit atau pada menu list gejala.
18. Sistem harus dapat melakukan diagnosa penyakit.
19. Pengguna dapat mengakses menu diagnosa peyakit.
20. Penggna dapat memilih gejala yang dirasakan.
21. Pengguna dapat menghasilkan hasil diagnosa.
    * 1. **Kebutuhan Non-Fungsional**

Analisis kebutuhan non-fungsional merupakan fungsi-fungsi pendukung yang diperlukan agar sistem yang dibangun dapat beroperasi dengan baik. Adapun kebutuhan non-fungsional ini terdiri dari beberapa macam kebutuhan, seperti:

1. Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan untuk membangun sistem ini adalah sebagai berikut:

* + - * 1. Laptop dengan spesifikasi :

1. Processor : Intel *Processor* : Intel® CoreTM i5-4210U CPU@1.70GHz (4 CPUs), 2.4GHz
2. *Memory*: 8192MB RAM DDR3
3. *Hard Drive*: 1000 GB SATA
4. *Display*: 12.0 inch WXGA
5. *I/O Port*: 3x USB, VGA, LAN, MMC Card, HDMI, Audio
6. Sambungan kabel USB
7. *Handphone smartpohone* minimal OS android 4.0 (*Ice Cream Sandwich*)

Kebutuhan minimal *device* untuk menjalankan aplikasi ini adalah:

1. CPU 1 GHz
2. *Memory* 512 MB
3. *Memory internal* 1 GB
4. Resolusi Layar 768 x 1280 (Terbaik)
5. Android 4.0 (*Ice Cream Sandwich*)
6. Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan pernulis dalam pembuatan sistem ini adalah sebagai berikut:

1. Windows 10 Pro 64-bit sebagai Sitem Operasi.
2. Bahasa pemrograman Java
3. Android Studio
4. Android SDK Tools
5. SQLite Database

Kebutuhan minimal perangkat lunak dalam menjalanan aplikasi sistem pakar adalah sebagai berikut:

1. Semua jenis smartphone android dengan versi minimal OS yang digunakan 4.0 (*Ice Cream Sandwich*) keatas.
2. *Database* : SQLite
3. Kebutuhan Pengguna (*Brainware*)

Dalam proses perancangan suatu sistem maupun aplikasi tidak terlepas dari berbagai kebutuhan, baik itu kebutuhan hardware, software maupun brainware. Sumber daya manusia adalah orang yang terlibat dalam proses perancangan, pembuatan dan implementasi sistem. Adapun sumber daya manusia yang terlibat dalam pembuatan sistem pengecekan sistem pakar ini adalah sebagai berikut:

1. Analisis : orang yang bertanggungjawab atas penelitian, perencanaan, pengkoordinasian, dan merekomendasikan pemilihan perangkat lunak dan sistem yang paling sesuai kebutuhan.
2. Programmer : orang yang bertanggung jawab mengimplementasikan perancangan dari analisis dalam bentuk program dan aplikasi secara keseluruhan.
3. Pakar : orang yang bertanggungjawab atas validasi data penyakit.
   1. **Analisis Kelayakan Sistem**

Analisis kelayakan sistem merupakan proses yang menganalisa permasalahan yang telah ditentukan sesuai dengan tujuan akhir yang akan dicapai. Tujuan dari analisis kelayakan sistem adalah menentukan apakah sistem yang diusulkan layak atau tidak untuk diterapkan di objek penelitian, dalam kasus ini objek yang digunakan adalah paru-paru pada manusia.

* + 1. **Analisis Kelayakan Teknologi**

Kelayakan teknologi, tersedianya kebutuhan hardware dan software untuk sistem yang akan dibangun dengan harga yang akan terjangkau dan mudah didapat. Dalam pembuatan aplikasi sistem pakar ini, digunakan perangkat lunak yang sangat mudah didapatkan. Didukung dengan sistem operasi android yang bersifat *open* *source*, maka secara teknik aplikasi sistem pakar ini dapat dikatakan layak.

* + 1. **Analisis Kelayakan Hukum**

Pada sistem yang akan dibangun menggunakan software yang legal dan tidak melanggar hukum. Software yang dipakai merupakan software *open* *source* yaitu software yang dapat digunakan dan untuk pengembangan aplikasi juga gratis. Oleh karena itu sudah bisa dikatakan layak secara hukum.

* + 1. **Analisis Kelayakan Operasional**

Sistem yang dirancang didukung dengan adanya kelayakan operasional sebagai berikut:

1. Banyaknya pengguna handphone yang mampu mengoperasikan handphone smartphone android.
2. Pengguna dapat mengoperasikan aplikasi dengan tampilan yang *user* *friendly*.

Pengguna bisa melakukan pengobatan tanpa harus menggunakan pakar yang ahli dalam bidangnya.

* + 1. **Analisis Kelayakan Ekonomi**

Aspek yang dipertimbangkan dalam kelayakan ekonomi adalah besarnya dana yang diperlukan untuk mengembangkan sistem, serta manfaat yang akan diperoleh oleh sistem. Suatu sistem dikatakan ekonomis jika manfaat yang didapatkan lebih besar daripada biaya yang harus dikeluarkan.

1. Biaya
2. Biaya pengadaan, yaitu biaya pembelian perangkat keras.
3. Biaya persiapan operasional, yaitu biaya untuk pembelian perangkat lunak.
4. Biaya proyek, meliputi biaya pengembangan sistem, implementasi sistem.
5. Manfaat
   1. Manfaat berwujud, berupa penghematan dan peningkatan efisiensi waktu dan sumber daya, untuk mendapatkan hasil yang maksimal.
   2. Manfaat tak berwujud adalah manfaat yang tidak dapat diukur dengan satuan nilai. Penilaian dilakukan dengan taksiran yaitu peningkatan pelayanan terhadap kepuasan pengguna.
   3. **Analisis Pengetahuan**

Keberhasilan suatu sistem pakar terletak pada pengetahuan dan bagaimana mengolah pengetahuan tersebut agar dapat ditarik suatu kesimpulan. Pengetahuan yang diperoleh dari hasil wawancara dan analisa lewat buku dikonversi kedalam sebuah tabel penyakit dan gejala guna mempermudah proses pencarian solusi. Tabel jenis penyakit dan gejala ini digunakan sebagai pola pencocokan informasi yang dimasukan oleh pengguna (*user*) dan basis pengetahuan.

* + 1. **Daftar Penyakit**

Daftar jenis-jenis penyakit paru-paru dapat dilihat pada tabel 3.2.

**Tabel 3.2 Jenis-Jenis Penyakit**

|  |  |
| --- | --- |
| **Kode** | **Nama Penyakit** |
| **P001** | Asma |
| **P002** | Bronkiolitis |
| **P003** | TBC |

* + 1. **Daftar Gejala**

Daftar gejala pada penyakit paru-paru dapat dilihat pada tabel 3.3.

**Tabel 3.3 Gejala-Gejala Penyakit**

|  |  |
| --- | --- |
| **Kode** | **Gejala** |
| **G001** | Dada sesak |
| **G002** | Kesulitan bernapas |
| **G003** | Demam |
| **G004** | Nyeri dada |
| **G005** | Mengi |
| **G006** | Batuk ringan |
| **G007** | Kelelahan |
| **G008** | Gangguan kesadaran |
| **G009** | Terdapat sedikit dahak |
| **G010** | Keringat pada malam hari |
| **G011** | Retraksi dada |
| **G012** | Sesak napas |
| **G013** | Kuku kebiruan |
| **G014** | Otot leher menegang |
| **G015** | Batuk pada malam hari |
| **G016** | Sulit berbicara |
| **G017** | Napas yang cepat |
| **G018** | Penurunan berat badan |
| **G019** | Napas pendek yang lebih nyaman jika duduk |
| **G020** | Kulit kebiruan |
| **G021** | Batuk dengan dahak yang bercampur dengan darah |
| **G022** | Demam suhu rendah (sumer-sumer) |
| **G023** | Napas berbunyi |

Gejala merupakan basis pengetahuan untuk membuat suatu kesimpulan yang menjadi *goal*. Pada tabel 3.4 menjelaskan keterhubungan/relasi kode gejala dan kode penyakit.

**Tabel 3.4 Relasi Kode Gejala dan Kode Penyakit**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode Gejala** | **Kode Penyakit** | | |
| **P001** | **P002** | **P003** |
| **G001** | **\*** |  |  |
| **G002** | **\*** | **\*** |  |
| **G003** |  | **\*** |  |
| **G004** |  |  | **\*** |
| **G005** | **\*** | **\*** |  |
| **G006** |  | **\*** |  |
| **G007** | **\*** |  |  |
| **G008** | **\*** |  |  |
| **G009** | **\*** |  |  |
| **G010** |  |  | **\*** |
| **G011** |  | **\*** |  |
| **G012** |  |  | **\*** |
| **G013** |  | **\*** |  |
| **G014** | **\*** |  |  |
| **G015** | **\*** |  |  |
| **G016** | **\*** |  |  |
| **G017** |  | **\*** |  |
| **G018** |  |  | **\*** |
| **G019** | **\*** |  |  |
| **G020** | **\*** |  |  |
| **G021** |  |  | **\*** |
| **G022** |  |  | **\*** |
| **G023** | **\*** |  |  |

Keterangan:

G001, G002, ... , G023 : Kode Gejala

P001, P002, P003 : Kode Penyakit

\* : Relasi kode gejala dan kode penyakit

* + 1. **Nilai CF Rule untuk Gejala dan Penyakit / CF(H,E)**

Nilai CF Rule untuk gejala dan penyakit adalah nilai hipotesis dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika CF(E,e). Adapun nilai CF Rule tersebut di dapat dari pakar yang terkait dengan sistem pakar yang dibuat yaitu para dokter.

**Tabel 3.5 Nilai CF Rule**

|  |  |  |
| --- | --- | --- |
| **Rule** | **Kaidah Produksi** | **Nilai CF Rule** |
| R1 | IF G001 THEN P001 | CFR1 = 0,2 |
| R2 | IF G002 THEN P001 | CFR2 = 0,2 |
| R3 | IF G002 THEN P002 | CFR3 = 0,09 |
| R4 | IF G003 THEN P002 | CFR4 = 0,45 |
| R5 | IF G004 THEN P003 | CFR5 = 0,09 |
| R6 | IF G005 THEN P001 | CFR6 = 0,3 |
| R7 | IF G005 THEN P002 | CFR7 = 0,09 |
| R8 | IF G006 THEN P002 | CFR8 = 0,36 |
| R9 | IF G007 THEN P001 | CFR9 = 0,09 |
| R10 | IF G008 THEN P001 | CFR10 = 0,18 |
| R11 | IF G009 THEN P001 | CFR11 = 0,2 |
| R12 | IF G010 THEN P003 | CFR12 = 0,6 |
| R13 | IF G011 THEN P002 | CFR13 = 0,2 |
| R14 | IF G012 THEN P003 | CFR14 = 0,2 |
| R15 | IF G013 THEN P002 | CFR15 = 0,18 |
| R16 | IF G014 THEN P001 | CFR16 = 0,38 |
| R17 | IF G015 THEN P001 | CFR17 = 0,2 |
| R18 | IF G016 THEN P001 | CFR18 = 0,1 |
| R19 | IF G017 THEN P002 | CFR19 = 0,36 |
| R20 | IF G018 THEN P003 | CFR20 = 0,2 |
| R21 | IF G019 THEN P001 | CFR21 = 0,36 |
| R22 | IF G020 THEN P001 | CFR22 = 0,18 |
| R23 | IF G021 THEN P003 | CFR23 = 0,3 |
| R24 | IF G022 THEN P003 | CFR24 = 0,1 |
| R25 | IF G023 THEN P001 | CFR25 = 0,3 |

* + 1. **Nilai CF User untuk Gejala dan Penyakit / CF(E,e)**

Nilai CF User untuk gejala dan penyakit adalah nilai *evidence* E yang dipengaruhi *evidence* e. Adapun nilai CF User tersebut di dapat dari pakar yang terkait dengan sistem pakar yang dibuat yaitu para dokter.

**Tabel 3.6 Nilai CF User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kode CFUSER** | **Kode Gejala** | **Kode Penyakit** | **Keterangan** | **Nilai CFUSER** |
| U001 | G001 | P001 | Kemungkinan | CFUSER = 0,4 |
| U002 | G002 | P001 | Tidak Tahu | CFUSER = 0,2 |
| U003 | G002 | P002 | Kemungkinan Tidak | CFUSER = - 0,4 |
| U004 | G003 | P002 | Kemungkinan | CFUSER = 0,4 |
| U005 | G004 | P003 | Kemungkinan Besar | CFUSER = 0,6 |
| U006 | G005 | P001 | Kemungkinan | CFUSER = 0,4 |
| U007 | G005 | P002 | Hampir Pasti Tidak | CFUSER = - 0,8 |
| U008 | G006 | P002 | Kemungkinan Besar | CFUSER = 0,6 |
| U009 | G007 | P001 | Tidak Tahu | CFUSER = 0,2 |
| U010 | G008 | P001 | Kemungkinan | CFUSER = 0,4 |
| U011 | G009 | P001 | Tidak Tahu | CFUSER = 0,2 |
| U012 | G010 | P003 | Hampir Pasti | CFUSER = 0,6 |
| U013 | G011 | P002 | Kemungkinan | CFUSER = 0,4 |
| U014 | G012 | P003 | Pasti | CFUSER = 1,0 |
| U015 | G013 | P002 | Tidak Tahu | CFUSER = 0,2 |
| U016 | G014 | P001 | Kemungkinan | CFUSER = 0,4 |
| U017 | G015 | P001 | Kemungkinan | CFUSER = 0,4 |
| U018 | G016 | P001 | Kemungkinan Tidak | CFUSER = - 0,4 |
| U019 | G017 | P002 | Kemungkinan | CFUSER = 0,4 |
| U020 | G018 | P003 | Kemungkinan Tidak | CFUSER = - 0,4 |
| U021 | G019 | P001 | Kemungkinan Besar | CFUSER = 0,6 |
| U022 | G020 | P001 | Kemungkinan | CFUSER = 0,4 |
| U023 | G021 | P003 | Kemungkinan | CFUSER = 0,4 |
| U024 | G022 | P003 | Hampir Pasti | CFUSER = 0,8 |
| U025 | G023 | P001 | Kemungkinan Besar | CFUSER = 0,6 |

* + 1. **Rekomendasi Pakar**

Rekomendasi merupakan solusi atau saran yang diberikan kepada sistem atas hasil yang telah dilakukan.

**Tabel 3.7 Rekomendasi Pakar**

|  |  |
| --- | --- |
| 1. **ASTHMA** | |
| Pencegahan | 1. Hindari allergen dan polutan udara yang sudah diketahui. 2. Gunakan obat untuk pencegahan secara teratur, jangan berhenti walaupun sudah sembuh. |
| Tindakan | 1. Hindari allergen dan iritan di rumah maupun di kantor jika mungkin. 2. Bawalah selalu obat bersama anda. Mintalah pada dokter obat untuk keadaan gawat. 3. Duduk tegak disaat serangan datang. 4. Berlatihlah menarik napas panjang setiap pagi untuk mengurangi penimbunan secret paru. |
| Pengobatan | Dokter akan memberikan :   1. Ekspektoran untuk mencairkan dahak 2. Bronkodilator untuk melebarkan saluran pernapasan. 3. Kortison intravena (hanya pada keadaan gawat darurat) untuk mengurangi raksi alergi tubuh. 4. Kortison dengan cara disemprot ke hidung, yang ternyata memiliki efek samping yang lebih kecil jika dibandingkan dengan yang dimakan. 5. Sodium kromoglikat juga disemprot. Ini merupakan obat pencegahan. |
| 1. **TUBERKULOSIS (TB PARU)** | |
| Pencegahan | Vaksinasi dengan BCG, suatu strain bakteri tuberculosis ini dapat mencegah infeksi. |
| Tindakan | 1. Membawa penderita teberkulosis ke rumah sakit yang terdapat pelayanan DOTS. 2. Pengobatan “Obat Anti Tuberculosis” (OAT). Kuman akan meluas setelah pengobatan OAT dilakukan. 3. Kadangkala anda perlu mengecek dahak pada pagi hari selama 3 hari, untuk melihat apakah TBC masih aktif. |
| Pengobatan | Dokter akan memberikan obat anti-TBC, termasuk : INH (Isotonic acid hydrizide); ethambutol; paraaminosalicylic acid; atau rifampin |
| 1. **BRONKIOLITIS** | |
| Pencegahan | 1. Pasang pengatur kelembapan di ruang tidur anak. Pasang alat ini setiap malam selama dan sesudah infeksi pernapasan yang mengenai anak yang rentan terhadap bronkiolitis. 2. Amati dan hindari setiap aktivitas yang kelihatannya dapat memicu serangan pada anak, seperti bermain di udara malam yang dingin. 3. Kurangi kontak antara anak dengan orang lain, khususnya anak lain untuk menghindari tertular demam. |
| Tindakan | Pasang pengatur kelembapan pada ruang anak setinggi mungkin, khususnya pengatur kelembapan dingin. Bila anda tidak mempunyai pengatur suhu, isi bak mandi dengan air panas atau dingin dan tutup pintu serta jendela untuk mendapatkan ruang dengan kelembapan tinggi. Biarkan anak di dalam ruang tersebut selama 20 menit beberapa kali sehari, khusunya sebelum tidur. Bila anak terbangun di malam hari akibat mengi atau sesak napas, ulangi proses tersebut. |
| Pengobatan | Dokter akan memberikan antibiotic untuk melawan infeksi bakteri. |

* 1. **Perancangan Sistem**

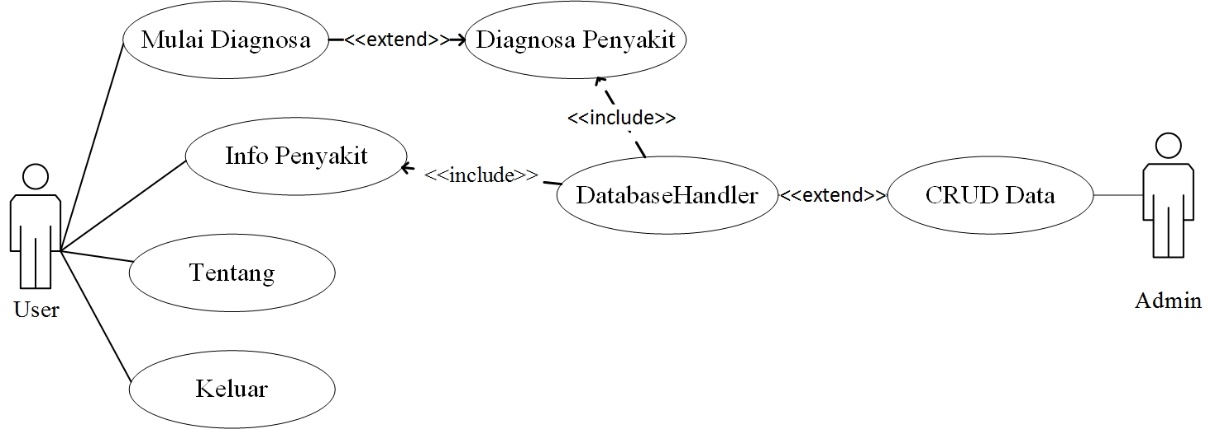
Perancangan sistem secara umum dilakukan dengan maksud untuk memberikan gambaran umum tentang sistem yang baru atau sistem yang sedang dirancang. Rancangan ini mengidentifikasi kompunen-komponen sistem informasi yang dirancang secara rinci.

* + 1. **Perancangan UML**

UML (*Unified Modeling Language*) adalah sebuah bahasa yang menggunakan grafik atau gambar untuk memisualisasi, menspesifikasikan, membangun, dan mendokumentasikan dari sebuah sistem pengembangan *software* berbasis *object-oriented*.

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

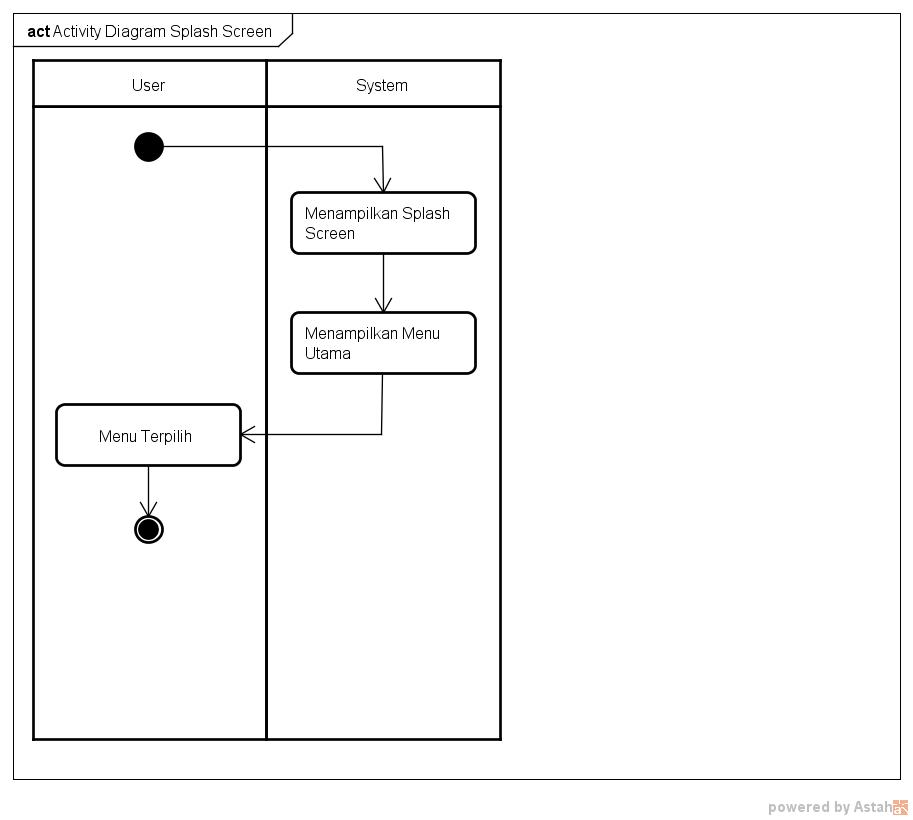
Berikut ini adalah bentuk dari *use* *case* *diagram* yang digambarkan dalam perancangan sistem. Disini akan dijelaskan siapa dan apa saja yang dilakukan ketika aplikasi jalan.



**Gambar 3.1 *Use* *Case* *Diagram* Aplikasi Android**

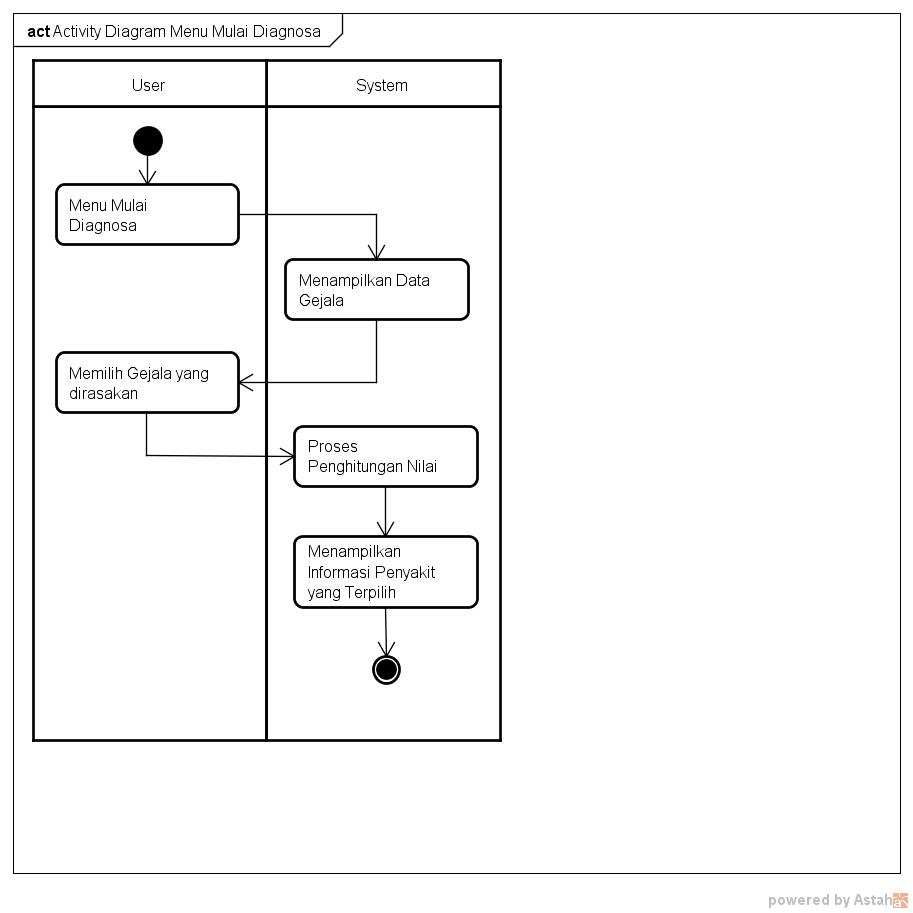
* + - 1. ***Activity Diagram***

Berikut *activity* *diagram* pada aplikasi sistem pakar ini:

1. *Activity* *Diagram* aplikasi Paru-Paruku (menampilkan menu *splash* *screen*)

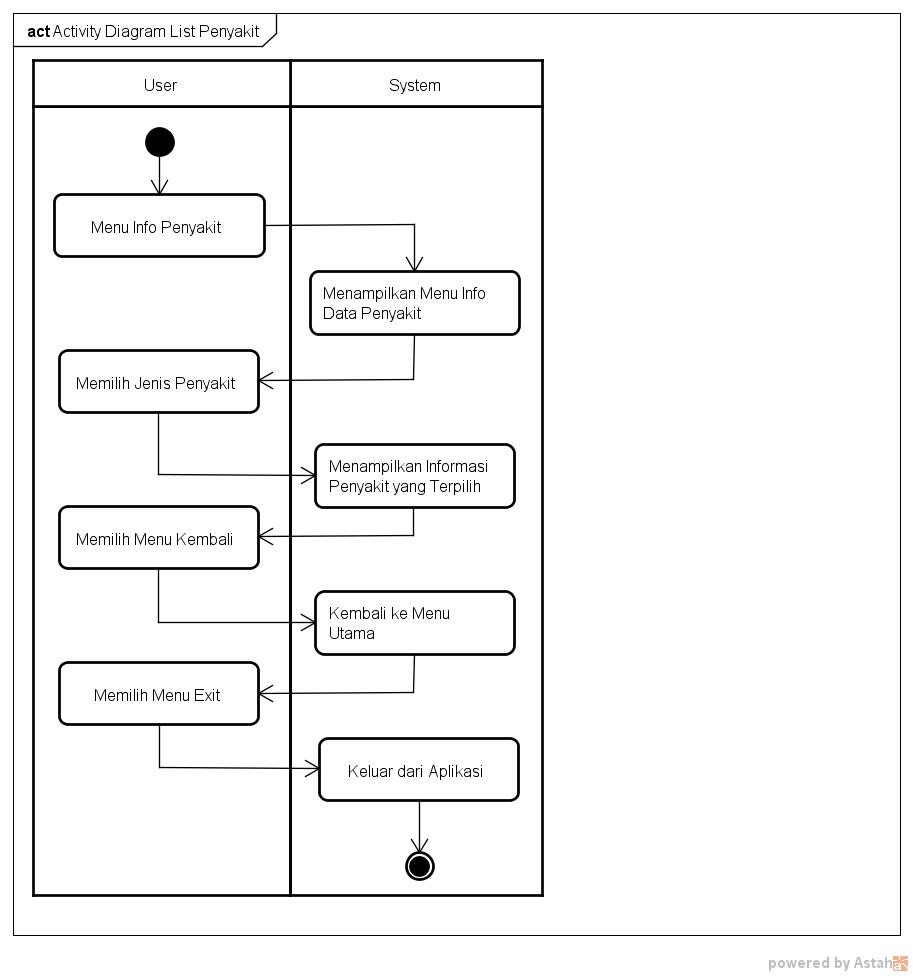
**Gambar 3.2 *Activity* *Diagram* menu *splash screen***

1. *Activity* *Diagram* menu Mulai Diagnosa



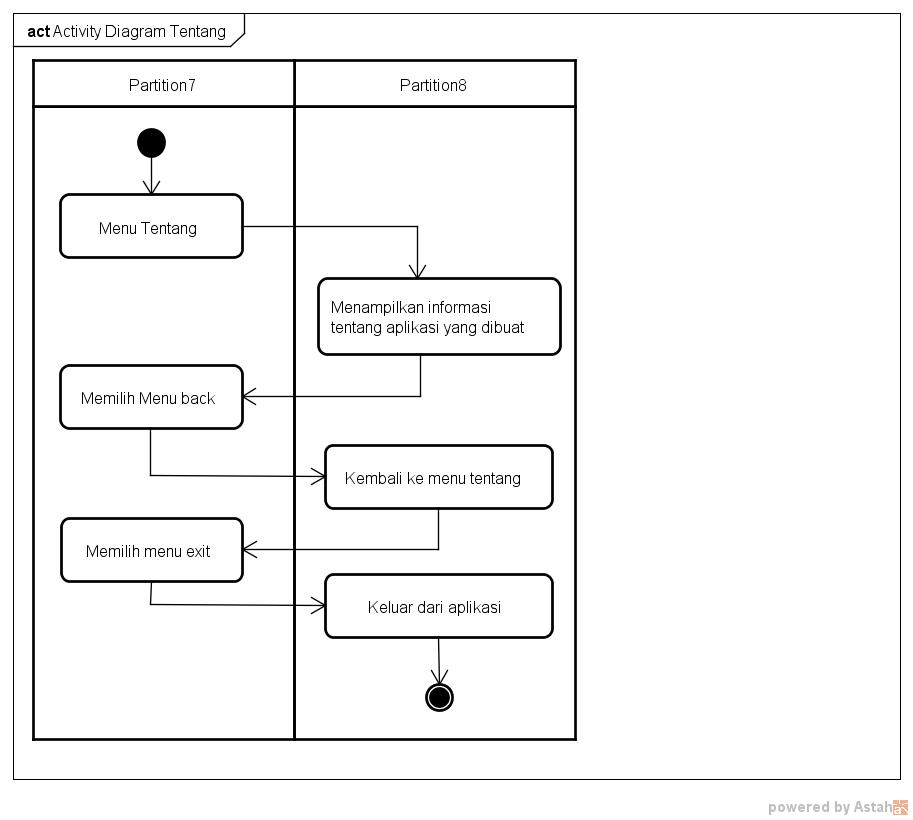
**Gambar 3.3 *Activity* *Diagram* menu mulai diagnosa**

1. *Activity* *Diagram* menu Info Penyakit

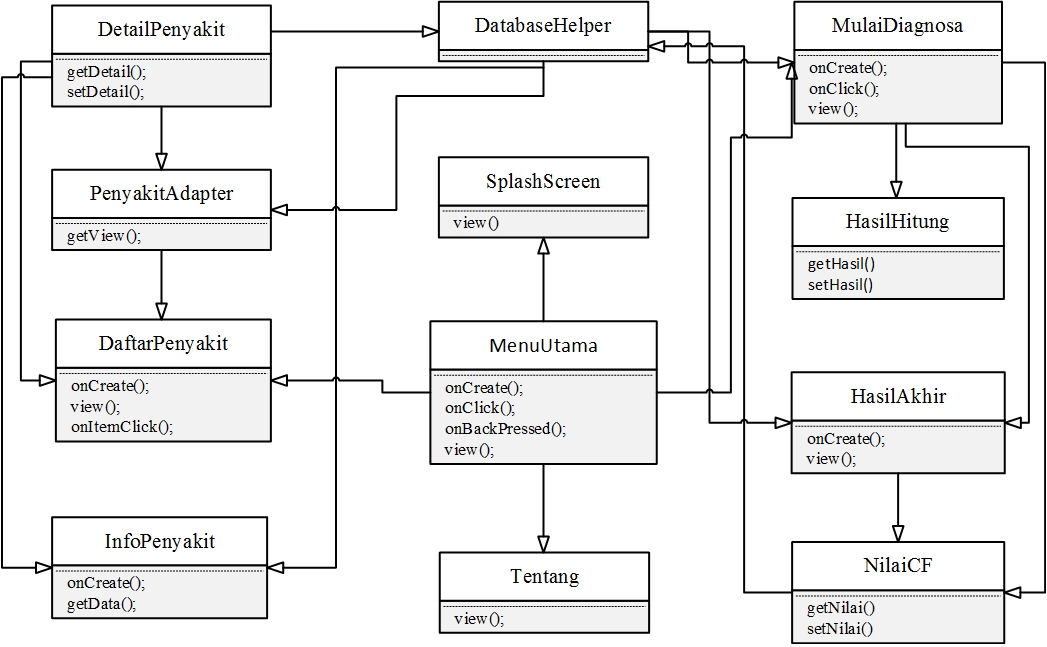


**Gambar 3.4 *Activity* *Diagram* menu info penyakit**

1. *Activity* *Diagram* menu Tentang



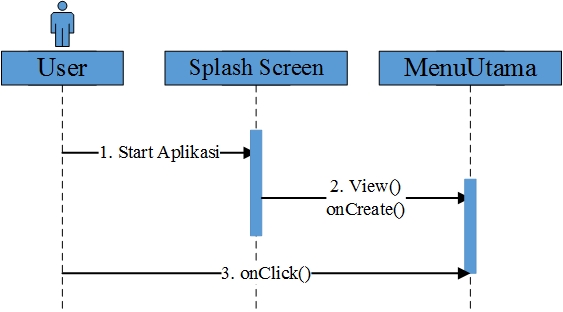
**Gambar 3.5 *Activity* *Diagram* menu tentang**

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

**Gambar 3.6 *Class Diagram***

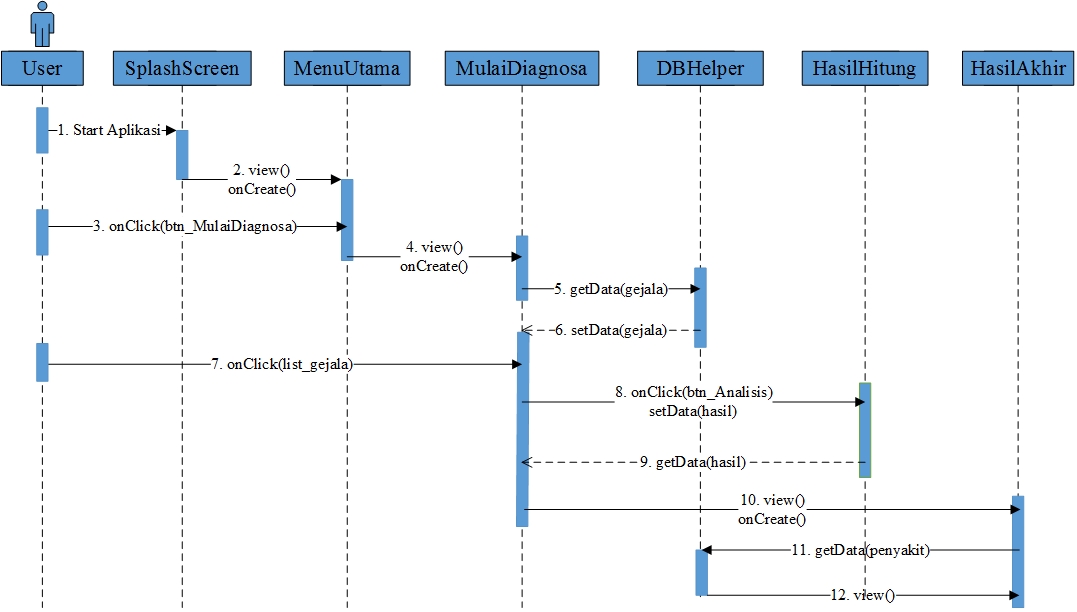
* + - 1. ***Sequence* *Diagram***

Berikut *sequence* *diagram* pada aplikasi sistem pakar ini:

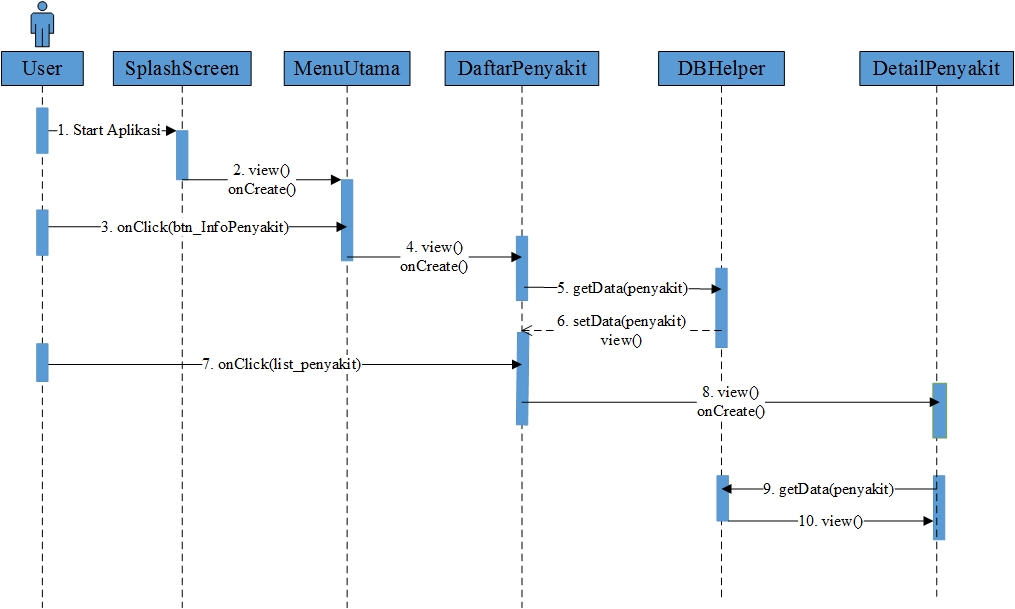
1. *****Sequence* *Diagram* menu splash screen

**Gambar 3.7 *Sequence Diagram* menu *splash screen***

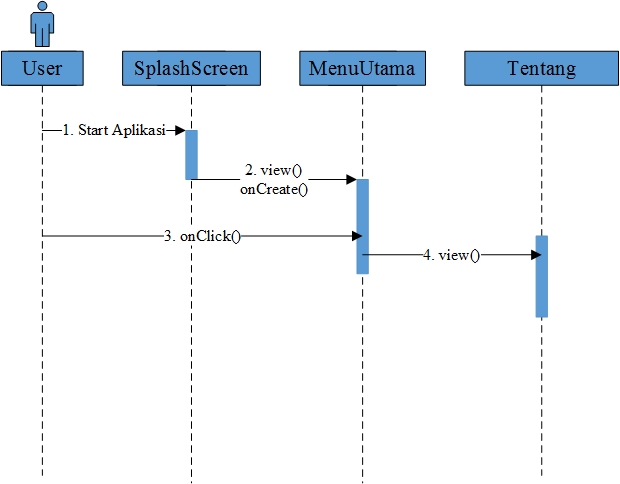
1. *Sequence* *Diagram* menu mulai diagnosa



**Gambar 3.8 *Sequence Diagram* menu mulai diagnosa**

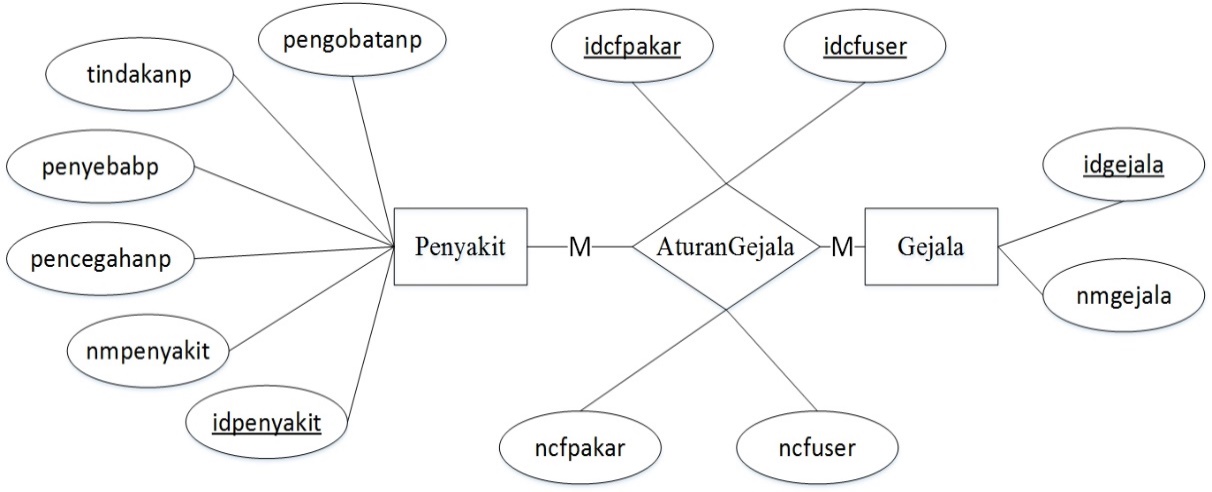
1. *****Sequence* *Diagram* menu info penyakit

**Gambar 3.9 *Sequence Diagram* menu info penyakit**

1. *Sequence* *Diagram* menu tentang

**Gambar 3.10 *Sequence Diagram* menu tentang**

* + 1. **Perancangan Basis Data Aplikasi Mobile**
       1. **Diagram ERD**

****

**Gambar 3.11 Rancangan Relasi Antar Tabel Aplikasi *Mobile***

* + - 1. **Rancangan Struktur Tabel**

Adapun perancangan struktur tabel aplikasi mobile, akan dibagi kedalam 4 tabel diantaranya, gejala, penyakit, cfpakar, cfuser, akan dibahas secara detail pada lampiran 3.

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

Perancangan interface atau antar muka program berperan untuk menghubungkan antara pengguna dengan aplikasi. Peranan interface sangat penting karena dengan interface yang baik akan membuat pengguna program menjadi lebih mudah dan enak di gunakan serta tidak membingungkan. Berikut adalah interface yang ada pada aplikasi Aplikasi Sistem Pakar Untuk Diagnosa Penyakit Paru-Paru.

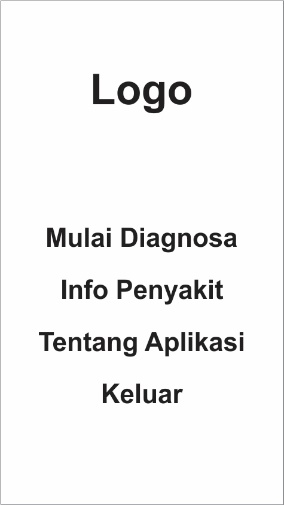
* + - 1. **Halaman *Splash* *Screen***

Tampilan ini adalah tampilan pembuka atau splash screen, tampilan ini akan muncul ketika aplikasi pertama kali dibuka dan setelah splash screen selesai akan langsung menuju tampilan awal.



**Gambar 3.12 Tampilan *Splash* *Screen***

* + - 1. **Halaman Menu Utama**

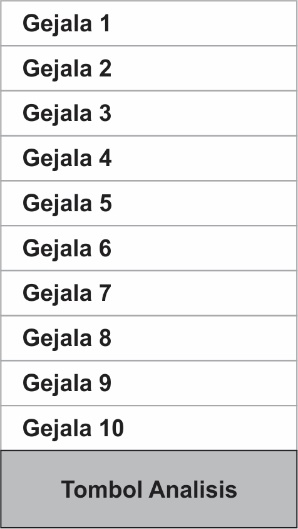
Tampilan ini adalah tampilan awal dari aplikasi setelah menu *splash screen* selesai berjalan. Pada halaman ini akan menampilkan isi dari menu-menu yang telah disediakan oleh sistem yang dibuat.



**Gambar 3.13 Tampilan Menu Utama**

* + - 1. **Halaman Mulai Diagnosa**

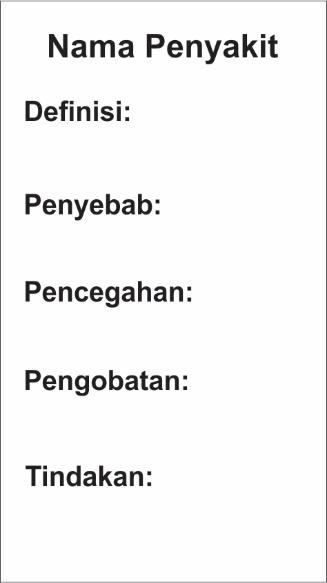
Halaman Mulai Diagnosa merupakan tempat yang dimana *user* bisa memilih gejala penyakit yang dialami oleh *user*. Pada halaman ini terdapat tombol analisa yang digunakan untuk memulai analisa



**Gambar 3.14 Tampilan Mulai Diagnosa**

* + - 1. **Halaman Hasil Diagnosa**

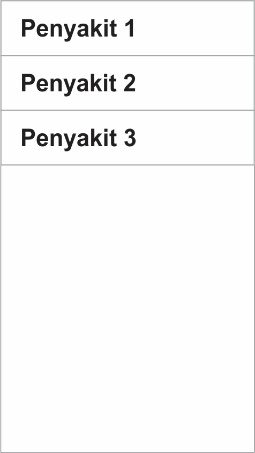
Halaman hasil diagnosa merupakan tempat yang berisi definisi penjelasan tentang kemungkinan penyakit yang dialami *user*, dimana *user* bisa mengetahui penyakit yang diderita.



**Gambar 3.15 Tampilan Hasil Diagnosa**

* + - 1. **Halaman Info Penyakit**

Halaman ini menyediakan beberapa jenis penyakit paru-paru. Pada halaman ini akan mempermudah pengguna dalam menggunakannya dan terdapat tombol menu yang digunakan untuk kembali ke halaman utama.



**Gambar 3.16 Tampilan Menu Info Penyakit**

* + - 1. **Halaman Info Detail Penyakit**

Halaman ini menyediakan informasi penyakit paru-paru secara detail. Pada halaman ini menampilkan definisi, penyebab, pencegahan, tindakan, dan pengobatan.



**Gambar 3.17 Tampilan Menu Info Detail Penyakit**

* + - 1. **Halaman Tentang**

Halaman ini merupakan tampilan yang berisi tentang versi aplikasi yang dibuat. Pada halaman ini juga tersedia tombol menu untuk menuju ke halaman utama.

**Gambar 3.18 Tampilan Menu Tentang**

**BAB IV**

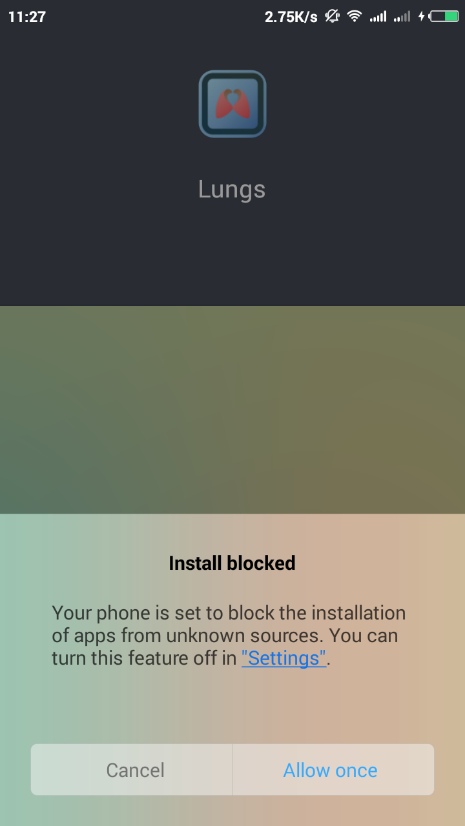
**IMPLEMENTASI DAN PEMBAHASAN**

* 1. **Implementasi**

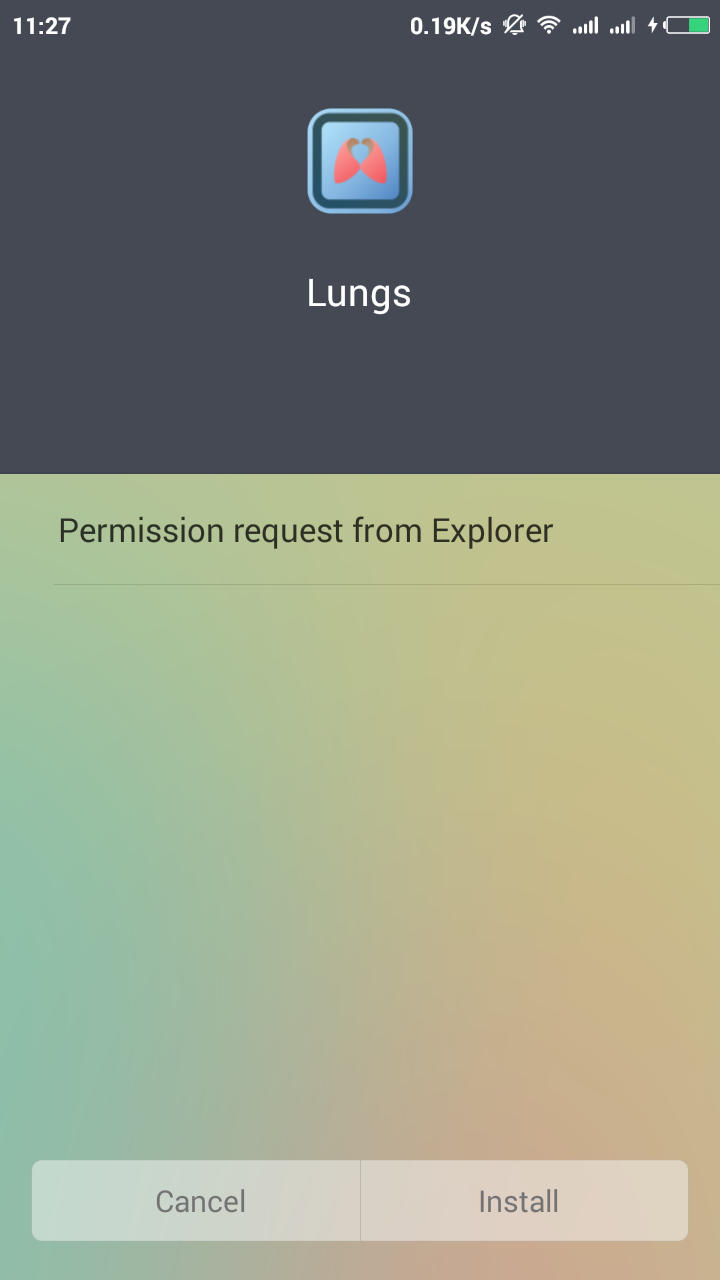
Implementasi sistem dilakukan setelah melewati tahap analisis dan perancangan sistem, pada tahap ini aplikasi siap dioperasikan pada keadaan yang sebenarnya sehingga akan diketahui program atau sistem yang dibuat dapat menghasilkan keluaran yang diinginkan.

* + 1. **Manual Instalasi**

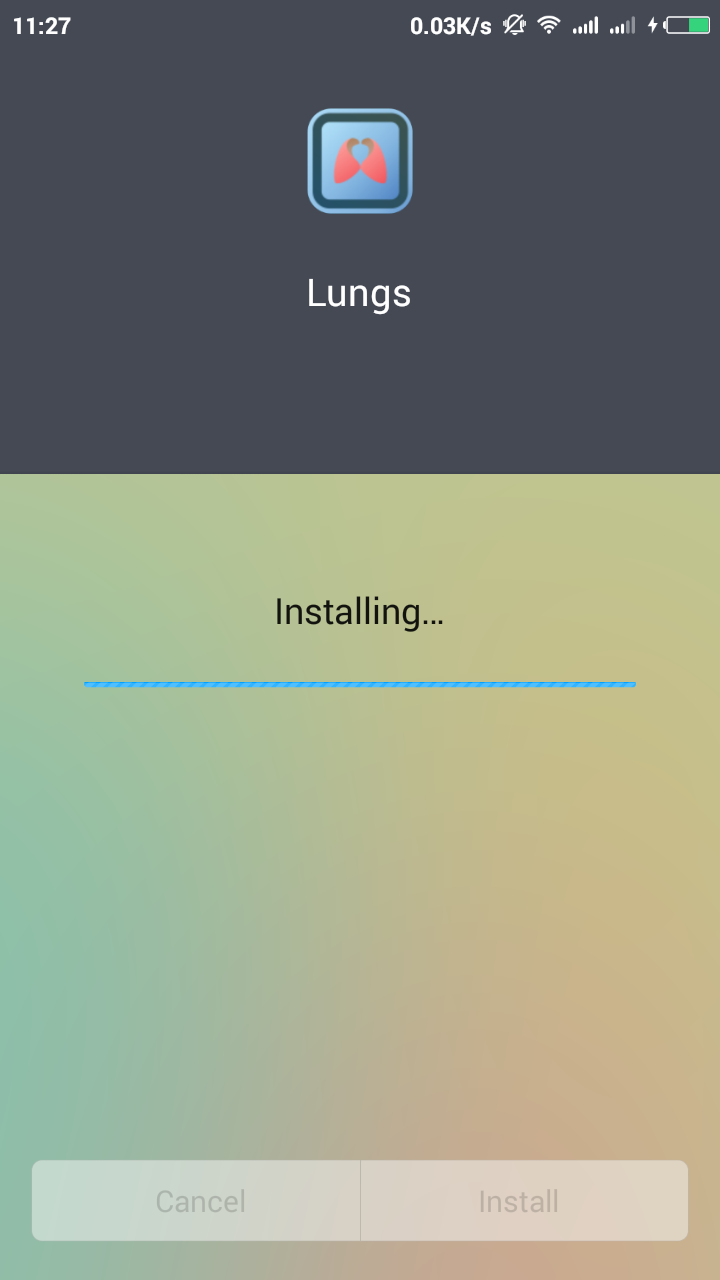
Pengguna dapat melakukan instalasi dengan men-*copy* file sistem pakar .apk ke media penyempanan pada perangkat *mobile*. Kemudian memulai instalasi dengan meng-klik file tersebut dan menyetuji *permission* yang diperlukan oleh aplikasi, kemudian proses instalasi akan berlangsung hingga selesai.



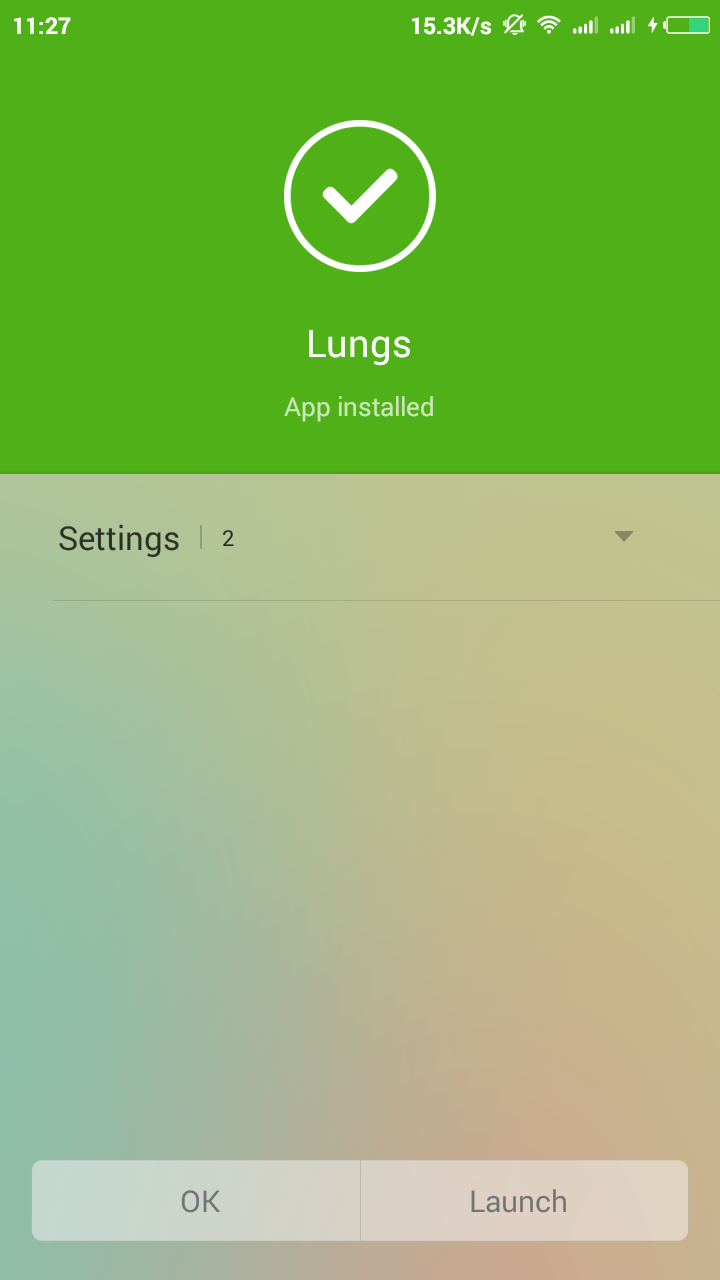
**Gambar 4.1 Manual Instalasi Pertama**



**Gambar 4.2 Manual Instalasi Kedua**



**Gambar 4.3 Manual Instalasi Ketiga**



**Gambar 4.4 Manual Instalasi Keempat**

* + 1. **Implementasi *Interface***

Implementasi *interface* atau antar muka program berperan untuk menghubungkan antara pengguna dengan aplikasi. Peranan *interface* sangat penting karena dengan *interface* yang ada pada Aplikasi Sistem Pakar untuk Diagnosa Penyakit Paru-Paru.

1. ***Interface* *Splash* *Screen***

*Splash* *Screen* merupakan tampilan awal dari aplikasi *mobile*. Pada saat membuka aplikasi *mobile* sistem pakar ini, maka modul *Splash* *Screen* yang akan muncul.



**Gambar 4.5 *Interface* *Splash* *Screen***

1. ***Interface* Menu Utama**

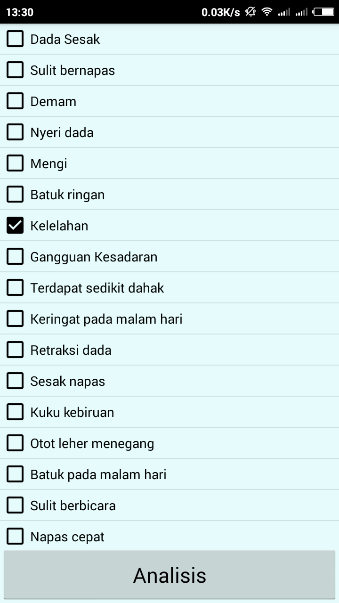
Menu utama merupakan tampilan utama dari aplikasi mobile. Dalam menu utama terdapat 3 tombol yang mengarahkan ke modul lainnya.



**Gambar 4.6 *Interface* Menu Utama**

1. ***Interface* Mulai Diagnosa**

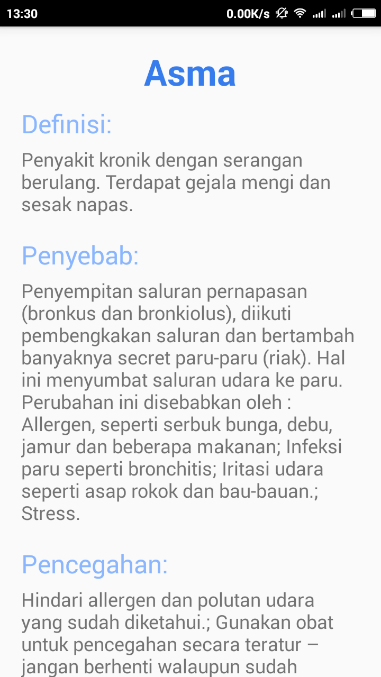
Menu Mulai Diagnosa digunakan untuk melakukan konsultasi dengan sistem, mengenai kemungkinan penyakit yang diderita oleh pasien.



**Gambar 4.7 *Interface* Menu Mulai Diagnosa**

1. ***Interface* Hasil Diagnosa**

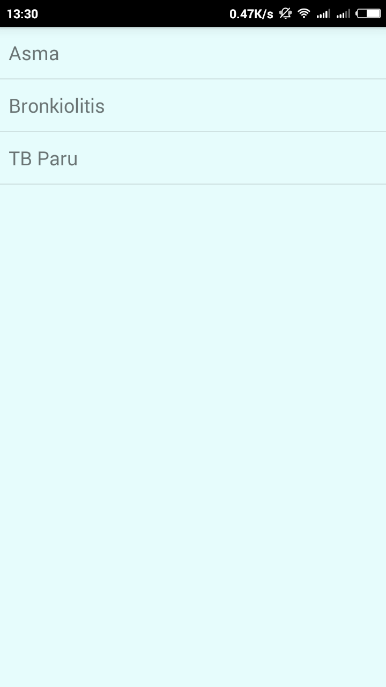
Menu hasil diagnosa akan muncul setelah semua gejala yang terjadi dipilih dalam proses diagnosa.



**Gambar 4.8 *Interface* Hasil Diagnosa**

1. ***Interface* Info Penyakit**

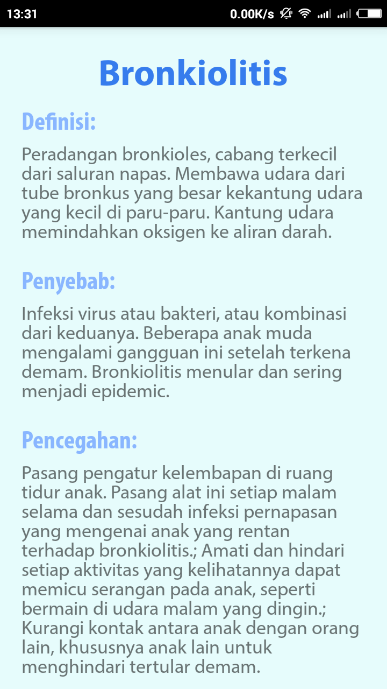
Menu info penyakit berisikan *list* atau daftar nama penyakit yang terdapat pada paru-paru.



**Gambar 4.9 *Interface* Menu Info Penyakit**

1. ***Interface* Detail Penyakit**

Menu detail penyakit menampilkan informasi penyakit dari *list* sebelumnya, dengan demikian semua info bisa diketahui.



**Gambar 4.10 *Interface* Detail Penyakit**

1. ***Interface* Tentang**

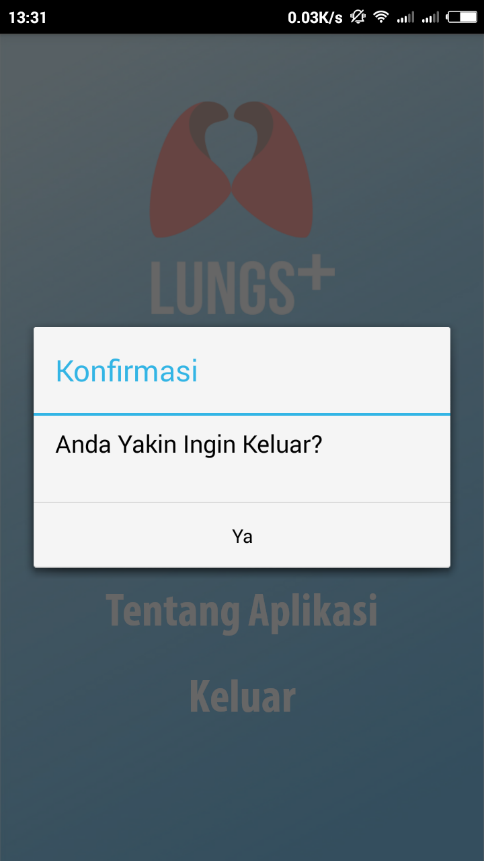
Menu tentang berisikan tentang aplikasi dan pembuat aplikasi.



**Gambar 4.11 *Interface* Menu Tentang**

1. ***Interface Exit***

Menu keluar digunakan untuk keluar dari aplikasi sistem pakar, dan akan muncul peringatan saat tombol keluar dipilih.



**Gambar 4.13 *Interface* Keluar**

* 1. **Uji Coba Sistem dan Program**

Uji coba sistem dan program dilakukan untuk mengetahui apakah sistem aplikasi dapat berjalan dengan baik sesuai yang diharapkan atau tidak. Pengujian sistem juga bertujuan untuk menghindari kesalahan yang mungkin terjadi saat pembuatan sistem aplikasi.

* + 1. ***Black*-*box* *Testing***

Metode pengujian perangkat lunak dari aplikasi yang dibuat.

**Tabel 4.1 *Black-box* *Testing* Menu Utama**

|  |  |  |
| --- | --- | --- |
| Input | Output | Status |
| Tombol Mulai Gejala | Menuju ke menu mulai gejala | Sesuai |
| Tombol Info Penyakit | Menuju ke menu info penyakit | Sesuai |
| Tombol Tentang | Menuju ke menu tentang | Sesuai |

**Tabel 4.2 *Black Box* *Testing* Menu Mulai Diagnosa**

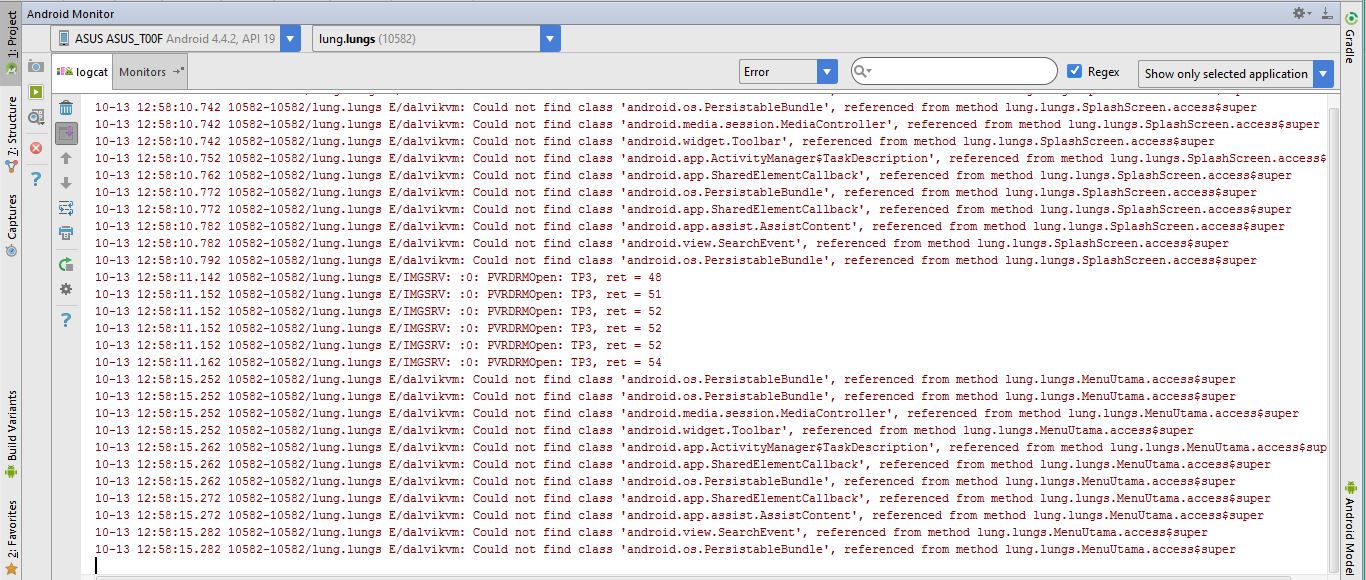
|  |  |  |
| --- | --- | --- |
| Input | Output | Status |
| Daftar Gejala | Memberikan pilihan daftar gejala penyakit | Sesuai |
| Tombol Lihat Hasil | Menuju ke menu lihat hasil, memproses data | Sesuai |

**Tabel 4.3 *Black Box* *Testing* Menu Info Penyakit**

|  |  |  |
| --- | --- | --- |
| Input | Output | Status |
| List Penyakit | Menampilkan detail penyakit paru-paru yang dicari | Sesuai |

* + 1. ***White*-*box* *Testing***

*White-box Testing* dilakukan dengan cara melihat kedalam modul untuk meneliti kode program dan untuk menganalisa, apakah terdapat kesalahan dalam program tersebut atau tidak. Dalam aplikasi *mobile*, *white box testing* dapat dilihat dari *logcat*, yang terdapat error pada program jika program tersebut tidak sesuai dengan aturan yang sesuai.



**Gambar 4.13 *White-box Testing* Aplikasi *Mobile***

* + 1. **Kesalahan Kode Program (*Syntax* *Error*)**

Kesalahan dalam penulisan program dapat terjadi apabila *programmer* menuliskan kode program yang tidak sesuai dengan aturan *syntax*. Dalam beberapa bahasa pemrograman berlaku *case* *sensitive* yaitu besar kecil huruf diperhitungkan, merupakan salah satu sebab kesalahan kode.

* + 1. **Kesalahan Proses (*Run* *Time* *Error*)**

Kesalahan proses (*run time error*) terjadi saat program dijalankan. Hal ini menyebabkan proses program berhenti sebelum selesai pada waktunya karena *compiler* menemukan kondisi yang belum terpenuhi dan tidak layak untuk dikerjakan.

* + 1. **Kesalahan Logika (*Logical* *Error*)**

Kesalahan logika (*Logical Error*) terjadi akibat kesalahan dalam logika menulis kode program. Hal ini menyebabkan hasil dari program tidak sesuai dengan yang diharapkan. Untuk mengetahui kesalahan logika adalah dengan melakukan pengecekan keseluruhan kode program sampai menemukan logika yang tidak sesuai.

* 1. **Pengujian Hasil Diagnosa**

Dari seluruh kemungkinan hasil jawaban yang diambil 15 sampel untuk menguji keakuratan hasil diagnosa sistem dengan hasil diagnosa pakar. Hasil pengujian tersebut dapat dilihat pada tabel berikut ini:

**Tabel 4.4 Pengujian Hasil Diagnosa**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | GEJALA | HASIL | | KESIMPULAN |
| SISTEM | PAKAR |
| 1 | 1. Dada Sesak 2. Mengi 3. Batuk Ringan 4. Sesak Napas | B | B | SAMA |
| NO | GEJALA | HASIL | | KESIMPULAN |
| SISTEM | PAKAR |
| 2 | 1. Batuk pada malam hari 2. Demam suhu rendah | A | A | SAMA |
| 3 | 1. Napas berbunyi 2. Kulit kebiruan 3. Penurunan berat badan | A | A | SAMA |
| 4 | 1. Retraksi dada 2. Sesak napas 3. Otot leher menegang | TB | A | TIDAK SAMA |
| 5 | 1. Sulitan bernapas 2. Dada sesak 3. Nyeri dada 4. Kelelahan 5. Batuk ringan | B | B | SAMA |
| 6 | 1. Mengi 2. Batuk dengan dahak 3. Penurunan berat badan | A | A | SAMA |
| 7 | 1. Demam suhu rendah 2. Gangguan kesadaran 3. Keringat pada malam hari 4. Kuku kebiruan | TB | TB | SAMA |
| 8 | 1. Batuk pada malam hari 2. Sesak napas 3. Retraksi dada | TB | TB | SAMA |
| NO | GEJALA | HASIL | | KESIMPULAN |
| SISTEM | PAKAR |
| 9 | 1. Demam 2. Mengi | B | B | SAMA |
| 10 | 1. Batuk dengan dahak 2. Keringat pada malam hari 3. Demam | TB | TB | SAMA |
| 11 | 1. Napas berbunyi 2. Sesak napas 3. Sulit berbicara | TB | A | TIDAK SAMA |
| 12 | 1. Retraksi dada 2. Gangguan kesadaran | B | B | SAMA |
| 13 | 1. Keringat pada malam hari 2. Terdapat sedikit dahak 3. Sesak napas | TB | TB | SAMA |
| 14 | 1. Sulit berbicara 2. Napas yang cepat 3. Sesak Napas 4. Terdapat sedikit dahak | TB | TB | SAMA |
| 15 | 1. Penurunan berat badan 2. Demam suhu rendah 3. Sulit berbicara | TB | TB | SAMA |

Keterangan :

A = Asma

B = Bronkiolitis

TB = Tuberkulosis Paru

Dari tabel 4.4 dapat dilihat bahwa dari 15 sampel menghasilkan 13 hasil yang sama dan 1 hasil yang berbeda dengan hasil di lapangan. Maka dari itu pengujian sampel di atas sistem ini menghasilkan akurasi ketepatan sebesar 86.67% dan dikatakan layak oleh pakar.

Nilai akurasi = 13/15 \* 100% = 86.67%

* 1. **Implementasi Kode Program**

Implementasi kode program adalah suatu tahapan dalam pembuatan sistem yang bertujuan agar sistem yang digunakan sesuai dengan rancangan yang telah dibuat. Tujuan dari implementasi kode program ini adalah untuk memperjelas pembaca tentang program-program yang digunakan dalam pembuatan aplikasi. Pada implementasi kode program, akan dibagi kedalam beberapa *class* dasar diantaranya, *Class* MenuUtama merupakan menu utama yang memuat tombol untuk menjalankan aktivitas selanjutnya, *Class* *DBHelper* berfungsi sebagai pengontroluntuk mengambil data dari *database* yang dibutuhkan pada *class* lain, *Class* MulaiDiagnosa digunakan oleh user untuk memilih gejala yang dirasakan dan berisi tentang logika perhitungan *certainty* *factor*, *Class* HasilAkhir menampilkan informasi penyakit yang berasal dari hasil dari perhitungan *certainty* *factor* berdasarkan gejala yang terpilih, untuk lebih jelasnya bisa dilihat pada lampiran 4.

**BAB V**

**PENUTUP**

* 1. **Kesimpulan**

Berdasarkan hasil pembahasan perancangan aplikasi sistem pakar berbasis *mobile*, maka diambil kesimpulan sebagai berikut:

1. Aplikasi sistem pakar ini berbasis *mobile* yang terkoneksi dengan *database* untuk melakukan proses diagnosa sesuai gejala yang diketahui.
2. Aplikasi sistem pakar ini menggunakan metode *certainty* *factor* yang melakukan proses diagnosa dengan gejala-gejala yang diketahui untuk memperoleh keputusan.
3. Aplikasi sistem pakar ini melakukan pengumpulan data penyakit paru-paru, aplikasi ini ditujukan untuk pengguna yang ingin mengetahui tentang penyakit paru-paru.
   1. **Saran**

Dibutuhkan kritik atau saran demi terciptanya aplikasi yang lebih baik dan dapat dimanfaatkan kedepannya. Saran yang ingin penulis sampaikan antara lain:

1. Penambahan fitur *chatting* *online* antara pengguna dengan dokter spesialis paru-paru atau pakar yang ahli dalam bidang paru-paru.
2. Penambahan data penyakit paru-paru.
3. Membuat aplikasi menjadi lebih interaktif lagi seperti menambahkan gambar atau animasi didalamnya.

# **DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | Soft Ilmu, "Soft Ilmu," October 2015. [Online]. Available: http://www.softilmu.com/2015/10/Pengertian-Fungsi-Struktur-Paru-Paru-Adalah.html. [Accessed September 2016]. |
| [2] | D. H. T. Rab, "Ilmu Penyakit Paru," in *Asthma*, Jakarta, Hipokrates, 1996, p. 165. |
| [3] | H. W. G. M.D., "Buku Pintar Kesehatan 796 Gejala, 520 Penyakit, 160 Pengobatan," in *Asthma*, Jakarta, Arcan, 1994, p. 156. |
| [4] | D. H. T. Rab, "Ilmu Penyakit Paru," in *Tuberculosis Paru*, Jakarta, Hipokrates, 1996, p. 236. |
| [5] | H. W. G. M.D., "Buku Pintar Kesehatan 796 Gejala, 520 Penyakit, 160 Pengobatan," in *Tuberculosis*, Jakarta, Arcan, 1994, p. 610. |
| [6] | International Child Health, "Bronkiolitis," [Online]. Available: http://www.ichrc.org/441-bronkiolitis. [Accessed September 2016]. |
| [7] | H. W. G. M.D., "Buku Pintar Kesehatan 796 Gejala, 520 Penyakit, 160 Pengobatan," in *Bronkiolitis*, Jakarta, Arcan, 1994, p. 182. |
| [8] | S. Setiawan, Artificial Intelligence, Yogyakarta: ANDI Offset Yogyakarta, 1993. |
| [9] | S. Kusumadewi, Artificial Intelligence Teknik dan Aplikasi, Yogyakarta: Graha Ilmu, 2003. |
| [10] | M. Arhami, Konsep Dasar Sistem Pakar, Yogyakarta: ANDI Offset Yogyakarta, 2005. |
| [11] | S. Kusrini, Sistem Pakar, Teori dan Aplikasi, Yogyakarta: ANDI Offset Yogyakarta, 2006. |
| [12] | M. Kusrini, Aplikasi Sistem Pakar, Yogyakarta: ANDI Offset Yogyakarta, 2008. |
| [13] | H. A. Fattah, Analisis dan Perancangan Sistem Informasi, Yogyakarta: ANDI Offset Yogyakarta, 2007. |
| [14] | F. Rangkuti, Analisis SWOT Teknik Membedah Kasus Bisnis, Jakarta: PT. Gramedia Pustaka Utama, 2006. |
| [15] | D. Suprianto and S. M. Rini Agustina, Pemrograman Aplikasi Android, MediaKom, 2012. |
| [16] | Y. Supardi, Sistem Operasi Andal Android, 2012. |
| [17] | M. Kusrini, Strategi Perancangan dan Pengelolaan Basis Data, Yogyakarta: ANDI Offset Yogyakarta, 2007. |
| [18] | Fathansyah, "Basis Data Informatika," Bandung, 2004, p. 2. |
| [19] | A. Nugroho, Perancangan dan Implementasi Sistem Basis Data, Yogyakarta: ANDI Offset Yogyakarta, 2011. |
| [20] | Munawar, Pemodelan Visual Dengan UML, Yogyakarta: Graha Ilmu, 2005. |
| [21] | B. Hernawan, Menguasai JAVA 2 dan Object Oriented Programming, Yogyakarta: ANDI Offset Yogyakarta, 2007. |
| [22] | S. B. Sakur, PHP 5 Pemrograman Berorientasi Objek - Konsep dan Implementasi, Yogyakarta: ANDI Offset Yogyakarta, 2011. |
| [23] | N. S. H, "Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android Edisi Revisi," 2012, p. 5. |
| [24] | Android Studio, "Mengenal Android Studio," [Online]. Available: https://developer.android.com/studio/intro/index.html?hl=id. |