

BAB I

PENDAHULUAN

1.1 Latar Belakang

Anak perlu diperhatikan kesehatannya dikarenakan lemahnya ketahanan tubuh membuat Anak sangat rentan terhadap kuman, bakteri dan penyakit. Sering kali anak mengalami kesulitan dalam menjelaskan keluhanannya, oleh karena itu orang tua diharapkan dapat menangkap setiap bahasa tubuh dari anak mereka.

Penyakit pada anak sangat lah berbahaya karena penyakit yang menyerang anak umumnya bisa terjadi komplikasi, untuk menegakkan diagnosis penyakit terutama pada anak perlu kejelian dalam pemeriksaan, bahkan diperlukan dokter spesialis penyakit anak. Namun hampir setiap daerah belum mempunyai dokter spesialis anak yang cukup, sehingga sering terjadi keterlambatan dalam penanganannya.

Desa Kait Kait Kecamatan Bati-Bati Kabupaten Tanah Laut Kalimantan Selatan, sebuah desa kecil yang berjarak sekitar 90 KM dari Ibukota Provinsi (Banjarmasin) yang dimana jalur darat yang harus ditempuh ke Desa ini sangat minim karena kurangnya pembangunan dari pemerintah setempat, Menurut Pengamatan Peneliti yang telah tinggal di Desa ini kurang lebih 20 tahun bahwa masyarakat Setempat hampir tak ada yang memiliki Komputer maupun *Notebook* jika ada itu banyak digunakan di bagian sipil saja, meski demikian pengguna Gadget seperti Smartphone dan Handphone banyak digunakan masyarakat setempat karena memang beberapa tahun terakhir ini outlet penjualan *Gadget* ini mulai dibuka diperdesaan tersebut.

Sistem pakar adalah sebuah perangkat lunak komputer yang memiliki basis pengetahuan untuk tujuan tertentu dan menggunakan penalaran yang menyerupai seorang pakar dalam memecahkan masalah.[1]

Algoritma *best first search* ini merupakan kombinasi dari algoritma *depth first search* dengan algoritma *breadth first search* dengan mengambil kelebihan dari kedua algoritma tersebut. Apabila pada pencarian dengan algoritma *hill climbing* tidak diperbolehkan untuk kembali ke node pada level yang lebih rendah meskipun node di level yang lebih rendah tersebut memiliki nilai heuristik yang lebih baik, lain halnya pada algoritma *best first search*, pencarian diperbolehkan mengunjungi node yang ada di level yang lebih rendah, jika ternyata node di level yang lebih tinggi memiliki nilai heuristik yang lebih buruk.

Android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. *Android* merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, *i-Phone OS*, *Symbian*, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka [2]

Dengan Menggunakan Teknologi *Android* Sekarang dan perkembangan UI/UX, sehingga membangun Sistem ini sangat cocok dengan berbasis *Android*

karena masyarakat sekarang sering sekali menggunakan Smartphone kapanpun dan dimanapun, dengan dibangunnya sistem ini di Android maka akan mempermudah *user* dalam menggunakan

Diharapkan Dengan Sistem Pakar Ini, mampu mempermudah dalam mendeteksi penyakit yang menyerang Anak.

1.2 Rumusan Masalah

Dalam proses membangun aplikasi ini dibutuhkan suatu teknologi informasi yang relevan untuk mencapai solusi tersebut, maka dapat ditemukan masalah yang akan diselesaikan dalam proposal ini adalah :

1. Bagaimana merancang Sistem Pakar untuk mendeteksi penyakit anak?
2. Bagaimana implementasi metode *Best First search* ?
3. Bagaimana sistem bisa memberikan informasi berupa nama penyakit dari hasil diagnosa ?

1.3 Batasan masalah

Adapun batasan-batasan masalah dalam penelitian ini diantaranya antara lain adalah :

1. Sistem pakar ini berbasis android.
3. Sistem pakar ini mendiagnosis Anak umur 5 hingga 10 tahun
4. Sumber pengetahuan diperoleh dari pakar
5. Metode yang digunakan adalah *Best First Search*
6. Penyakit yang dibahas adalah penyakit umum yang biasanya diderita anak

7. Interaksi antara sistem dan user menggunakan pertanyaan berupa gejala yang dirasakan *user*, dimana *user* akan diminta untuk memilih gejala pada setiap pertanyaan berdasarkan kondisi anak tersebut.

1.4 Maksud dan Tujuan Penelitian

Penelitian Ini Bermaksud Membuat sebuah Aplikasi Berbasis Android, yaitu Aplikasi Sistem Pakar yang dimana memiliki tujuan yaitu :

1. Membangun suatu sistem yang dapat memberikan sebuah diagnosa untuk mendeteksi penyakit anak.
2. Mengimplementasikan metode *Best First Search* dalam memberikan hasil diagnose penyakit anak.
3. Menyajikan solusi/penangan dari penyakit anak yang umum ditemui

1.5 Manfaat Penelitian

1.5.1 Bagi Mahasiswa

Manfaat bagi mahasiswa adalah agar bisa membangun sebuah sistem yang mampu memberikan gambaran tentang ilmu pengetahuan seputar pengembangan sistem pakar, dan juga implementasi sebuah metode atau algoritma kedalam sebuah sistem

1.5.2 Bagi Dokter

Manfaat bagi Dokter adalah agar dokter bisa mengidentifikasi penyakit yang mungkin dalam diagnosa sang dokter sendiri ada keraguan, maka dengan sistem ini diharapkan mampu memberikan keputusan dalam diagnosa yang lebih maksimal

1.6 Metode Penelitian

1.6.1 Metode Pengumpulan Data

Teknik pengumpulan data yang dilakukan dalam penyusunan laporan skripsi ini adalah sebagai berikut :

1.6.1.1 Metode Studi Pustaka

Metode ini dilakukan dengan mengumpulkan data-data dengan menanyakan langsung ke pakar tentang gejala dan penyakit yang biasanya diderita oleh anak-anak berusia sekitar 5 hingga 10 tahun, penyakit umum yang diambil adalah penyakit yang sering pakar temui di Studi kasus pada Skripsi ini.

1.6.1.2 Metode Observasi

Metode ini dilakukan dengan melakukan ke beberapa aplikasi sejenis seperti aplikasi *Android* maupun berbasis *WEB* yang dimana punya konsep sistem pakar .

1.6.2 Metode Analisis

Metode analisis perancangan sistem informasi ini menggunakan metode *System Development Life Cycle* (SDLC) dengan tahapan-tahapan sebagai berikut:

1. Perencanaan Sistem (*Systems Planning*)

Lebih menekankan pada aspek studi kelayakan pengembangan sistem. Aktivitas perencanaan sistem meliputi :

- a. Mendefinisikan tujuan dan ruang lingkup pengembangan.
- b. Mengidentifikasi apakah masalah-masalah yang ada bisa diselesaikan melalui pengembangan sistem.
- c. Menentukan dan evaluasi strategi yang akan digunakan dalam pengembangan sistem.

- d. Penentuan prioritas teknologi dan pemilihan aplikasi.

2. Analisis Sistem (*Systems Analysis*)

Analisa sistem adalah tahap di mana dilakukan beberapa aktivitas berikut:

- a. Melakukan studi literatur untuk menemukan suatu kasus yang bisa ditangani oleh sistem.
- b. Melakukan analisa kebutuhan dan mendefinisikan kebutuhan pada sistem.

3. Perancangan Sistem (*Systems Design*)

Pada tahap ini, operasi-operasi pada sistem dideskripsikan secara detail.

Aktivitas yang dilakukan meliputi:

- a. Merancang skema *database*.
- b. Merancang antar muka sistem.

4. Implementasi Sistem (*Systems Implementation*)

Tahap berikutnya adalah mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba. Dalam implementasi, dilakukan aktivitas-aktivitas sebagai berikut:

- a. Pembuatan *database* sesuai skema rancangan.
- b. Pembuatan aplikasi berdasarkan antar muka sistem.
- c. Pengujian dan perbaikan aplikasi (*debugging*).

5. Pemeliharaan Sistem (*Systems Maintenance*)

Dilakukan oleh administrator yang ditunjuk untuk menjaga sistem tetap mampu beroperasi secara benar melalui kemampuan sistem dalam mengadaptasikan diri sesuai dengan kebutuhan.[7].

1.6.3 Metodologi Perancangan

Proses perancangan sistem dalam penelitian ini menggunakan permodelan waterfall dimana pembangunan sistem dilakukan secara linier atau berurutan dari tahapan awal hingga tahapan akhir.

Tahapan awal dilakukan dengan mengumpulkan data-data yang sesuai dengan kebutuhan sistem yang akan dibangun.

Tahapan kedua adalah melakukan desain sistem dengan menerjemahkan syarat kebutuhan maupun data-data yang ada kedalam sebuah perancangan perangkat lunak yang berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail prosedural.

Tahapan ketiga adalah melakukan penulisan kode program sesuai dengan prosedur dan fungsi-fungsi yang diperlukan dalam pembangunan sistem.

Tahapan keempat adalah melakukan uji coba sistem pakar penyakit yang telah dibangun dengan melalui beberapa tahapan sebelumnya. Tahapan ini dilakukan untuk memastikan apakah sistem yang telah dibangun dapat berjalan dengan baik tanpa adanya *error* dan apakah sistem pakar yang dibangun mampu memberikan hasil/diagnosa yang konkrit dari hasil inputan.

Tahapan terakhir adalah melakukan implementasi pada sistem pakar penyakit anak yang telah dibangun dan melakukan pemeliharaan sistem dimana ketika terjadi kendala-kendala dalam sistem tersebut, penulis selaku pengembang harus mampu memperbaiki kerusakan dan melakukan perbaruan pada sistem pakar Penyakit anak tersebut.

1.6.4 Metode Testing

Metode yang digunakan pada penelitian ini adalah metode *black-box testing* dan *white-box testing* untuk menguji dan mengetahui *bugs* maupun *error* yang terdapat dalam sistem sebelum sistem masuk ke-tahap produksi serta menguji apakah alur kode program algoritma yang digunakan sudah terpenuhi.[7]

1.7 Sistematika Penulisan

Materi-materi yang tertera pada laporan skripsi ini dikelompokkan menjadi beberapa sub bab dengan sistematika penyampaian sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisikan pembahasan materi dimana sebagian besar berupa penguraian dari seluruh latar belakang, perumusan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini berisikan teori yang berupa pengertian dan definisi yang diambil dari buku, karya ilmiah lain, serta tinjauan dari penelitian sebelumnya

BAB III : ANALISIS DAN PERANCANGAN

Bab ini berisi analisis kebutuhan dan rancangan UML dan ERD

BAB IV : IMPLEMENTASI DAN PEMBAHASAN

Bab ini menjelaskan mengenai desain antar muka, analisis studi kasus, hasil testing dan Implementasi metode pada sistem.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran, yang akan berguna dan dapat dimanfaatkan untuk penelitian selanjutnya .

DAFTAR PUSTAKA

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penulis ingin melakukan peningkatan kemudahan untuk mengetahui dengan cepat penyakit pada anak, Karena kebiasaan orang tua panik saat anaknya tiba tiba sakit tanpa ada ilmu kenapa anaknya sakit, sehingga perlu ada system untuk mempermudah orang tua mengetahui apa yang terjadi pada anaknya [1] Untuk mengatasi permasalahan tersebut, diusulkan sistem informasi dengan menggunakan sistem kecerdasan buatan. Dimana sistem ini dapat membantu dokter yang belum berpengalaman dalam mendukung diagnosa dokter terhadap jenis penyakit mata glaukoma. Sehingga dokter dapat melakukan diagnosa secara spesifik. [2]

Di Tinjauan yang lain Penulis ingin membuat sebuah system yang dimana diperuntukan bagi rumah sakit yang kekurangan dokter spesialis Kanker darah, karena di Studi kasus penulis yang menjelaskan bagaimana bahayanya kanker darah pada anak jika tak cepat diatasi sedangkan untuk mendiagnosa sendiri perlu dokter ahli sehingga system pakar akan sangat membantu dalam mendiagnosa penyakit ini[3] tapi dalam *BEST FIRST SEARCH* data yang digunakan cukup banyak sehingga diperlukan Kombinasi dengan *BEST FIRST SEARCH* yang menggunakan data yang lebih sedikit [2]

Penulis membangun system ini karena rentannya balita dan bayi dalam terserang penyakit, karena kelemahan fisik inilah penulis membangun system ini untuk memudahkan orang tua dalam mengetahui penyakit apa yang terjadi pada

anaknya. Tetapi tak ada pembahasan lebih lanjut tentang penyakit penyakit apa saja yang bisa dikenali [4] maka sebab itu lah diperlukan *memory* yang cukup besar agar system bisa belajar sendiri ketika mengenali kasus yang belum dikenali [5]

Tabel 2.1 Penelitian Sebelumnya

NO	JUDUL PENELITIAN	PERBANDINGAN FITUR
1	Sistem Pakar Untuk Mendiagnosa Penyakit Pada Balita Berbasis <i>DFS</i>	<ol style="list-style-type: none"> 1. User digunakan oleh banyak kalangan termasuk orang awam 2. Hanya untuk mendiagnosa penyakit balita umur 2 – 5 tahun
2	Sistem Pakar Penyakit Mata Glaukoma Dengan Metode <i>Breathd First Search</i>) dan <i>Fuzzy Tsukamoto</i>	<ol style="list-style-type: none"> 1. Sistem dibuat dan hanya digunakan oleh pihak rumah sakit 2. Sistem pakar khusus penyakit mata glaucoma
3	Diagnosis Kanker Darah pada Anak menggunakan Inferensi <i>Forward Chaining</i>	<ol style="list-style-type: none"> 1. Hanya digunakan oleh dokter ahli 2. Hanya mendiagnosis penyakit kanker darah pada anak

2.2 Konsep Dasar Kecerdasan Tiruan

Artificial Intelligence merupakan bagian tengah atau inti dari ilmu komputer dan merupakan salah satu ilmu dasar dari ilmu komputer yang harus dipahami, bila kita ingin membuat perangkat lunak dengan penerapan konsep *Artificial Intelligence* dalam memecahkan persoalan yang ada di dunia ini, sehingga

keberadaannya mutlak harus ada. Karena kecerdasan tiruan adalah ilmu yang berdasarkan proses manusia berpikir, maka penelitian bagaimana proses manusia berpikir adalah hal yang pokok.

Semua proses berpikir menolong manusia untuk menyelesaikan sesuatu masalah. Pada saat otak manusia mendapat informasi dari luar, maka suatu proses berpikir memberikan petunjuk tindakan atau respon apa yang dilakukan. Hal ini merupakan suatu reaksi otomatis dan respon yang spesifik dicari untuk menyelesaikan masalah tertentu. Hasil akhir dari semua proses berpikir tersebut disebut tujuan (goal). Berdasarkan fakta-fakta yang ada keterangan diatas, maka komputer dapat digunakan sebagai alat bantu dalam mengambil keputusan.

2.3 Sistem pakar

2.3.1. Definisi sistem pakar

Sistem pakar atau sistem berbasis pengetahuan adalah yang paling banyak aplikasinya dalam membantu menyelesaikan masalah-masalah dalam dunia nyata. Perangkat lunak ini dapat sekali dijalankan oleh perangkat komputer pribadi, sehingga untuk aplikasi kecerdasan tiruan ini dapat dilakukan dengan mudah dan dengan biaya yang relatif lebih murah. Sistem pakar adalah sebuah program yang dapat :

- 1) Menangani masalah dunia nyata, masalah yang kompleks yang sangat membutuhkan interpretasi pakar.
- 2) Menyelesaikan masalah dengan menggunakan komputer dengan model penalaran manusia dan mencapai kesimpulan yang sama dengan yang dicapai oleh

seorang pakar jika berhadapan langsung dengan sebuah masalah yang sedang di hadapinya.

Komputer yang berbasis sistem pakar adalah program komputer yang mempunyai pengetahuan yang berasal dari manusia yang berpengetahuan luas dalam suatu domain tertentu, dimana pengetahuan di sini adalah pengetahuan manusia yang sangat minim penyebarannya, mahal dan serta susah untuk di dapatkannya.

Kondisi-kondisi dimana sistem pakar dapat membantu manusia dalam menyelesaikan masalahnya, antara lain:

- 1) Kebutuhan tenaga ahli yang banyak, tetapi pakar yang tersedia jumlahnya hanya sedikit dan sangat terbatas.
- 2) Pemakaian pakar yang berlebihan dalam membuat keputusan, walaupun dalam suatu tugas yang rutin.
- 3) Pertimbangan kritis harus dilakukan dalam waktu yang singkat untuk menghindari hal-hal yang tidak diinginkan.
- 4) Hasil yang optimal, seperti dalam hal perencanaan atau konfigurasi.
- 5) Sejumlah besar data yang harus diteliti oleh pakar secara berkelanjutan.

2.3.2 Pembentukan umum expert system

Dengan metode *Forward-Chaining*, data-driven karena inference engine menggunakan informasi yang ditentukan oleh user untuk memindahkan keseluruhan jaringan dari logika 'AND' dan 'OR' sampai sebuah terminal di tentukan sebagai obyek.

Bila inference engine tidak dapat menentukan obyek maka akan meminta informasi lain. Aturan (*Rule*) di mana menentukan obyek, membentuk lintasan (*Path*) yang mengarah ke obyek. Oleh karna itu, hanya satu cara untuk mencapai obyek adalah memenuhi semua aturan.

2.3.3 Ciri-ciri sistem pakar

Ciri-ciri dari sistem pakar adalah:

- 1) Terbatasnya pada domain keahlian tertentu.
- 2) Dapat memberikan penalaran data yang tidak pasti.
- 3) Dapat mengemukakan rangkaian alasan-alasan yang tidak diberikannya dengan cara yang tidak dapat di pahami.
- 4) Berdasarkan kaidah-kaidah/ketentuan/rule yang berlaku.
- 5) Dirancang untuk dapat dikembangkan secara bertahap.
- 6) Pengetahuan dan mekanisme penalaran jelas terpisah.
- 7) Keluaran bersifat anjuran.
- 8) Sistem dapat mengaktifkan kaidah secara searah yang sesuai di tuntun oleh dialog dengan user.

2.3.4 Klasifikasi sistem pakar

1) Sistem Pakar Diagnosis

Digunakan untuk melakukan pencarian penyelesaian masalah atau kerusakan suatu alat atau mesin tertentu. Prinsipnya adalah menemukan masalah atau kerusakan apa yang terjadi. Sistem ini merupakan jenis sistem pakar yang paling populer saat ini, melakukan diagnosis, menggunakan deskripsi keadaan karakteristik tingkah laku, atau pengetahuan tentang pembuatan komponen sehingga dapat menentukan kemungkinan kerusakan pada sistem.

Contoh: Diagnosis medis kedokteran, diagnosis kerusakan alat elektronik dan perancangan sistem komunikasi dan radio.

2) Sistem Pakar Pengajaran

Digunakan untuk mengajar, mulai dari murid Sekolah Dasar sampai mahasiswa Perguruan Tinggi. Kelebihan dari sistem pakar yang digunakan untuk mengajar adalah membuat diagnosa penyebab kekurangan dari seorang siswa, kemudian memberi cara untuk memecahkannya.

Contoh: Contoh sistem pakar untuk pengajaran bahasa.

3) Sistem Pakar Interpretasi

Digunakan untuk menganalisa data yang tidak lengkap, tidak teratur dan data yang kontradiktif, misalnya untuk interpretasi pengawasan, pengertian bahasa, analisis citra, dan lain-lain.

Contoh: sistem pakar interpretasi lingkungan pengendapan delta.

4) Sistem Pakar Prediksi

Digunakan untuk melakukan peramalan suatu keadaan dengan ditunjang oleh data yang diperoleh sebelumnya, sistem ini dapat memberikan kemungkinan penyelesaian tertentu.

Contoh: Prediksi Penyebab dan Solusi Ketidaknyamanan Kerja dengan Aplikasi Sistem Pakar.

5) Sistem Pakar Perencanaan

Merupakan suatu sistem yang sangat luas mulai dari perencanaan mesin-mesin sampai manajemen bisnis. Penggunaan sistem pakar jenis ini menghemat biaya, waktu, dan material, sebab pembuatan model sudah tidak diperlukan lagi.

Contoh: Perencanaan Sistem Distribusi Menggunakan Sistem Pakar.

6) Sistem Pakar Kontrol

Digunakan untuk mengendalikan kegiatan yang membutuhkan presisi waktu yang tinggi. Sistem ini memperhatikan tingkah laku sistem yang dapat disebut normal atau tidak normal. Sistem ini bergantung pada waktu untuk menginterpretasikan tingkah laku yang diamati. Misalnya pada pengendali manajemen bisnis dan pengendali pada industri berteknologi tinggi.

2.3.5. Keuntungan Sistem Pakar

Secara garis besar, banyak manfaat yang dapat di ambil dengan adanya sistem pakar, antara lain:

- 1) Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
- 2) Bisa melakukan proses secara berulang-ulang secara otomatis.
- 3) Menyiapkan pengetahuan dan keahlian para pakar.
- 4) Meningkatkan output dan produktifitas.

- 5) Meningkatkan kualitas.
- 6) Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
- 7) Mampu beroperasi dalam lingkungan yang berbahaya.
- 8) Memiliki kemampuan untuk mengakses pengetahuan
- 9) Memiliki reliabilitas.
- 10) Meningkatkan kapabilitas sistem komputer.
- 11) Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dengan mengandung ketidakpastian.
- 12) Sebagai media pelengkap dalam pelatihan.
- 13) Meningkatkan kapabilitas dalam penyelesaian masalah.
- 14) Menghemat waktu dalam pengambilan suatu keputusan.

Tabel 2.2 Perbedaan Sistem Konvensional dan Sistem Pakar

Sistem Konvensional	Sistem Pakar
1. Informasi dan pemrosesan umumnya digabung dalam suatu program yang suquential 2. Program tidak pernah salah (kecuali karena programernya) 3. Tidak menjelaskan kenapa input dibutuhkan atau bagaimana hasil diperoleh 4. Membutuhkan semua input data	1. Knowledge base terpisah dari mekanisme pemrosesan 2. Program bisa saja melakukan kesalahan 3. Penjelasan (Explanation) merupakan bagian dari Expert system 4. Tidak harus membutuhkan data input atau data fakta

5. Perubahan pada program sangat merepotkan	5. Perubahan pada rule dapat dilakukan dengan mudah
6. Sistem bekerja jika sudah lengkap	6. Sistem dapat bekerja pada rule yang sedikit
7. Eksekusi secara algoritma (step by step)	7. Eksekusi dilakukan secara heuristik dan logik
8. Manipulasi efektif pad data base yang besar	8. Manipulasi efektif pada knowledge-base yang besar
9. Efesiensi adalah tujuan utama	9. Efektifitas adalah tujuan utama
10. Data kuantitatif	10. Data kualitatif
11. Representasi data dalam numerik	11. Representasi pengetahuan dalam simbol
12. menangkap, menambah dan mendistribusikan data numerik atau informasi	12. Menangkap, menambah, dan mendistribusi pertimbangan (judgment) dan pengetahuan

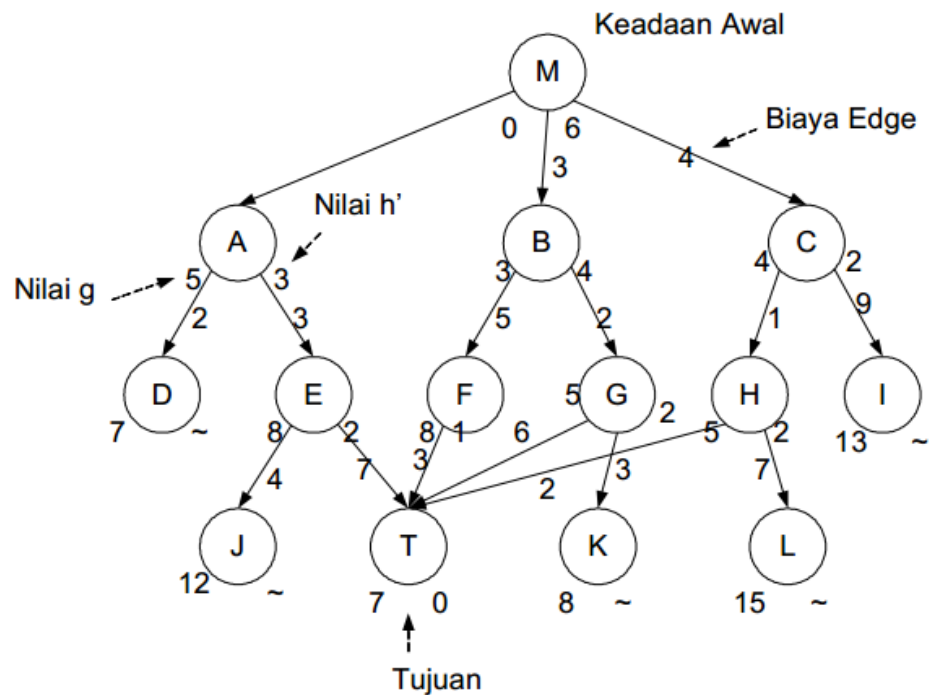
2.4 Best First Search

Best first search merupakan kombinasi dari metode *Depth First Search* dan *Breadth First Search* dimana pencarian diperbolehkan mengunjungi node pada level lebih rendah jika node pada level lebih tinggi memiliki nilai heuristik lebih buruk. Algoritma pada *best first search* yaitu :

1. Buat sebuah stack kemudian inisialisasikan node akar sebagai node pertama.

2. Bila node pertama \neq GOAL, node diganti dan dihapus dengan anak-anaknya.
3. Kemudian, keseluruhan node pada stack diurutkan secara ascending berdasarkan fungsi heuristik yang digunakan.
4. Bila node pertama \neq GOAL, ulangi langkah 2. Bila node pertama = GOAL, cari solusi dengan menelusuri jalur dari GOAL ke node akar.

Teknik ini akan mengunjungi dimulai dari node terdalam selama node tersebut merupakan node yang terbaik. Jika node yang sedang dikunjungi ternyata tidak mengarah kepada solusi yang diinginkan, maka akan dilakukan runut balik ke arah node akar untuk mencari node anak lainnya yang lebih menjanjikan daripada node yang terakhir dikunjungi. Bila tidak ada juga, maka akan terus mengulang mencari ke arah node akar sampai ditemukan node, yang lebih baik untuk dibangkitkan suksesornya. Teknik ini bertolak belakang dengan teknik *Depth First Search* yang mencari sampai kedalaman yang, terdalam sampai tidak ada node suksesor yang bisa dibangkitkan sebelum melakukan runut balik, dan *Breadth First Search* yang tidak akan melakukan pencarian secara mendalam sebelum pencarian secara melebar selesai. Dalam pengembangan Sistem Pakai ini, penulis akan menggunakan teknik *Best First Search*.



Gambar 2.1 Contoh Alur *Best First Search*

2.5 Platform Android

Android merupakan salah satu sistem operasi yang terkenal dikalangan perangkat mobile yang merupakan pesaing dari sistem operasi perangkat *mobile* lainnya seperti *Windows Phone*, *iOS*, *BlackBerry*, *MeeGo*, *Bada* dan *Symbian*. Namun berbeda dengan sistem operasi mobile lainnya, karena *Android* bersifat *Open Source* yang memungkinkan untuk dikembangkan lebih lanjut oleh pihak ketiga.

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* diakusisi oleh Google pada Juli 2005, dan baru dirilis perdana pada 5 November 2007. *Android* berlisensi di bawah GNU, General Public Lisensi Versi 2 (GPLv2), yang

memperbolehkan pihak ketiga untuk mengembangkannya dengan menyertakan term yang sama. Pendistribusiannya di bawah Lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Android dirancang dengan arsitektur sebagai berikut

1. *Application dan Widgets*, merupakan layer dimana kita berhubungan dengan aplikasi saja, seperti aplikasi untuk browsing. Selain itu, fungsi-fungsi seperti telepon dan sms juga terdapat pada layer ini.
2. *Application Frameworks*, merupakan layer dimana para pembuat aplikasi melakukan pengembangan/ pembuatan aplikasi yang akan dijalankan di sistem operasi Android. Beberapa komponen yang terdapat pada layer ini adalah, Views, Content Provider, Resource Manager, Notification Manager dan Activity Manager.
3. *Libraries*, merupakan *layer* dimana fitur-fitur Android berada yang dapat digunakan untuk menjalankan aplikasi. Library yang disertakan seperti library untuk pemutaran audio dan video, tampilan, grafik, SQLite, SSL dan Webkit, dan 3D.
4. *Android Run Time*, merupakan layer yang berisi *Core Libraries* dan *Dalvik Virtual Machine* (DVK). Core libraries berfungsi untuk menerjemahkan bahasa *Java/C*. Sedangkan DVK merupakan sebuah virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.
5. *Linux Kernel*, merupakan layer yang berfungsi sebagai abstraction/ pemisah antara hardware dan software. *Linux kernel* inilah yang merupakan inti sistem operasi dari Android yang berfungsi untuk

mengatur sistem proses, memory, resource, dan driver. Linux kernel yang digunakan Android adalah linux kernel release 2.6.

fitur versi sebelumnya yaitu *Honeycomb* untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline*, dan berbagi informasi dengan NFC .

Beberapa keunggulan Platform Android adalah sebagai berikut (Safaat, 2001:3):

1. Lengkap (*Complete Platform*). Para desainer dapat melakukan pendekatan yang komprehensif ketika sedang mengembangkan platform Android. Android menyediakan banyak tools dalam membangun software dan merupakan sistem operasi yang aman.
2. Terbuka (*Open Source Platform*). Platform *Android* disediakan melalui lisensi open source.
3. Bebas (*Free Platform*). Android merupakan platform atau aplikasi yang bebas untuk dikembangkan. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada platform Android.

2.6 Pengujian Software

Pengujian software sangat diperlukan untuk memastikan software/aplikasi yang sudah/sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengembang atau penguji software harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan

dapat dideteksi sejak awal dan dikoreksi secepatnya. Pengujian atau testing sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan software seperti halnya analisis, desain, dan pengkodean [7].

2.6.1 White Box Testing

White box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Apabila modul yang telah dan sudah di hasilkan berupa output yang tidak sesuai dengan yang di harapkan maka akan dikompilasi ulang dan dilakukan pengecekan kembali kode-kode tersebut hingga sesuai dengan yang diharapkan.[7]

White box testing di sistem pakar penyakit anak ini akan menggunakan metode *Unit Testing Ekspresso* yang dimana terdapat *Tools* pada *Android Studio* yang secara mandiri akan mengecek setiap fungsi fungsi dari sistem.

2.6.2 Black Box Testing

Black box testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Black box testing bukanlah solusi alternatif dari white box testing tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh white box testing. [7]

Black box testing cenderung untuk menemukan hal-hal berikut [7] :

1. Fungsi yang tidak benar atau tidak ada.

2. Kesalahan antar muka (interface errors).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (performance errors).
5. Kesalahan inisialisasi dan terminasi.

2.7 SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. *SQLite* merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp.

Tidak seperti pada paradigma *client-server* umumnya, Inti *SQLite* bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan *API* secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi overhead, latency times, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

2.8 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu model jaringan yang disimpan dalam sistem secara abstrak dan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship* data.

Biasanya ERD ini digunakan oleh professional sistem untuk berkomunikasi dengan pemakaian eksekutif tingkat tinggi dalam suatu organisasi. ERD juga menguntungkan bagi profesional sistem karena ERD memperlihatkan hubungan antar data store pada DFD. Ada 3 (tiga) macam simbol yang digunakan dalam ERD yaitu[7] :

a. Entitas (*entity*)

Entitas merupakan individu yang mewakili sesuatu yang nyata eksistensinya dan dapat dibedakan dari sesuatu yang lain. Entitas dapat berupa objek, orang, konsep, abstrak atau kejadian.

b. Relasi (*relationship*)

Adalah hubungan atau asosiasi suatu entitas dengan dirinya sendiri atau dengan entitas lainnya. *Relationship* digambarkan sebagai garis yang menghubungkan entitas-entitas yang dipandang memiliki hubungan antara satu dengan yang lainnya.

c. Atribut (*attribute*)

Atribut mendeskripsikan karakteristik dari suatu entitas. Umumnya penetapan atribut bagi sebuah entitas didasarkan pada fakta yang ada.

Menurut Ladjamudin (2005:142), ada 3 (tiga) jenis relasi hubungan atribut dalam satu file, yaitu relasi satu ke satu, relasi satu ke banyak dan relasi banyak ke banyak [7].

a. Relasi Satu ke Satu

Hubungan antara *file* pertama dengan kedua adalah satu banding satu dengan relasi antar keduanya diwakilkan dengan tanda panah tunggal.

b. Relasi Satu ke Banyak

Hubungan antara *file* pertama dengan *file* kedua adalah satu banding banyak atau dapat pula dibalik, banyak banding satu dengan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

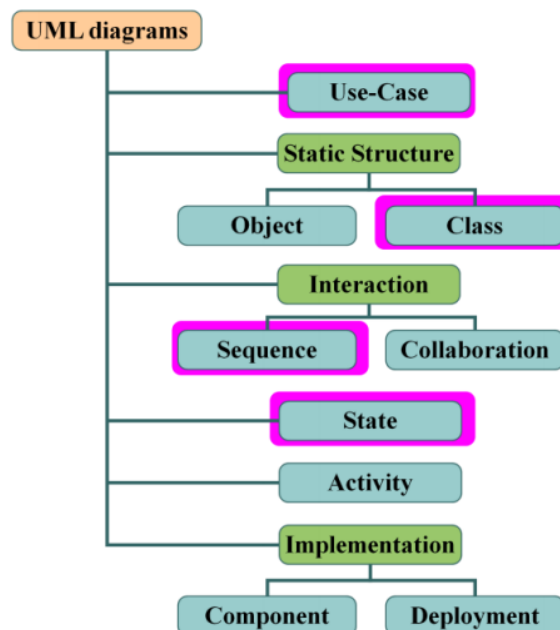
c. Relasi Banyak ke Banyak

Hubungan antara *file* pertama dengan *file* kedua adalah banyak banding banyak dengan relasi antar keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

Penentuan struktur *file* dilakukan dengan menentukan *file-file* apa saja yang harus disediakan dan yang akan digunakan dalam aktivitas kerja jika sistem diimplementasikan. Pemilihan *file* dapat ditentukan dari skema ER yang telah dalam bentuk tabel normal. Dari tabel tersebut baru akan dibuatkan struktur filenya. Struktur *file* berisi data yang harus disimpan oleh perusahaan serta cara data tersebut akan diakses.

2.8.1 Unified Modeling Language (UML)

Menurut Haviluddin (2011:1) *Unified Modeling Language* (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual juga merupakan satu umpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek [7]. Berikut gambar dari diagram UML.



Gambar 2.2. Gambar Diagram UML [7]

Tujuan dari penggunaan diagram menurut Schmuller J.(2004), “*The purpose of the diagram is to present multiple views of a system; this set of multiple views is called a model*” [7] yang memiliki arti dalam bahasa Indonesia “Tujuan dari diagram adalah untuk menampilkan atau menyajikan beberapa tampilan dari suatu sistem; beberapa tampilan ini disebut dengan model”.

Menurut Sugrue J.(2009) tujuan utama dari desain UML adalah [7] :

- a. Menyediakan analisis dan desain sistem suatu bahasa permodelan visual yang ekspresif sehingga pengguna dapat mengembangkan dan melakukan pertukaran model data yang bermakna;
- b. Sebagai penyedia mekanisme yang spesialisasi untuk memperluas konsep tersebut;
- c. Memberikan dasar formal untuk pemahaman dalam bahasa permodelan;

- d. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek;
- e. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.

2.8.2 *Use Case Diagram*

Use case adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan klien.

Diagram *use case* berguna dalam tiga hal, antara lain :

- a. Menjelaskan fasilitas yang ada (*requirements*)

Use case baru selalu menghasilkan fasilitas baru ketika sistem di analisa, dan design menjadilebih jelas.

- b. Komunikasi dengan klien

Penggunaan notasi dan simbol dalam diagram *use case* membuat pengembang lebih mudah berkomunikasi dengan klien-kliennya.

- c. Membuat test dari kasus-kasus secara umum

Kumpulan dari kejadian-kejadian untuk *use case* bisa dilakukan test kasus layak untuk kejadian-kejadian tersebut.

Menurut Menurut Haviluddin (2011:6) komponen-komponen pembentuk *Use Case* terdiri dari aktor dan usecase [21].

a. *Actor*

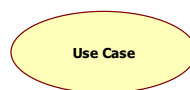
Actor merupakan penentu peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* merupakan segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer yang bertugas memberikan informasi kepada sistem serta dapat memberikan perintah kepada sistem untuk melakukan suatu tugas.



Gambar 2.3. *Use case actor*

b. *Use Case*

Use Case merupakan deskripsi fungsi dari sebuah sistem yang bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistem itu sendiri melalui sebuah cerita mengenai bagaimana sistem tersebut digunakan.



Gambar 2.3. *Use case*

2.8.3 *Sequence Diagram*

Diagram *Class* dan diagram *object* merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti diagram *interaction*. Diagram *sequence* merupakan salah satu diagram *interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan. Diagram ini diatur berdasarkan waktu. Objek-objek

yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Menurut Nugroho (2009:38), tujuan penggunaan *sequence* diagram adalah sebagai berikut [7] :

- a. Mengkomunikasikan *requirement* kepada tim teknis karena diagram ini dapat lebih mudah untuk menjadi model desain serta mudah digunakan dalam melakukan permodelan suatu sistem.
- b. Merupakan diagram yang paling cocok untuk mengembangkan model deskripsi *use case* menjadi spesifikasi desain.
- c. Analisa dan desain, memfokuskan pada identifikasi method di dalam sebuah sistem. *Sequence* diagram biasanya dipakai untuk memodelkan :
 1. Deskripsi tentang sistem yang ada pada sebuah/beberapa *use case* pada *use case* diagram, yang menggambarkan hubungan antara *actor* dan *use case* diagram;
 2. Logika dari *method* (*operation*, *function* atau *procedure*);
 3. Logika dari *service* (*high level method*).

Komponen-komponen yang digunakan dalam melakukan permodelan menggunakan *sequence* diagram adalah sebagai berikut :

- a. *Actor*

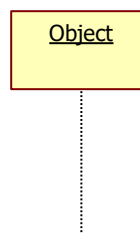
Actor merupakan segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer yang bertugas memberikan informasi kepada sistem serta dapat memberikan perintah kepada sistem untuk melakukan suatu tugas.



Gambar 2.4. *Actor* pada *sequence* diagram

b. *Object*

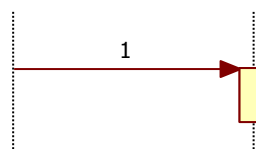
Object Merupakan komponen dalam melakukan permodelan menggunakan *sequence* diagram dimana komponen ini digunakan sebagai objek-objek yang akan digunakan didalam pembuatan diagram.



Gambar 2.5. *Object* pada *sequence* diagram

c. *Stimulus*

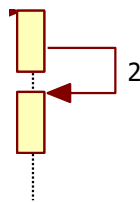
Stimulus Merupakan komponen dalam melakukan permodelan menggunakan *sequence* diagram dimana komponen ini digunakan sebagai penanda adanya hubungan yang terjadi antara dua buah *object* maupun *actor* dimana terjadinya komunikasi berupa pengiriman pesan atau *message*.



Gambar 2.6. *Stimulus*

d. *Self Stimulus*

Self stimulus Merupakan komponen dalam melakukan permodelan menggunakan *sequence* diagram dimana komponen ini memiliki fungsi yang sama dengan *stimulus* yaitu digunakan sebagai penanda adanya hubungan yang terjadi antara dua buah *object* maupun *actor*. Namun, komunikasi berupa pengiriman pesan atau *message* dilakukan kepada dirinya sendiri, bukan pada objek lainnya.



Gambar 2.7. *Self stimulus*

2.8.4 Activity Diagram

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *statechart*. Diagram *statechart* berfokus pada obyek dalam suatu proses (atau proses menjadi suatu objek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.

Komponen-komponen yang digunakan dalam melakukan permodelan menggunakan *activity* diagram adalah sebagai berikut:

a. *Activity*

Activity merupakan komponen yang digunakan dalam pembuatan *activity* diagram yang digunakan untuk memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.



Gambar 2.8. *Activity* pada *activity* diagram

b. *Action*

Action merupakan komponen yang berisikan suatu *statement* dari suatu sistem yang mencerminkan eksekusi dari suatu aksi.



Gambar 2.9. *Action* pada *activity* diagram

c. *Start State*

Start state merupakan komponen yang digunakan dalam pembuatan *activity* diagram yang digunakan untuk mendeskripsikan bagaimana suatu objek dibentuk dan diawali.



Gambar 2.10. *Start state* pada *activity* diagram

d. *End State*

End state merupakan komponen yang digunakan dalam pembuatan *activity* diagram yang digunakan untuk mendeskripsikan bagaimana suatu objek dibentuk dan dihancurkan.



Gambar 2.11. *End state* pada *activity* diagram

e. *State Transition*

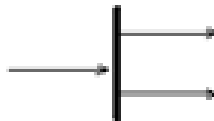
State transition merupakan komponen yang digunakan dalam pembuatan *activity* diagram yang digunakan untuk memperlihatkan atau menunjukkan kegiatan selanjutnya setelah suatu kegiatan telah selesai dilakukan.



Gambar 2.12. *State transition* pada *activity* diagram

f. *Fork*

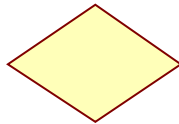
Fork merupakan komponen yang digunakan dalam pembuatan *activity* diagram yang digunakan untuk menunjukkan aliran percabangan pada permodelan ini.



Gambar 2.13. *Fork* pada *activity* diagram

g. *Decision*

Decision merupakan komponen yang digunakan dalam pembuatan *activity* diagram yang digunakan untuk menunjukkan pilihan dalam mengambil sebuah keputusan.



Gambar 2.14. *Decision* pada *activity* diagram

h. *Flow Final*

Flow final merupakan komponen yang digunakan dalam pembuatan *activity* diagram yang digunakan untuk menunjukkan aliran akhir.



Gambar 2.15. *Flow Final* pada *activity* diagram

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa “*Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek / *Object Oriented programming (OOP)*”.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Garis Besar Sistem

Secara garis besar Sistem Pakar Penyakit anak yang dirancang dengan menggunakan metode *Best First Search* ini bertujuan untuk mempermudah pekerjaan Dokter/petugas kesehatan yang kesulitan dalam mendiagnosis penyakit anak . Data yang digunakan di sistem pakar ini didapat dari pakar langsung yaitu ibu Siti Fitriah seorang petugas kesehatan yang sudah mengabdikan selama 20 tahun dalam penyembuhan penyakit anak dan ibu . Sistem ini berjalan di *Smartphone* karena saat ini pengguna *Smartphone* lebih banyak daripada pengguna *Personal Computer*. Sistem Pakar ini hanya berisi penyakit penyakit yang umum diderita anak dan perlu penanganan yang tak sulit.

3.2 Analisis Sistem

Analisis sistem (*system analysis*) merupakan tahap penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat memberikan solusi atas permasalahan yang ada.

3.3 Analisis Kebutuhan

Analisis kebutuhan sistem merupakan tahap analisis yang digunakan untuk mengetahui kebutuhan yang diperlukan dalam merealisasikan sistem yang akan dibuat. Hal ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan informasi, dan kebutuhan pengguna.

3.3.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dapat dilakukan oleh sistem. Berikut adalah kebutuhan fungsional dari sistem informasi pencarian lowongan kerja dengan menggunakan algoritma *Best First Search*:

1. Kebutuhan fungsional User/Dokter
 - Sistem dapat menampilkan info tentang penyakit anak
 - Sistem dapat memberikan diagnosa penyakit anak
 - Sistem dapat memberikan hasil dari diagnosa yang telah dilakukan
 - Sistem dapat memberikan persentase dari hasil diagnosa

3.3.2 Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah tipe kebutuhan mengenai faktor-faktor pendukung sistem agar sistem dapat berjalan secara optimal.

3.3.2.1 Kebutuhan Perangkat Keras (Hardware)

Kebutuhan perangkat keras (hardware) merupakan analisis kebutuhan sistem yang digunakan untuk mengetahui secara jelas perangkat yang dibutuhkan untuk mendukung proses pengembangan dan penggunaan dari sistem aplikasi yang akan dibuat. Adapun spesifikasinya sebagai berikut :

1. Kebutuhan perangkat keras yang digunakan untuk pengembangan sistem
 - a. Processor Intel Core I7 2,8 GHz
 - b. RAM 8 GB DDR3
 - c. Penyimpanan HDD 1 TB SATA
 - d. VGA NVIDIA GTX950M 4 GB

2. Kebutuhan perangkat keras untuk pengguna
 - a. CPU Quad-core 1.4 GHz Cortex-A53 .
 - b. Minimal memory 1 GB RAM.
 - c. Sistem Operasi Jellybean

3.3.2.2 Kebutuhan Perangkat Lunak (Software)

Kebutuhan perangkat lunak (*software*) yaitu program yang diperlukan untuk melakukan proses intruksi atau menjalankan perangkat keras. Agar sistem dapat dibuat dan diimplementasikan sesuai perancangan, maka diperlukan suatu perangkat lunak. Software yang digunakan dalam pembuatan sistem adalah sebagai berikut:

1. Kebutuhan perangkat lunak yang digunakan untuk pengembangan sistem
 - Android Studio Versi 3.1
 - Gradle 4.6
 - SQLite

3.3.3 Analisis Kebutuhan Pengguna

Kebutuhan pengguna (*user*) ditekankan pada pengguna sistem yaitu Dokter ataupun petugas kesehatan yang punya izin melakukan diagnosa penyakit. Sistem Pakar penyakit anak Memberikan sebuah gejala gejala yang bisa dibandingkan dengan kondisi anak, gejala yang ditampilkan oleh sistem adalah gejala umum bukan gejala yang harus pemeriksaan lebih lanjut untuk diketahui, hasil dari diagnosa akan ditampilkan jika gejala yang ada sudah dipilih, hasil dari gejala tersebut akan diberikan dalam sebuah, dengan menggunakan Metode *Best First Search* maka pengguna tak perlu terlalu lama dalam memilah diagnosa yang ada.

3.3.4 Analisis Hasil Diagnosa

Untuk memberikan hasil dan gejala yang mudah dipahami penulis langsung membahas gejala-gejala yang ada dengan sang pakar untuk memastikan hasil dari diagnosa ini bisa memberikan hasil yang memuaskan

3.3.5 Analisis Informasi Pada Sistem

1. Sistem akan menampilkan pilihan saat memilih sebuah gejala, jika tak memilih maka ada pesan pemberitahuan dimana user/pengguna harus memilih
2. Sistem akan memberikan informasi berupa penyakit, deskripsi penyakit dan penanganan dari penyakit yang ada

3.3.6 Analisis Kinerja Sistem

1. Sistem dapat melakukan diagnosa penyakit anak.
2. Sistem dapat memberikan hasil sebuah penyakit dari diagnosa yang telah dilakukan.
3. Sistem dapat memberikan penanganan dari hasil diagnose yang telah dilakukan.

3.3.7 Analisis Kelemahan Sistem

Analisis Kelemahan Sistem (*menurut KONSEP PIECES*)

PERFORMANCE (ANALISIS KINERJA)

Sistem Baru : user dapat melakukan diagnosa penyakit anak langsung dari *Smartphone*

Sistem Lama : User harus membukan buku catatan untuk mengetahui penyakit yang diderita anak

INFORMATION (Analisis Informasi)

Sistem Baru : Pengguna leluasa dalam mendalami dan mendiagnosa penyakit anak yang bersangkutan, jadi tak perlu mencari di buku buku maupun mengatakan bukan spesialisnya

Sistem Lama : Pengguna harus menolak jika ada Gejala yang tidak diketahui, karena tak memiliki Ilmu yang cukup untuk mendiagnosa penyakitnya

ECONOMY (Analisis Ekonomi)

Sistem Baru : Pengguna tak perlu membeli banyak buku gejala dan diagnosa penyakit, bisa menekan biaya membeli Buku buku tersebut, dan diganti dengan Aplikasi yang khusus untuk mendiagnosa Penyakit yang bersangkutan

Sistem Lama : Pengguna harus membeli buku buku dan belajar kepada para Dokter Spesialis untuk mengetahui Gejala gejala tentang penyakit anak

Control (Analisis Keamanan)

Sistem Baru : hasil dari diagnosa lebih pasti karena dari Pakar Langsung

Sistem Lama : Jika dalam mendiagnosa dengan Dokter langsung bisa terjadi kesalahan yang mungkin terjadi sehingga hasil Diagnosa dan kenyataan berbeda

EFICIENCY (Analisis Efisiensi)

Sistem Baru : Pengguna lebih cepat dan hemat dalam mendiagnosa penyakit anak, tak perlu berlama lama ini juga akan mempercepat pasien menerima informasi yang diperlukan

Sistem Lama : Pengguna harus menulis satu persatu gejala yang ada lalu menyimpulkan yang tidak sebentar semakin banyak gejala maka semakin lama Kesimpulannya

SERVICE

Sistem baru : dengan Aplikasi Pengguna lebih dimudahkan untuk mendiagnosa penyakit dan waktu yang digunakan relative lebih cepat dengan hasil yang lebih pasti

Sistem Lama : Pengguna harus berfikir dengan tenkonsetrasi lebih demi mendapatkan Diagnosa kesimpulan yang tepat

3.4 Analisis Kelayakan Sistem

Visi : Membangun Sistem yang mempermudah Mendiagnosis Penyakit

Anak yang Memiliki Hasil yang Tepat

Misi :

- Membangun Sistem dengan data penyakit yang banyak
- Mencari pakar yang benar benar ahli di bidang penyakit anak
- Terus mengembangkan sistem

3.4.1 Kelayakan Teknis

Hardware : Tak memerlukan Hardware yang besar karena ini hanya

Aplikasi Ringan yang

Dimana HP smartphone sudah dirancang dengan spesifikasi umumnya lebih dari cukup

Software : Software yang dipakai adalah Android yang versinya mampu menjalankan Aplikasi ini

3.4.2 Kelayakan Hukum

Sistem ini dirancang dengan bantuan Data dari Ahlinya yang sudah berkecimpung puluhan tahun dibidang kesehatan anak maupun penyakit anak, serta pakar yang ada sudah memiliki perizinan praktek dari pemerintah secara resmi

3.4.3 Kelayakan Operasional

Dalam menjalankan Sistem Ini Penggunaannya mudah Sudah disediakan Langkah langkah, apalagi di zaman dimana Aplikasi *Android* memang mudah digunakan banyak orang dari berbagai umur dan kalangan

3.4.5 Kelayakan Ekonomi

Penggunaan Sistem ini mengurangi biaya ke Dokter, Mencari Spesialis Penyakit anak maupun biaya Konsultasi Penyakit yang diderita Anak

3.5 Metode Best First Search

Berikut adalah rancangan algoritma atau langkah-langkah kerja layanan sistem pakar penyakit anak menggunakan Metode *Best First Search*.

Langkah pertama adalah dengan mengembangkan gejala gejala dan membahasnya bersama pakar lalu menanyakan penyakit apa saja yang umum menyerang anak umur 5 sampai 10 tahun, dan gejala gejala umum apa saja yang mudah di pahami

Langkah kedua adalah didapatkan data-data sebagai berikut :

Tabel 3.1 gejala

ID	Gejala
G001	Berkeringat pada malam hari
G002	Tubuh terasa lemas
G003	nafsu makan berkurang
G004	bersin bersin
G005	hidung tersumbat
G006	tenggorokan kering
G007	hidung berlendir
G008	sering BAB
G009	BAB nya encer / mencret
G010	terlalu banyak makan
G011	makan makanan yang baru
G012	mual mual
G013	Pusing
G014	Sakit perut
G015	mengigil
G016	Tubuh Panas
G017	muncul bintik merah
G018	gatal gatal
G019	sering terdengar suara uhuk uhuk dari anak

Tabel 3.2 Penyakit

KODE	PENYAKIT	DESKRIPSI	PENANGANAN
P01	Demam	Demam adalah kondisi ketika suhu tubuh berada di atas angka 38 derajat celsius. Demam merupakan bagian dari proses kekebalan tubuh yang sedang melawan infeksi akibat virus, bakteri, atau parasit. Selain itu, demam juga bisa terjadi pada kondisi hipertiroidisme, artritis, atau karena penggunaan beberapa jenis obat-obatan, termasuk antibiotik. Kenaikan suhu tubuh akibat konsumsi obat ini disebut dengan demam obat atau “drug fever”	Istirahat. Anda butuh cukup istirahat untuk memulihkan diri dan membantu kekebalan tubuh. Terlalu banyak beraktivitas dapat meningkatkan suhu tubuh
P02	Pilek	Pilek adalah infeksi ringan pada hidung, saluran sinus, tenggorokan, dan saluran pernapasan bagian atas akibat serangan virus. Pilek bisa dialami oleh siapa saja, mulai dari anak-	Mengonsumsi suplemen zinc dan vitamin C. Kedua cara ini dipercaya dapat menurunkan tingkat keparahan gejala dan mempercepat penyembuhan pilek apabila langsung

		<p>anak hingga dewasa', 'Meneteskan larutan garam ke dalam hidung. Cara ini dapat meredakan gejala hidung tersumbat serta melancarkan pernapasan, terutama pada bayi dan anak-anak</p>	<p>diterapkan ketika gejala pilek baru muncul.</p>
P03	Diare	<p>Diare adalah penyakit yang membuat penderitanya menjadi sering buang air besar, dengan kondisi tinja yang encer. Pada umumnya, diare terjadi akibat akibat makanan dan minuman yang terpapar virus, bakteri, atau parasit.', 'Meningkatkan konsumsi cairan.</p>	<p>Mengganti kehilangan cairan dan elektrolit adalah salah satu kunci penting dalam penanganan diare. Hal ini juga diperlukan untuk mencegah terjadinya dehidrasi. Caranya adalah dengan mengonsumsi cairan sebanyak-banyaknya, bisa berupa air putih, jus, atau kaldu. Pada anak-anak, pemberian oralit sangat disarankan. Pada bayi yang masih menyusui, asupan ASI harus selalu terjaga Mengonsumsi makanan yang</p>

			<p>tepat</p> <p>Saat mengalami diare, penderita dianjurkan untuk mengonsumsi makanan lunak selama beberapa hari. Selain itu, hindari juga makanan yang sarat lemak, serat, atau bumbu. Jika kondisi usus sudah membaik, ganti ke makanan semi padat dengan kadar serat yang ditingkatkan secara bertahap.</p>
P04	<p>Muntah</p> <p>Muntah</p>	<p>adalah suatu refleks yang tidak dapat dikontrol untuk mengeluarkan isi lambung dengan paksa melalui mulut. Gejala yang sering terjadi bersama dengan muntah yaitu mual. Pada beberapa kasus, muntah akan berhenti jika isi racun sudah keluar.',</p>	<p>jika Anda muntah berkali-kali, konsumsi cairan saja. Awali dengan makan lunak jika Anda mampu. Lalu perlahan kembali ke makanan biasa</p> <p>konsumsi obat-obatan sesuai anjuran dokter dan apoteker</p> <p>jika Anda mengalami motion sickness atau mabuk kendaraan, menyandarlah di tempat duduk dan tarik napas</p>

			dalam perlahan saat Anda di dalam mobil, kapal atau pesawat. Duduklah pada tempat yang Bergeraknya minimal
P05	Cacar Air	Cacar air adalah infeksi virus pada kulit dan membran mukosa, yang menyebabkan lenting pada seluruh tubuh dan wajah. Penyakit cacar air, atau biasa disebut chickenpox (varisela) ini, dapat menular ke orang yang belum pernah menderita cacar air atau belum pernah menerima vaksin cacar air	Cacar air tidak memiliki langkah penanganan khusus. Tujuan pengobatannya adalah untuk mengurangi gejala. Obat-obatan yang digunakan untuk menangani cacar air biasanya ada dua jenis. Pertama adalah parasetamol untuk menurunkan demam. Yang kedua adalah losion atau bedak kalamine untuk mengurangi rasa gatal pada kulit.

P06	Batuk Batuk	<p>Batuk adalah respons alami dari tubuh sebagai sistem pertahanan saluran napas jika terdapat gangguan dari luar. Respons ini berfungsi membersihkan lendir atau faktor penyebab iritasi atau bahan iritan (seperti debu atau asap) agar keluar dari paru-paru dan saluran pernapasan bagian atas.</p>	<p>Batuk ringan jarang membutuhkan langkah pengobatan yang serius, cukup diatasi dengan obat batuk untuk meredakan rasa gatal, atau untuk mengencerkan dahak agar lebih mudah dikeluarkan.</p> <p>Namun jika batuk masih berkepanjangan, itu bisa menjadi pertanda adanya infeksi saluran pernapasan yang lebih serius. Konsultasikanlah ke dokter jika Anda mengalami batuk yang parah dan tidak kunjung reda, sehingga pengobatan dapat segera dilakukan.</p>
-----	-------------	---	---

Dari Hasil diatas didapatkan basis pengetahuan Sebagai Berikut :

Tabel 3.3 Basis Pengetahuan

NO	Penyakit	Gejala	Bobot	Total
1	Demam	Berkeringat pada malam hari	25%	100%
		Tubuh terasa lemas	25%	
		nafsu makan berkurang	25%	
		tubuh panas	25%	
2	Pilek	Berkeringat pada malam hari	20%	100%
		Tubuh terasa lemas	20%	
		Bersin bersin	20%	
		hidung tersumbat	20%	
		hidung berlendir	20%	
3	Diare	Pusing	33.33%	100%
		sering BAB	33.33%	
		BAB nya encer/mencret	33.33%	
4	Muntah Muntah	terlalu banyak makan	33.33%	100%
		makan makanan yang baru	33.33%	
		mual mual	33.33%	
5	cacar air	pusing	20%	100%
		sakit perut	20%	
		mengigil	20%	

		muncuk bintik merah	20%	
		gatal gatal	20%	
6	batuk batuk	Berkeringat pada malam hari	33.33%	100%
		tenggorokan kering	33.33%	
		sering mengeluarkan suara uhuk uhuk	33.33%	

3.5.1 Menyusun Motor Inferensi

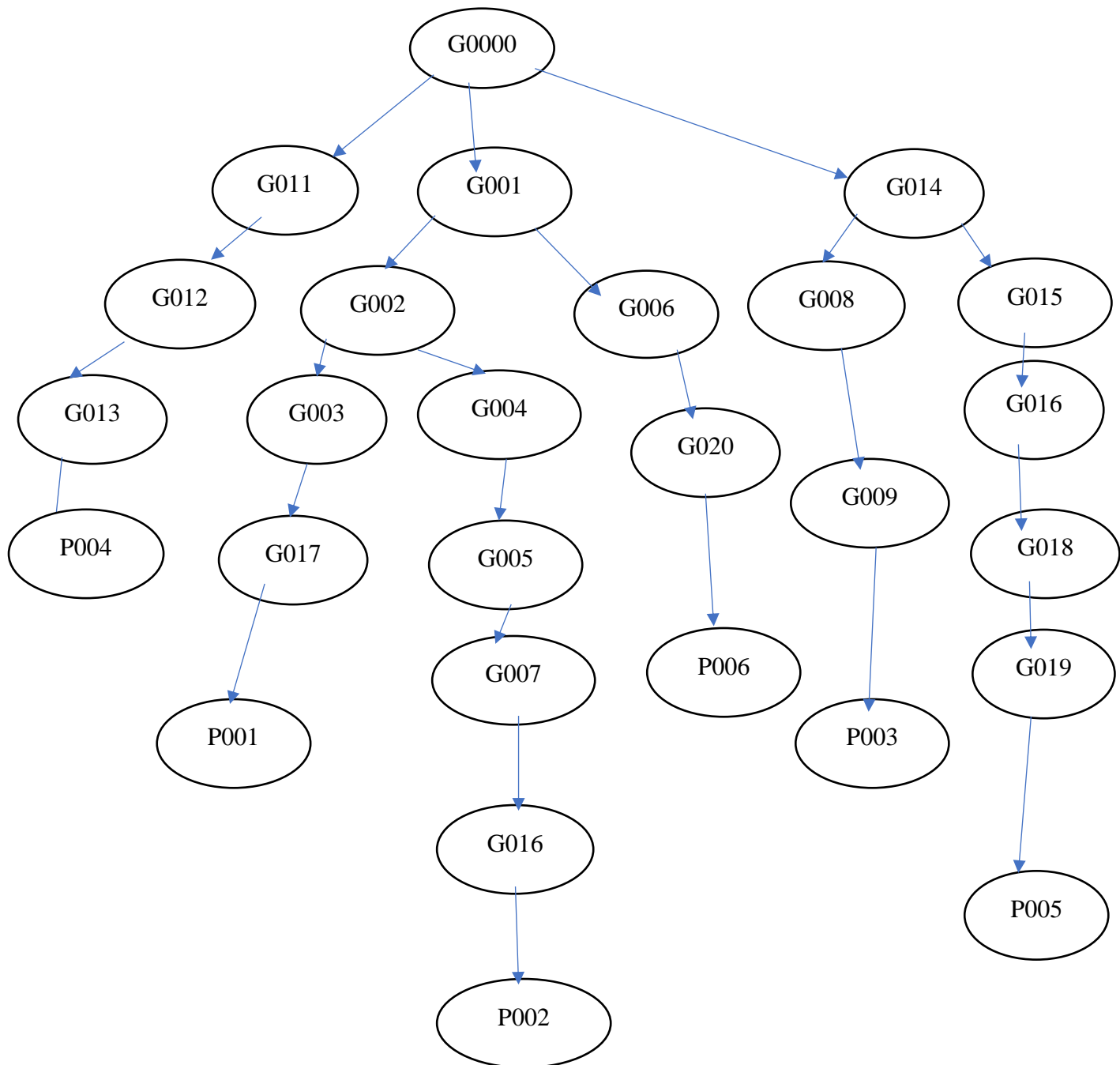
Penyusunan motor inferensi pada Sistem Pakar Diagnosa Penyakit anak metode *Best First Search* yaitu penelusuran fakta atau pernyataan dimulai dari bagian sebelah

kiri (*IF-Then*) dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis yang ada dalam basis pengetahuan dan menentukan nilai kemungkinan gejala yang sama dari penyakit yang berbeda berdasarkan bobot nilai yang telah ditentukan. Gejala yang sama merupakan gejala utama dari tiap penyakitnya, setiap gejala yang sama tersebut akan mempunyai nilai[6].

3.5.2 Pohon Keputusan

Data yang didapatkan akan menjadi rujukan dalam membuat *Descision Tree* atau Pohon Keputusan, pohon keputusan akan dibuat berdasarkan metode *Best First Search* yang dimana Metode ini akan menggunakan nilai *Heuristic* untuk teknik penelusurannya, dalam *Best First Search*, dibagi jadi beberapa Level Node dari Level 0 hingga level dimana Lv terendah adalah gejala yang paling umum ditemui dan gejala yang paling banyak di beberapa penyakit, karena metode *Best First*

Searcg menggabungkan Metode *Breadth First Search* dan *Depth First Search* maka dalam Pohon keputusan akan memasukan 2 metode tersebut , sehingga akan didapatkan Pohon keputusan sebagai berikut :

Gambar 3.1 Pohon Keputusan

G000 adalah saat dimana diagnosa belum dimulai disini disebut Level 0 , jika diagnosa dimulai maka akan masuk ke untuk menanyakan penyakit G011 di level 1 jika benar maka akan dilanjutkan ke G012 di level 2 jika menjawab salah maka pertanyaan yang muncul adalah node dengan level yang sama yaitu G001 dan seterusnya hingga ditemukan ujung dari Node level tersebut

3.6 Proses Diagnosa

Dari data-data masukan yang diperoleh sebelumnya, proses diagnosa Penyakit anak akan dilakukan setelah sistem menerima jawaban yang dimasukkan oleh pengguna dari pertanyaan yang diberikan sistem. Langkah-langkah yang terjadi dalam sistem adalah sebagai berikut:

- | | |
|-------------|--|
| Langkah I | user masuk ke menu sistem pakar penyakit anak, lalu memilih menu diagnosa penyakit, |
| Langkah II | user akan diberikan gejala awal yaitu gejala jika di pohon keputusan dimunculkan gejala Level 1 jika pengguna menekan “YA” maka akan lanjut ke node level 2 jika “Tidak ” maka akan masuk menuju ke Gejala Level 1 lain, jika user terus menekan tidak maka akan muncul pesan “ <i>Anak ada tidak terkena penyakit atau sistem ini tak mengenali penyakit tersebut</i> ” |
| Langkah III | Ketika telah mencapai titik terendah, maka akan muncul hasil diagnosanya, dengan mencocokkan dengan pohon keputusan, lalu sistem akan memilih penyakit mana yang |

punya nilai Bobot tertinggi, penyakit dengan nilai bobot tertinggi akan menjadi hasil dari diagnose

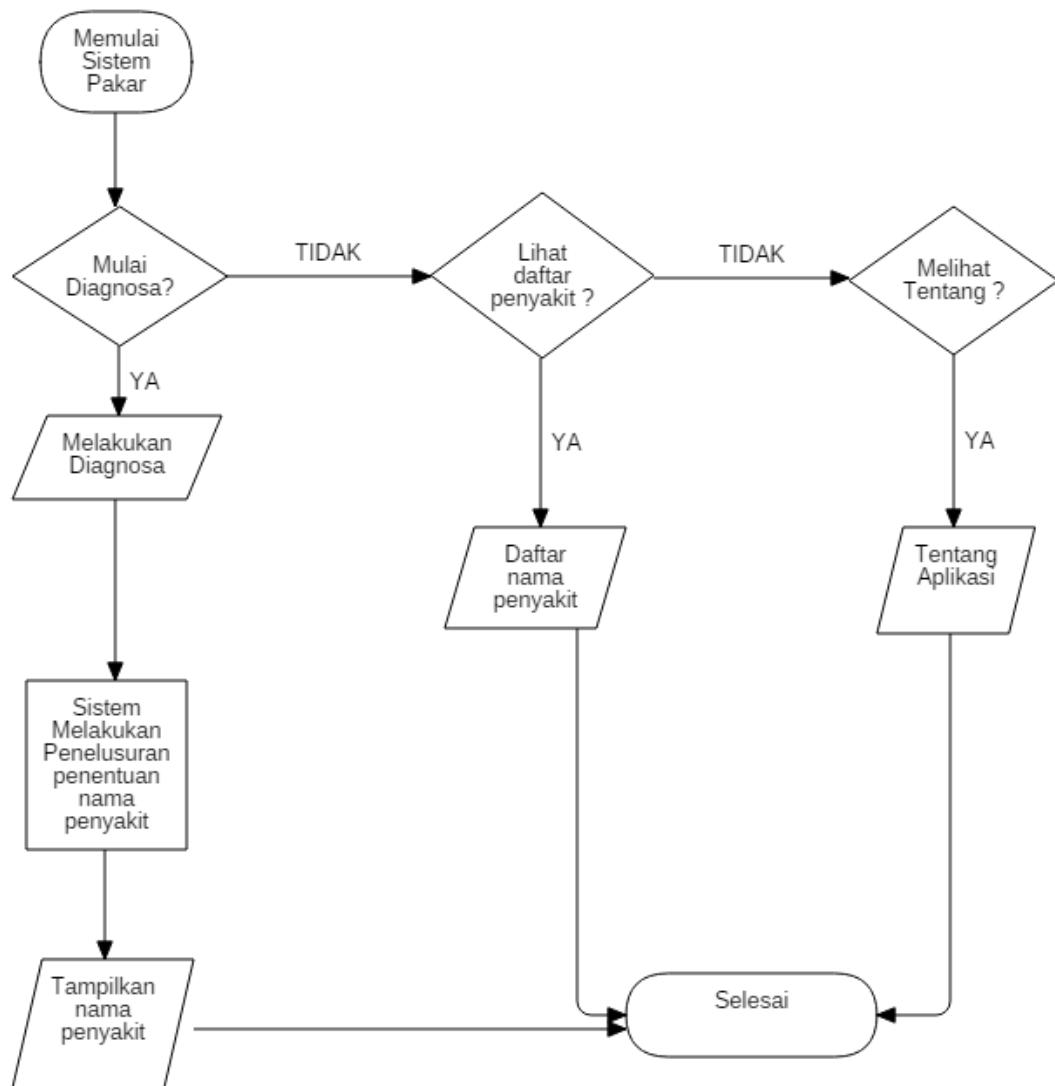
3.6.1 Hasil keluaran/*Ouput*

Tujuan akhir dari pembuatan sistem ini adalah menghasilkan informasi berupa jenis penyakit yang didagnosa penyakit anak untuk tiap diagnosa penyakit yang diperoleh[6].

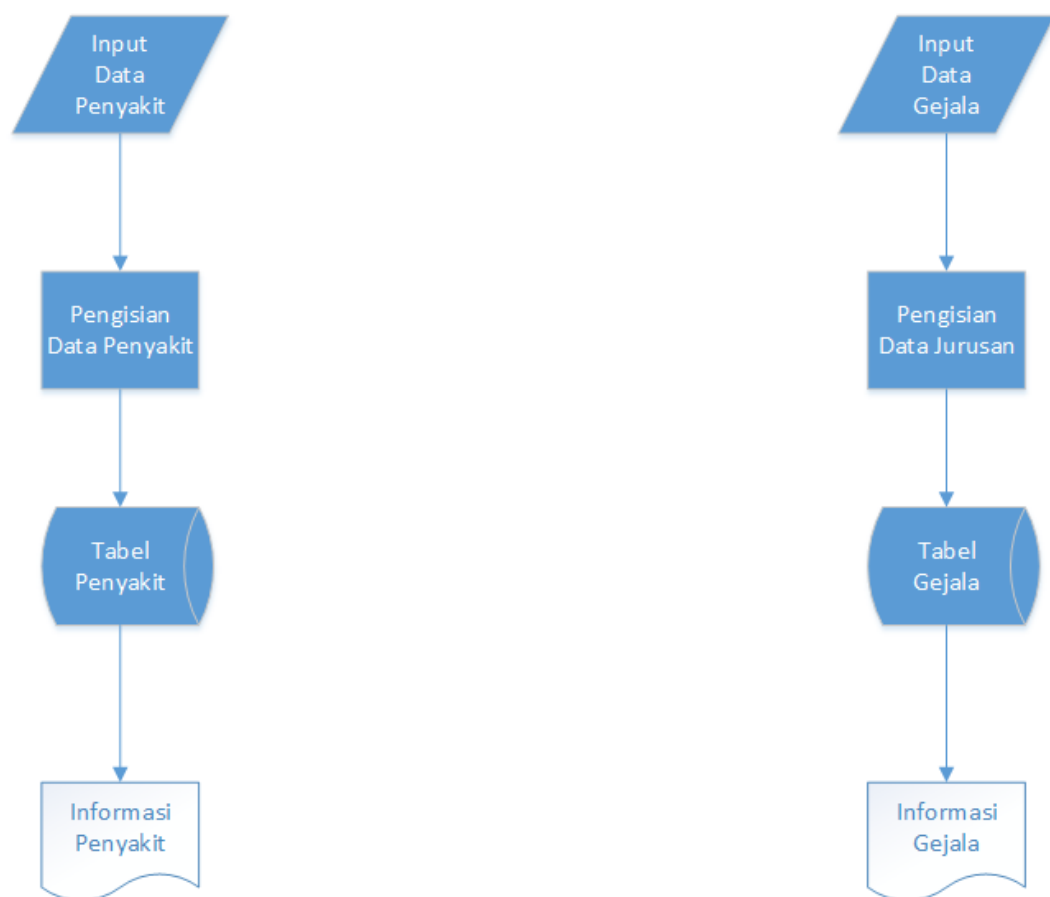
Selanjutnya untuk mempermudah perancangan sistem akan dibuat Perancangan Tabel, Bagan Alir (*Flowchart*) dan juga *Unified Modeling Language* (UML).

3.7 Perancangan Sistem

3.7.1 Flowchart



Gambar 3.2 Flowchart

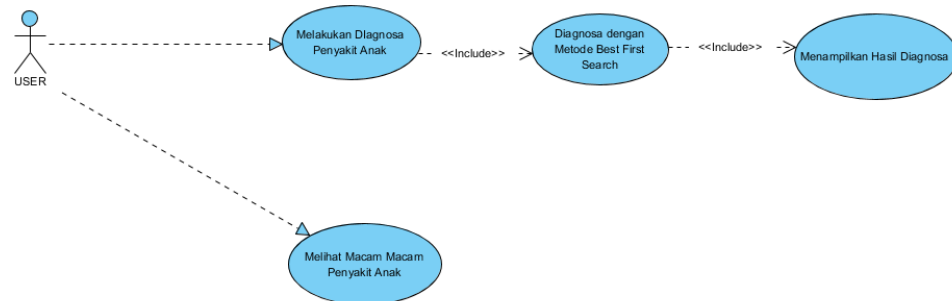


Gambar 3.3 Flowchart System

3.7.2 Unified Modeling Language (UML)

3.7.2.1 Use case

Use Case merupakan deskripsi lengkap mengenai suatu sistem ataupun perangkat lunak yang akan dibangun dan berperilaku untuk para aktor-aktor yang terlibat dalam penggunaan aplikasi yang akan dibangun. *Use Case* diagram merupakan alur yang mendeskripsikan mengenai interaksi yang terjadi antara aktor dengan sistem yang sedang dikembangkan.



Gambar 3.4 Use Case Diagram

1. Definisi Aktor

Definisi aktor bertujuan untuk mendeskripsikan atau menjelaskan peranan aktor dalam sistem. Seperti gambar 3.4

Tabel 3.4 Definisi Aktor

No	Nama Aktor	Deskripsi
1	User	Pengguna yang melakukan Diagnosa Penyakit anak dan bisa melihat informasi macam macam penyakit anak

2. Deskripsi Use Case

Deskripsi *Use Case* bertujuan untuk mendeskripsikan atau menjelaskan secara rinci suatu *Use Case* dalam suatu sistem. Berikut merupakan deskripsi *Use Case* Sistem Pakar Penyakit anak, tabel 3.5

Tabel 3.5 deskripsi Use Case Diagnosa

Nama Use case	Diagnosa
Aktor	User
Tujuan	Pengguna dapat mengakses menu Diagnosa dan macam macam Penyakit
Pra kondisi	Masuk ke Bagian Home
Trigger	Pengguna Menekan Tombol Mulai Diagnosa
Alur Utama	<ol style="list-style-type: none"> 1. Pengguna Memilih Gejala sesuai kondisi Anak 2. Menekan YA atau Tidak sesuai Kondisi Anak 3. Sistem Menampilkan Hasil Diagnosa berupa nama penyakit
Kondisi Sukses	Sistem Memberikan Hasil Diagnosa berupa nama penyakit
Kondisi Gagal	Sistem Menampilkan Pesan “Anak anda tak terkena penyakit atau sistem kami tak mengenali penyakit anak anda”

Tabel 3.6 deskripsi Use Case Macam Macam Penyakit

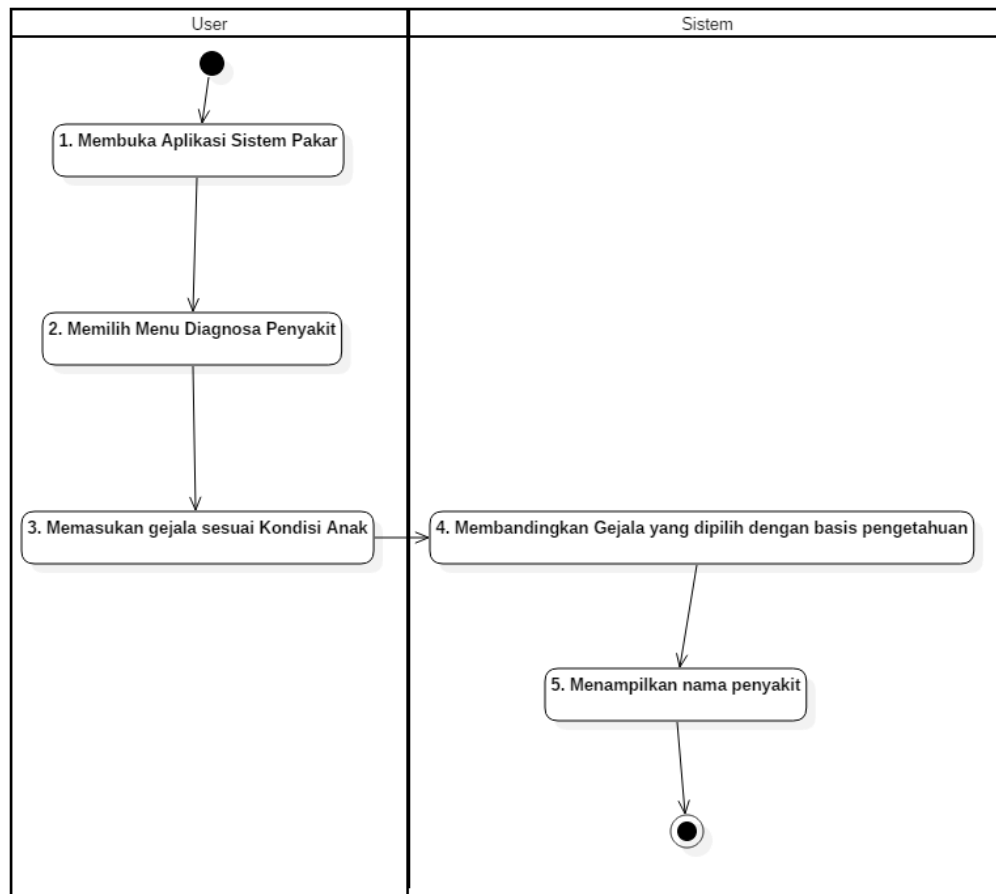
Nama Use case	Diagnosa
Aktor	User
Tujuan	Pengguna dapat mengakses menu Diagnosa dan macam macam Penyakit
Pra kondisi	Masuk ke Bagian Home
Trigger	Pengguna Menekan Tombol Macam Macam Penyakit

Alur Utama	<ol style="list-style-type: none"> 1. Pengguna menekan tombol macam macam penyakit 2. Pengguna melihat daftar penyakit 3. Pengguna mencari nama penyakit lewat pencarian 4. Sistem Menampilkan Nama-nama Penyakit
Kondisi Sukses	Sistem memberikan nama penyakit, deskripsi penyakit dan solusi dari penyakit
Kondisi Gagal	Sistem tak menemukan penyakit apa yang dicari pengguna

3.7.2.2 Activity Diagram

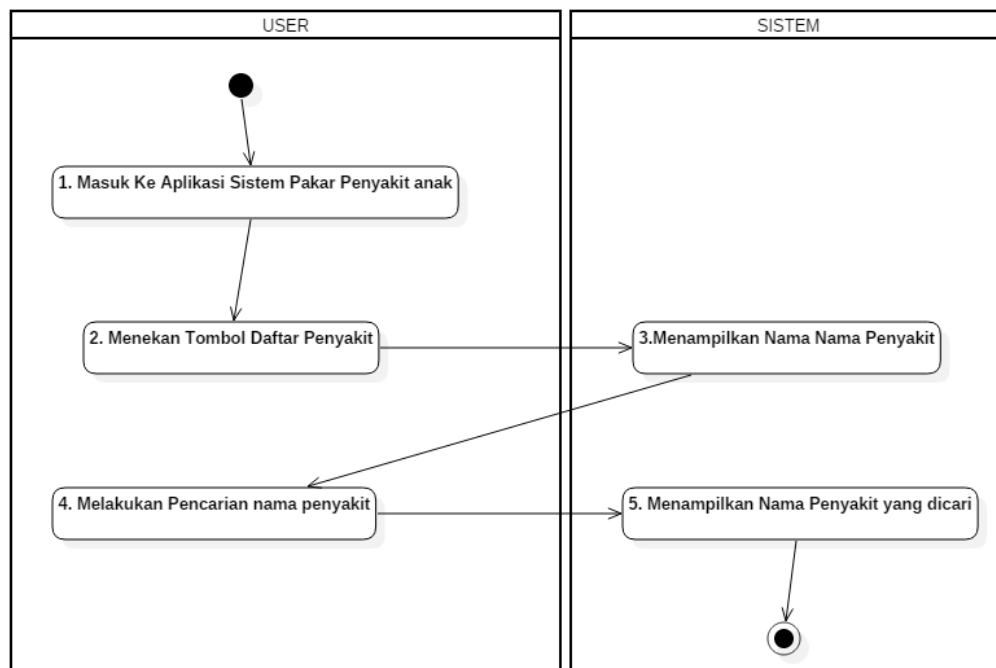
Activity diagram bertujuan untuk menunjukkan alur atau aktifitas yang dilakukan user terhadap sistem secara prosedural. Adapun activity diagram yang digunakan dalam sistem informasi pencarian lowongan kerja ini, ditunjukkan dengan gambar berikut[7]:

Activity diagram diagnosa merupakan diagram yang menjelaskan alur diagnosa yang ada pada sistem mulai dari gejala pertama hingga muncul penyakitnya oleh sistem. Gambar 3.5



Gambar 3.5 Activity Diagram Diagnosa

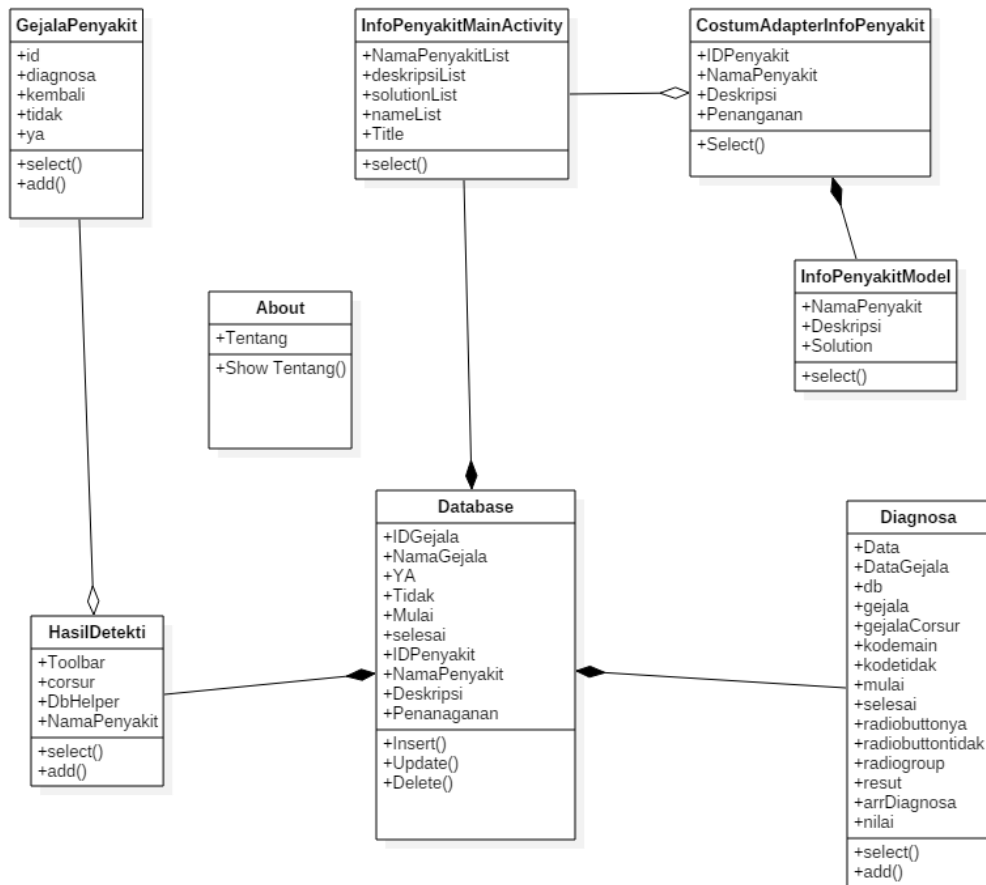
Activity diagram Macam Macam Penyakit merupakan diagram yang menjelaskan alur melihat macam macam penyakit yang ada pada sistem. Seperti gambar 3.6



Gambar 3.6 Activity Diagram Macam Macam Penyakit

3.7.2.3 Class Diagram

Class diagram mendeskripsikan struktur statis dari kelas-kelas dalam sistem dan mengilustrasikan atribut, operasi dan relasi antara satu kelas dengan kelas yang lain[7]. Pada Sistem Pakar Penyakit Anak ini, class diagram tersebut ditunjukkan oleh gambar berikut:

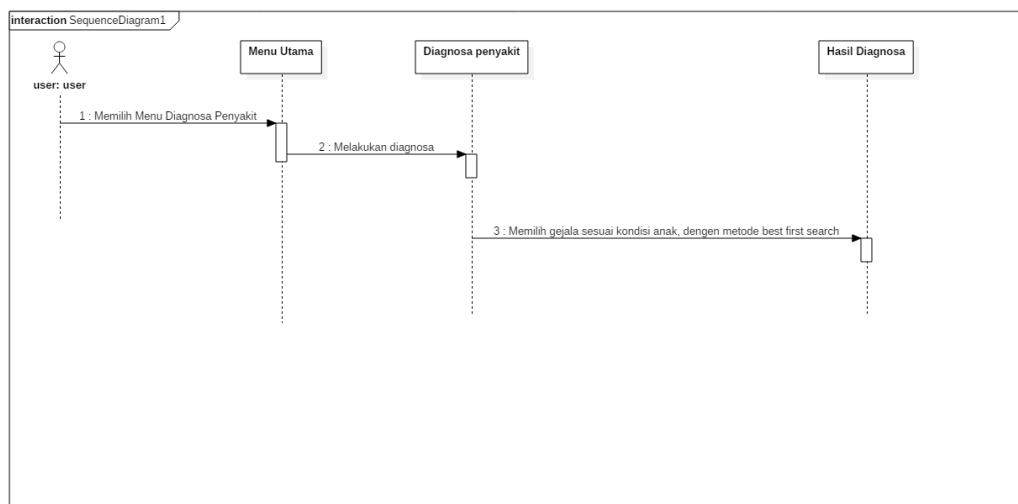


Gambar 3.7 Class Diagram

3.7.2.4 Sequence Diagram

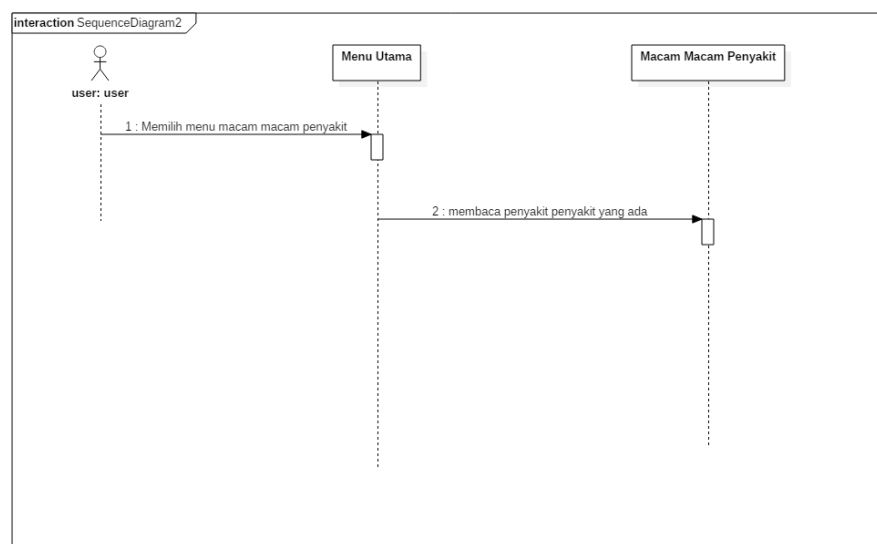
Sequence diagram mendeskripsikan bagaimana entitas dalam sistem berinteraksi, termasuk pesan yang digunakan saat interaksi. Semua pesan dideskripsikan dalam urutan eksekusi. *Sequence diagram* berhubungan erat dengan *Use Case diagram*. Adapun *sequence diagram* yang digunakan dalam sistem pakar penyakit anak ini direpresentasikan pada gambar berikut:

Sequence Diagnosa penyakit adalah untuk melakukan Diagnosa penyakit anak



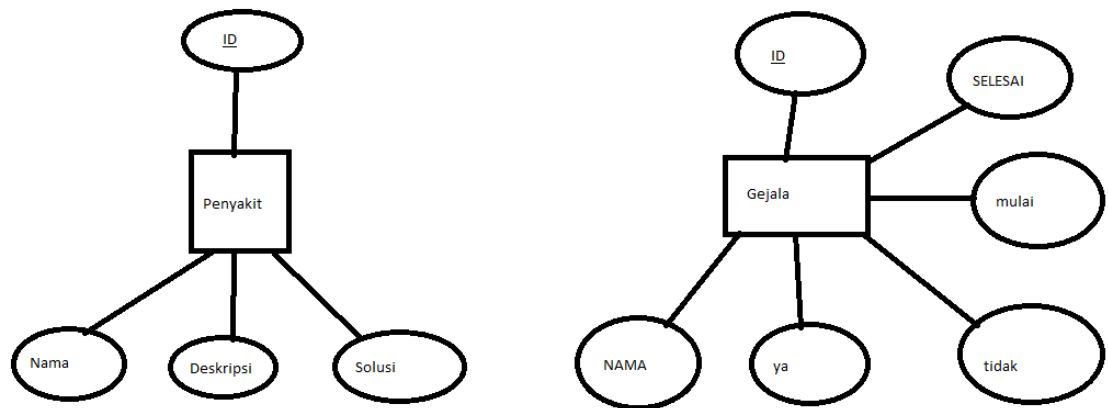
Gambar 3.8 Sequence Diagram Diagnosa penyakit

Sequence Macam macam penyakit adalah untuk melihat macam macam penyakit anak yang diketahui sistem (Gambar 3.9)



Gambar 3.9 Sequence Diagram Macam Macam Penyakit

3.7 Perancangan Basis Data



Gambar 3.10 ERD

3.8.1 Perancangan Basis Data

Struktur tabel digunakan untuk menjelaskan detail atau keterangan-keterangan field-field pada suatu tabel dalam basis data yang digunakan dalam pembuatan sistem Pakar penyakit anak.

a. Tabel gejala

Tabel penyakit adalah tabel yang menyimpan data tentang penyakit, sekaligus menyimpan nama nama node posisi metode dalam sistemnya

3.6 Tabel Gejala

NO	NAMA	TIPE DATA	KETERANGAN
1	ID	Int	Primary key
2	Nama	text	Menyimpan nama gejala
3	Ya	text	Menyimpan ID gejala jika user menekan YA maka akan lanjut ke gejala sesuai IDNya
4	Tidak	text	Menyimpan ID penyakit jika user menekan TIDAK maka akan lanjut ke gejala sesuai IDNya
5	Mulai	text	Jika valuenya YA maka diagnose dimulai dari gejala ini
6	Selesai	text	Jika bervalue YA maka diagnosa akan berakhir langsung menuju ke hasil diagnosa

b. Tabel Penyakit

Tabel penyakit merupakan tabel yang memuat data-data penyakit.

3.7 Tabel Penyakit

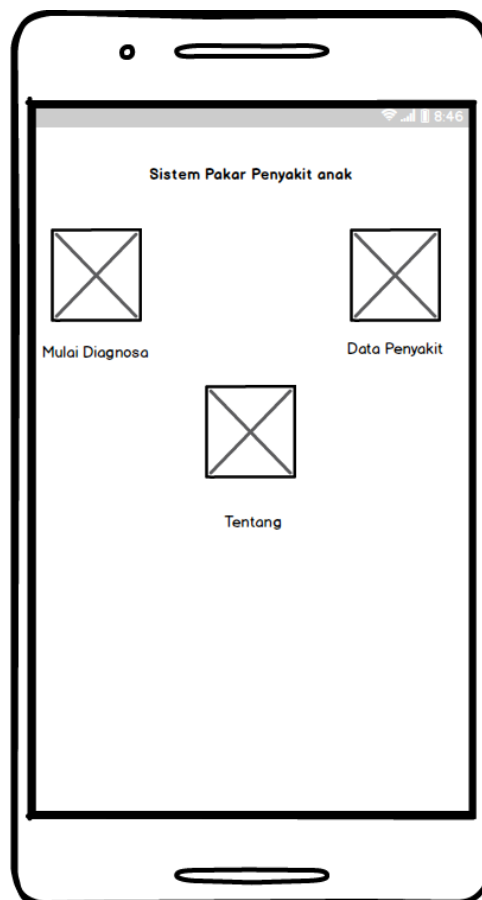
NO	NAMA	TIPE DATA	KETERANGAN
1	ID	Int	Primary key
2	Nama	text	Menyimpan nama penyakit

3	Deskripsi	text	Menyimpan deskripsi dari penyakit
4	solusi	text	Menyimpan data solusi solusi apa saja yang bisa dilakukan dari penyakit

3.8 Perancangan Antar Muka

Perancangan antar muka adalah tampilan menu menu yang akan digunakan oleh user untuk melakukan sistem pakar penyakit anak, tersedia 3 menu yang dapat diakses user yaitu menu Diagnosa penyakit, Macam Macam Penyakit dan Tentang

Menu Utama berisi , menu menu lain yang ada di aplikasi sistem pakar penyakit anak



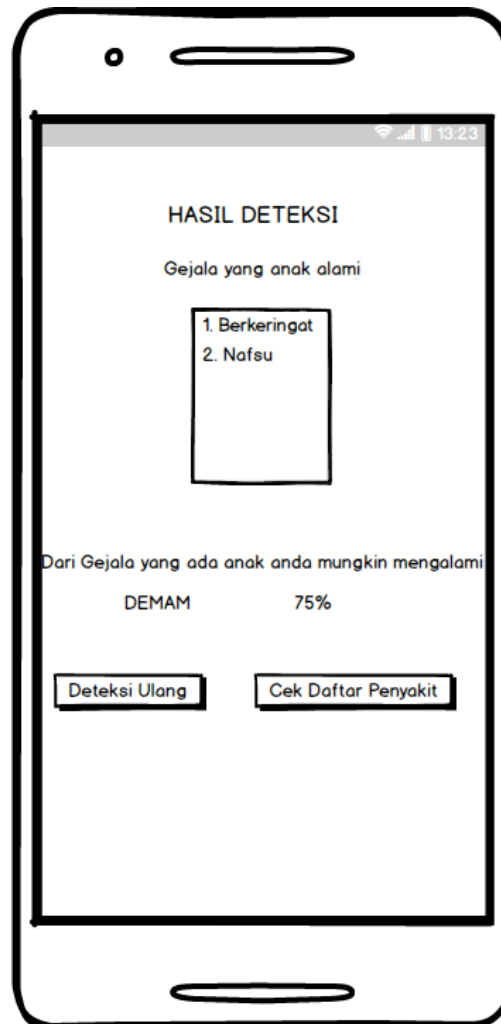
Gambar 3.11 Menu Utama

Tampilan Diagnosa adalah tampilan dimana User melakukan Diagnosa



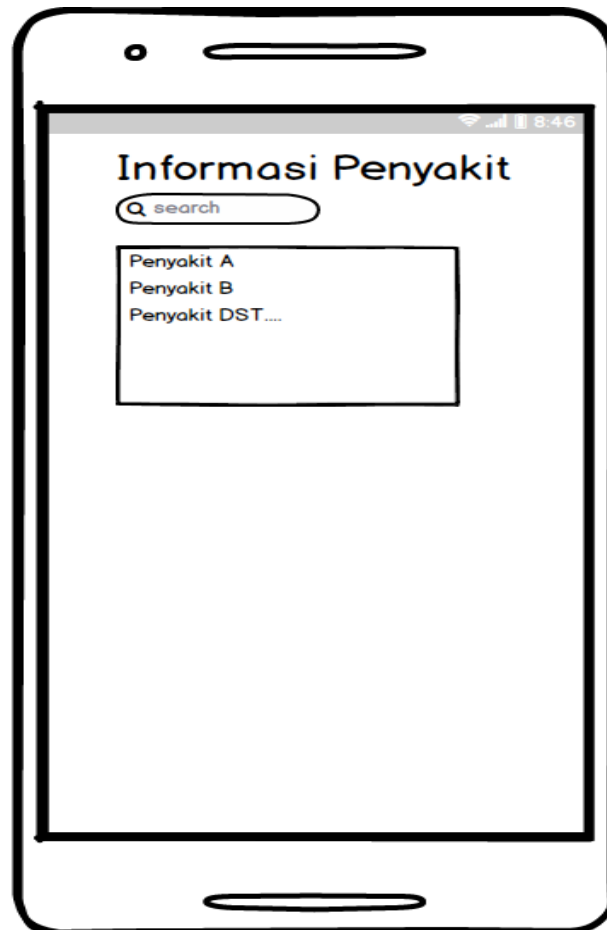
Gambar 3.12 Menu Diagnosa

Tampilan Hasil Diagnosa adalah tampilan setelah User selesai melakukan Diagnosa



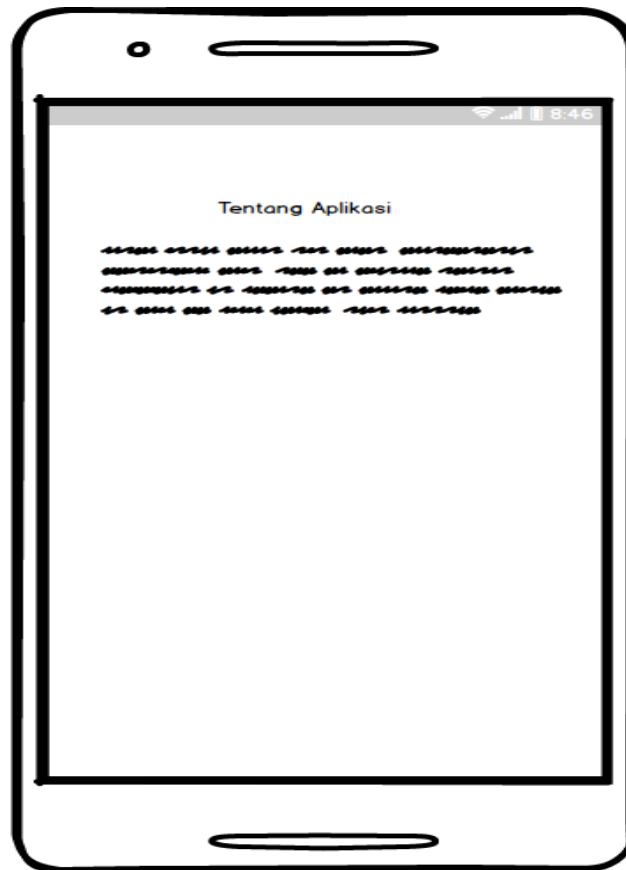
Gambar 3.13 Menu hasil Diagnosa

Halaman Macam macam Penyakit menampilkan, penyakit apa saja yang dikenali sistem



Gambar 3.13 Menu Macam Macam Penyakit

Halaman Tentang, mendeskripsikan tentang aplikasi sistem pakar penyakit anak



Gambar 3.14 Tentang

3.9 Analisis Studi Kasus

Klinik Aulia adalah Klinik yang meyalani Penyakit ibu dan anak terletak di Desa Kait Kait kecamatan Bati Bati Kabupaten Tanah Laut, Provinsi Kalimantan Selatan, adalah salah 1 Klinik yang ada di desa Kait kait, di desa tersebut ada sekitar 6000 kepala keluarga tetapi untuk pelayanan kesehatan seperti Puskesmas dan Posyandu masih kurang untuk melayani kesehatan penduduk disana, berjarak 40 KM dari Kota terdekat (kota Banjarbaru)

Tetangga desa Kait kait meliputi desa Martadah (disebelah utara), Imban (sebelah selatan), juga memiliki masalah serupa yaitu masalah tentang layanan kesehatan yang kurang baik dikarenakan kurangnya klinik, puskesmas maupun hal lain untuk meningkatkan kesehatan di masing masing desa, sehingga klinik terdekat yang ada di Kait kait ini menjadi incaran dari warga di desa lain.

3.9.1 Analisis Pakar

Menjelaskan tentang garis besar seorang pakar yang menjadi sumber dari data yang digunakan penulis dalam merancang sistem ini dan kenapa penulis memilih pakar dalam membangun sistem pakar penyakit anak.

3.9.2 Siti Fitriah

Siti fitriah Lahir di Kota Banjarmasin, 5 Oktober 1975, anak dari 4 bersaudara yang sekarang bekerja sebagai Pegawai Negeri Sipil di Kabupaten Tanah Laut, Kalimantan Selatan, Sembari menjadi Pegawai Negeri Sipil beliau juga membuka usaha sebuah klinik Kesehatan bernama Klinik Aulia, beliau menyelesaikan Studi dengan strata terakhir Diplomat 4 dan memiliki pengalaman 20 tahun sebagai pemilik Klinik sejak berdiri tahun 1997 di Desa kait kait sebagai Petugas kesehatan yang memahami seluk beluk penyakit ibu dan anak, karena itulah penulis memilih beliau sebagai sumber data untuk membuat sistem pakar penyakit anak, dalam sehari Klinik Aulia bisa kedatangan lebih dari 50 Pasien yang 60% periksa tentang kesehatan anak.

3.9.3 Analisis Penyakit

Dalam Sistem ini ada 6 penyakit yang dikenali sistem dari penyakit yang dikenali sistem ini adalah penyakit yang selain umum menyerang anak anak di

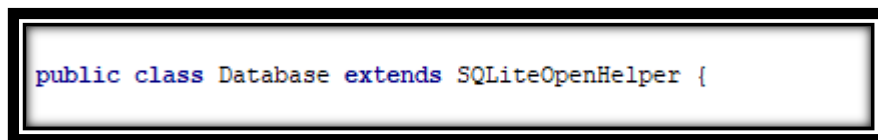
Studi Kasus penulis juga menyerang banyak daerah di seluruh Indonesia, karena penyakit ini memang umum sekali menyerang anak-anak, dan sebagai besar penyakit tak perlu harus ke dokter atau ke rumah sakit, penyakit yang menyerang anak biasanya bisa sembuh bila anak istirahat cukup dan mendapatkan perawatan oleh pihak keluarga, tetapi penyakit anak ini akan tetap berbahaya jika dibiarkan karena gejala-gejala penyakit anak biasanya juga sebagai awal penyakit yang lebih parah, oleh sebab itulah diperlukan penanganan bila penyakit tersebut menyerang.

BAB IV

PEMBAHASAN

4.1 Implementasi SQLite

Dalam Proses penyimpanan data Penyakit dan gejala semua dikerjakan dengan menggunakan *SQLite* yang dimana *SQLite* adalah sebuah aplikasi yang termasuk kedalam paket dari aplikasi *Android Studio* , untuk menggunakan *SQLite* di *Android Studio* dengan membuka *Class Java* baru dengan nama *Database* (nama tak harus database) lalu include kan *SQLiteOpenHelper* untuk deklarasi penggunaan *SQLite* dalam *Android Studio* seperti gambar 4.6

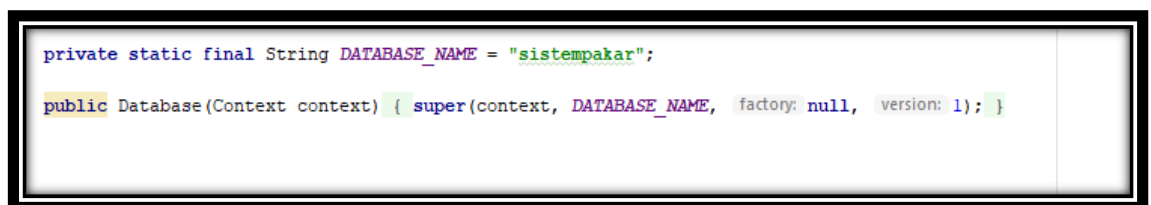


```
public class Database extends SQLiteOpenHelper {
```

Gambar 4.6 Deklarasi *SQLiteOpenHelper*

Lalu dilanjutkan dengan membuat method dan variable penamaan database, sistem pakar adalah nama databasenya seperti gambar4.7

,



```
private static final String DATABASE_NAME = "sistempakar";

public Database(Context context) { super(context, DATABASE_NAME, factory: null, version: 1); }
```

Gambar 4.7 Deklarasi *Variabel* dan penamaan *Database*

Dilanjutkan dengan Coding SQL Pembuatan Table databasenya seperti gambar 4.8 *create Table*

```

public void createTable(SQLiteDatabase db) {
    db.execSQL("DROP TABLE IF EXISTS gejala");
    db.execSQL("CREATE TABLE if not exists gejala (id TEXT PRIMARY KEY, nama TEXT, " +
        "ya VARCHAR(5), tidak VARCHAR(5), mulai VARCHAR(5), selesai VARCHAR(5));");
}

public void createTable2(SQLiteDatabase db2) {
    db2.execSQL("DROP TABLE IF EXISTS penyakit");
    db2.execSQL("CREATE TABLE if not exists penyakit (kode_penyakit TEXT PRIMARY KEY, " +
        "nama_penyakit TEXT, deskripsi TEXT, penanganan TEXT);");
}

```

Gambar 4.8 Create Table

Selesai membuat table dilanjutkan dengan Input data datanya seperti gambar 4.9 dan 4.10

```

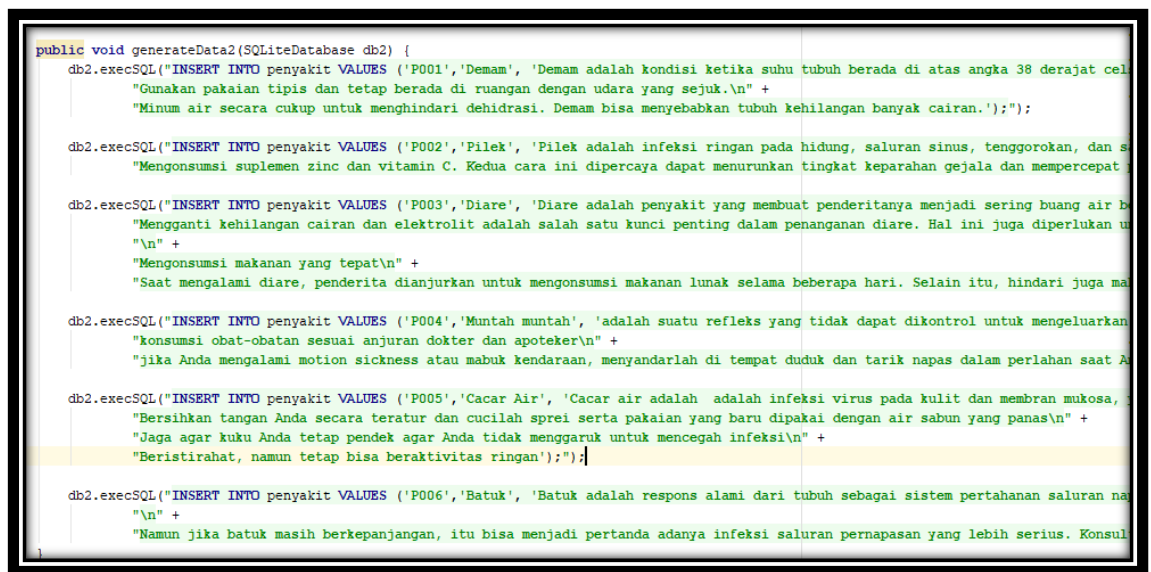
public void generateData(SQLiteDatabase db) {
    ContentValues cv = new ContentValues();

    db.execSQL("INSERT INTO gejala VALUES ('G001','Berkeringat Pada Malam Hari', 'G002', 'G014', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G002','Tubuh terasa lemas', 'G003', 'G006', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G003','nafsu makan berkurang', 'G017', 'G004', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G004','bersin bersin', 'G005', 'G005', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G005','hidung tersumbat', 'G007', 'G007', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G006','tenggorokan kering', 'G021', 'G021', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G007','hidung berlendir', 'G016', 'G016', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G008','sering BAB', 'G009', 'G015', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G009','BAB nya encer / mencret', 'P003', 'P003', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G011','terlalu banyak makan', 'G012', 'G001', 'Y', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G012','Memakan makanan yang asing ', 'G013', 'G013', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G013','mual mual ', 'P004', 'P004', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G014','Pusing', 'G008', 'G020', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G015','Sakit perut ', 'G016', 'G016', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G016','mengigil', 'G018', 'G018', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G017','Tubuh Panas', 'P001', 'P001', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G018','Muncul Bintik Merah', 'G019', 'G019', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G019','Gatal gatal', 'P005', 'P005', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G020','sehat', 'G001', 'B', 'T', 'T');");
    db.execSQL("INSERT INTO gejala VALUES ('G021','Sering mengeluarkan suara Uhuk Uhuk dari mulut', 'P006', 'P006', 'T', 'T');");

    db.execSQL("INSERT INTO gejala VALUES ('P001','Demam', '0', '0', 'T', 'Y');");
    db.execSQL("INSERT INTO gejala VALUES ('P002','Pilek', '0', '0', 'T', 'Y');");
    db.execSQL("INSERT INTO gejala VALUES ('P003','Diare', '0', '0', 'T', 'Y');");
    db.execSQL("INSERT INTO gejala VALUES ('P004','Muntah Muntah', '0', '0', 'T', 'Y');");
    db.execSQL("INSERT INTO gejala VALUES ('P005','Cacar Air', '0', '0', 'T', 'Y');");
    db.execSQL("INSERT INTO gejala VALUES ('P006','Batuk', '0', '0', 'T', 'Y');");
}

```

Gambar 4.9 Insert Data Gejala



Gambar 4.10 Insert data penyakit

4.2 Implementasi Metode Best First Search

Metode Best First Search dalam perancangan sistem pakar penyakit anak membagi gejala-gejala dalam sebuah Node Level yang dimana Level 0 belum dimulai, gejala yang paling banyak memiliki penyakit yaitu Gejala dengan Kode Penyakit G011, G001 dan G014 karena memiliki lebih dari 1 penyakit sehingga masuk ke dalam Node Level 1, maka pencarian berikutnya adalah pencarian dengan melihat pohon keputusannya seperti terlihat dalam tabel gejala yang memiliki *Field* yaitu YA, TIDAK, MULAI dan SELESAI.

Field Ya memiliki Value adalah Kode Gejala berapa bila Pengguna menekan YA, misal Gejala G011 di *FIELD* YA memiliki Value G012 maka diagnosa berikutnya akan memunculkan gejala dari kode G012.

TIDAK berisi value kode gejala saat pengguna dalam melakukan diagnosa memilih TIDAK maka, gejala selanjutnya yang ditampilkan Gejala dengan Kode

sesuai Value di field Tidak, misal G011, dalam field TIDAK memiliki value kode gejala G001, maka saat pengguna memilih tidak akan dilanjutkan ke gejala G001

Mulai adalah *Field* adalah *field* untuk menentukan kode gejala mana yang akan ditampilkan di awal diagnosa, disini G011 adalah awal diagnosa maka dari itu hanya gejala dengan kode G011 yang nilai Mulainya adalah “Y”

Selesai adalah field akhir, jadi saat diagnosa mencapai di Node dimana memiliki gejala yang memiliki *Value field* selesainya adalah “Y” maka diagnosa selesai dilanjutkan ke halaman hasil diagnosa.

Hasil persentase dari setiap penyakit ditentukan dari berapa banyak dalam 1 penyakit menyimpan berapa gejala, jumlah gejala sesuai pohon keputusan dan juga pembobotan misal Penyakit Pilek di Pohon keputusan ada 5 gejala dan di Tabel pembobotan 1 Gejala bernilai 20% jadi untuk diagnosa hanya memilih 4 gejala saja muncul hasil 80% dengan nama penyakit pilek, untuk menampilkan penyakit mana yang akan ditampilkan adalah penyakit dengan persentase tertinggi misal pengguna memilih gejala yang dimana awalnya ada di Penyakit A tetapi pada akhir diagnosa ternyata memilih banyak penyakit B maka hasil diagnosa hanya menampilkan gejala penyakit B dan menghasilkan Persentase penyakit B saja.

4.2.1 Implementasi Metode dalam Program

Implementasi Metode dimulai dari membuat Pohon keputusan sebagai basis pengetahuan untuk menentukan arah Node yang akan berjalan, implementasi Metode di Program sistem pakar penyakit anak dilihat dari gambar 4. Bisa dilihat bahwa G000 adalah saat diagnosa belum berjalan, jika diagnosa berjalan dimulai dari G011 jika YA maka dilanjutkan ke G012 jika tidak G001, untuk Node Level

selanjutnya dengan Keterangan Y/T maka pengguna memilih apapun akan tetap lanjut ke gejala yang sama, jika pengguna memilih T tetapi dalam node sebelumnya punya cabang ke Node lain maka pilihan gejala akan berpindah ke cabang lainnya

Dari pohon keputusan diatas didapatkan basis pengetahuan sebagai berikut

Tabel 4.1 Basis Pengetahuan

Kode	Gejala	Ya	Tidak
G001	Berkeringat pada malam hari	G002	G014
G002	Tubuh terasa lemas	G003	G006
G003	nafsu makan berkurang	G017	G004
G004	bersin bersin	G005	G005
G005	hidung tersumbat	G007	G007
G006	tenggorokan kering	G014	G014
G007	hidung berlendir	G016	G016
G008	sering BAB	G009	G015
G009	BAB nya encer / mencret	P003	P003
G011	terlalu banyak makan	G012	G001
G012	Memakan makanan yang asing	G013	G013
G013	mual mual	P004	P004
G014	Pusing	G008	G008
G015	Sakit perut	G016	G016
G016	mengigil	P002	P002

G017	Tubuh Panas	P001	P001
G018	muncul bintik merah	G019	G019
G019	gatal gatal	P005	P005
G021	sering terdengar suara uhuk uhuk dari anak	P006	P006

Sehingga dibuat tabel gejala dengan source coding

Tabel 4.2 Source Coding Gejala

NO	Source code
1	<pre>db.execSQL("CREATE TABLE if not exists gejala (id TEXT PRIMARY KEY, nama TEXT, " + "ya VARCHAR(5), tidak VARCHAR(5), mulai VARCHAR(5), selesai VARCHAR(5));");</pre>
2	<pre>db.execSQL("INSERT INTO gejala VALUES ('G001','Berkeringat Pada Malam Hari', 'G002', 'G014', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G002','Tubuh terasa lemas', 'G003', 'G006', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G003','nafsu makan berkurang', 'G017', 'G004', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G004','bersin bersin', 'G005', 'G005', 'T', 'T');");</pre>

	<pre> db.execSQL("INSERT INTO gejala VALUES ('G005','hidung tersumbat', 'G007', 'G007', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G006','tenggorokan kering', 'G021', 'G021', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G007','hidung berlendir', 'G016', 'G016', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G008','sering BAB', 'G009', 'G015', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G009','BAB nya encer / mencret', 'P003', 'P003', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G011','terlalu banyak makan', 'G012', 'G001', 'Y', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G012','Memakan makanan yang asing ', 'G013', 'G013', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G013','mual mual ', 'P004', 'P004', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G014','Pusing', 'G008', 'G020', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G015','Sakit perut ', 'G016', 'G016', 'T', 'T');"); </pre>
--	---

	<pre> db.execSQL("INSERT INTO gejala VALUES ('G016','mengigil', 'G018', 'G018', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G017','Tubuh Panas', 'P001', 'P001', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G018','Muncul Bintik Merah', 'G019', 'G019', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G019','Gatal gatal', 'P005', 'P005', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G020','sehat', 'G001', 'B', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('G021','Sering mengeluarkan suara Uhuk Uhuk dari mulut', 'P006', 'P006', 'T', 'T');"); db.execSQL("INSERT INTO gejala VALUES ('P001','Demam', '0', '0', 'T', 'Y');"); db.execSQL("INSERT INTO gejala VALUES ('P002','Pilek', '0', '0', 'T', 'Y');"); db.execSQL("INSERT INTO gejala VALUES ('P003','Diare', '0', '0', 'T', 'Y');"); db.execSQL("INSERT INTO gejala VALUES ('P004','Muntah Muntah', '0', '0', 'T', 'Y');"); db.execSQL("INSERT INTO gejala VALUES ('P005','Cacar Air', '0', '0', 'T', 'Y');"); </pre>
--	---

	<code>db.execSQL("INSERT INTO gejala VALUES ('P006','Batuk', '0', '0', 'T', 'Y');");</code>
--	---

Source Coding no 1 dalam tabel 4. Adalah source coding create tabel dan source code no 2 adalah meinputkan data dari tabel

Tabel 4.3 Keterangan Field Gejala

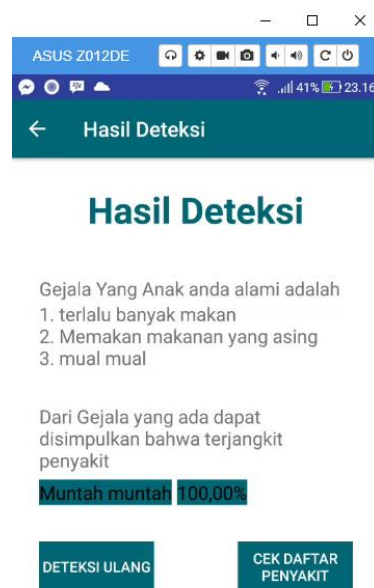
NO	Nama Field	Keterangan
1	ID	Menyimpan kode gejala, ID ini yang akan menjadi value dalam penelusuran Best First Search nya
2	Nama	Nama gejala
3	ya	Field yang berfungsi sebagai arah gejala selanjutnya bila pengguna memilih ya di radio button lalu lanjut maka akan menampilkan gejala sesuai dengan value kode gejala penyakit lain di fieldnya
4	tidak	Field yang berfungsi sebagai arah gejala selanjutnya bila pengguna memilih tidak di radio button lalu lanjut maka akan menampilkan gejala sesuai dengan value kode gejala penyakit lain di fieldnya
5	mulai	Bagi gejala yang memiliki value Y di gejala penyakitnya maka akan menjadi gejala pertama yang akan ditampilkan dalam program ini hanya G011 yang memiliki value Y di fieldnya selain itu maka value field lain adalah T

6	selesai	Jika didalam field tersebut memiliki value Y maka diagnosa berakhir dan akan menuju ke hasi diagnosa, jika T maka masih ada diagnosa yang tersisa, field yang miliki value Y di program sistem pakar penyakit anak ini adalah kode penyakitnya
---	---------	--

4.5 Uji Diagnosa penyakit

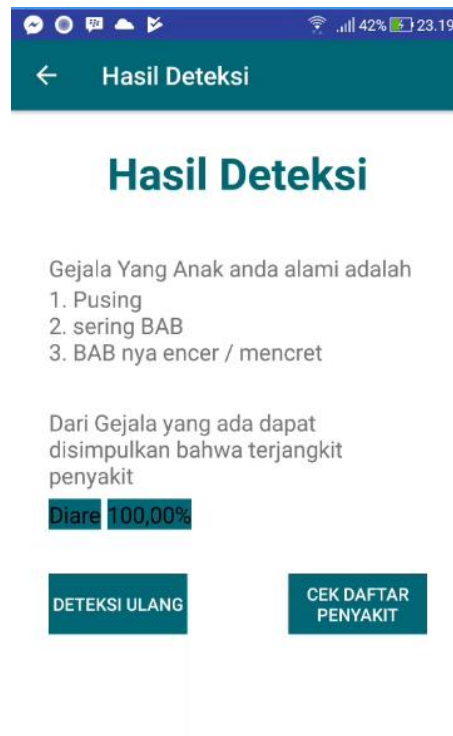
Uji Coba Diagnosa ini akan memilih beberapa pilihan gejala berbeda sebanyak 3 kali untuk membandingkan masing masing hasil persentase.

Percobaan pertama saat melakukan diagnosa hanya memilih Pilihan YA sebanyak 3 kali sehingga didapatkan hasil diagnosa sebagai berikut :



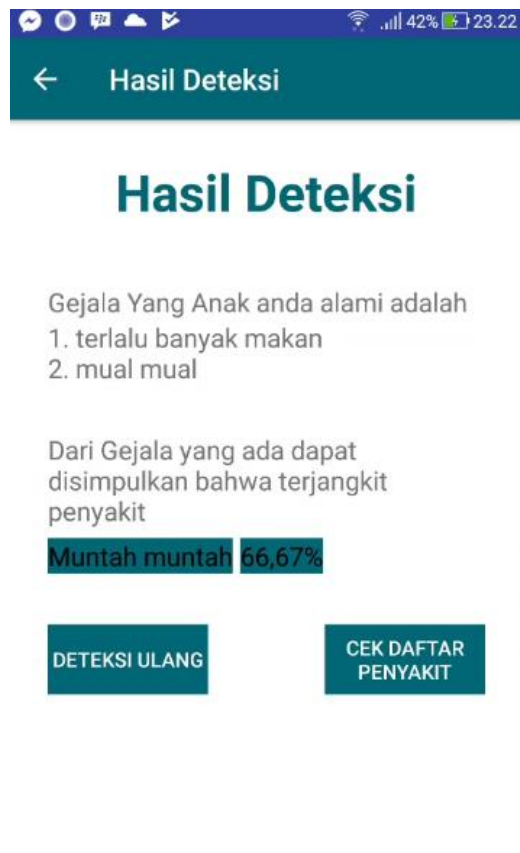
Gambar 4.11 Percobaan pencarian pertama

Pada Percobaan Kedua memilih TIDAK sebanyak 2 kali lalu dilanjutkan dengan Memilih YA sampai diagnosa selesai, didapatkan hasil diagnosa sesuai gambar 4.12



Gambar 4.12 Percobaan pencarian kedua

Untuk percobaan terakhir akan dilakukan diagnose dengan menekan YA untuk gejala pertama lalu TIDAK untuk gejala kedua, dan YA lagi di gejala ketiga dan TIDAK di gejala keempat dan seterusnya sehingga mendapatkan hasil diagnose, didapatkan hasil diagnosa seperti gambar 4.13



Gambar 4.13 Percobaan pencarian ketiga

4.6 Pengujian Sistem

Pengujian sistem dilakukan untuk mencari kelemahan, atau kesalahan sistem, sehingga memastikan sistem bebas dari kesalahan maupun meminimalisir kesalahan yang ada. Metode pengujian yang akan dilakukan terdiri dari *black box testing* dan *white box testing*[7].

4.6.1 Black Box Testing

Metode *black box testing* dilakukan untuk mencari kesalahan dengan melakukan uji coba pada *interface software*. Pegujian kotak hitam (*black box*

Testing) mendemonstrasikan fungsi dari perangkat lunak yang beroperasi, dengan mengecek apakah input sudah bisa diterima dengan baik, dan hasil outputnya sesuai dengan apa yang diharapkan. Uji coba menggunakan metode *black box* ini melakukan pengecekan pada integritas informasi eksternal dimana pengujian hanya memeriksa hasil output yang dihasilkan apakah sudah sesuai dengan apa yang diharapkan dan dinyatakan benar tanpa perlu mengecek logika dari perangkat lunak tersebut[7].

1. Black Box testing Alur Diagnosa

Tabel 4.4 Pengujian Diagnosa

No	Nama Pengujian	Kondisi Pengujian	Harapan	Hasil
1	Halaman Diagnosa	Memilih ya pada saat diagnosa	Ketika pengguna memilih YA maka akan dilanjutkan ke gejala sesuai pohon keputusan	Valid
	Halaman Diagnosa	Memilih tidak pada saat diagnosa	Ketika pengguna memilih Tidak maka akan dilanjutkan ke gejala sesuai pohon keputusan	Valid

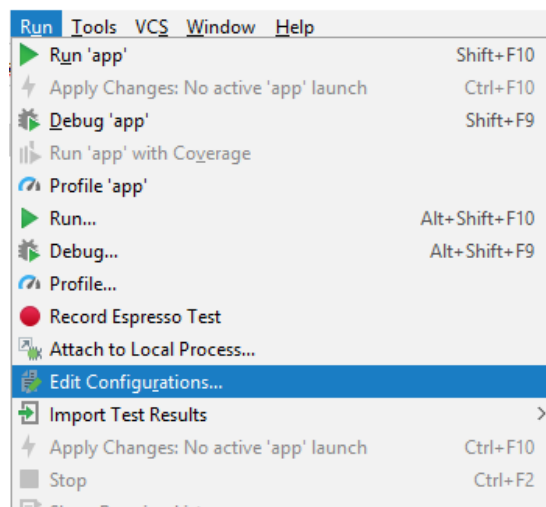
2. Black Box testing Hasil Diagnosa

Tabel 4.5 Pengujian Halaman diagnosa

No	Nama Pengujian	Kondisi Pengujian	Harapan	Hasil
1	Halaman Hasil Diagnosa	Menyelesaikan semua gejala diagnosa	Hasil diagnosa menampilkan nama penyakit sesuai gejala yang telah dipilih	Valid
2	Halaman Hasil Diagnosa	Menyelesaikan semua gejala diagnosa	Hasil diagnosa memberikan persentase sesuai dengan kondisi saat pemilihan gejala	Valid

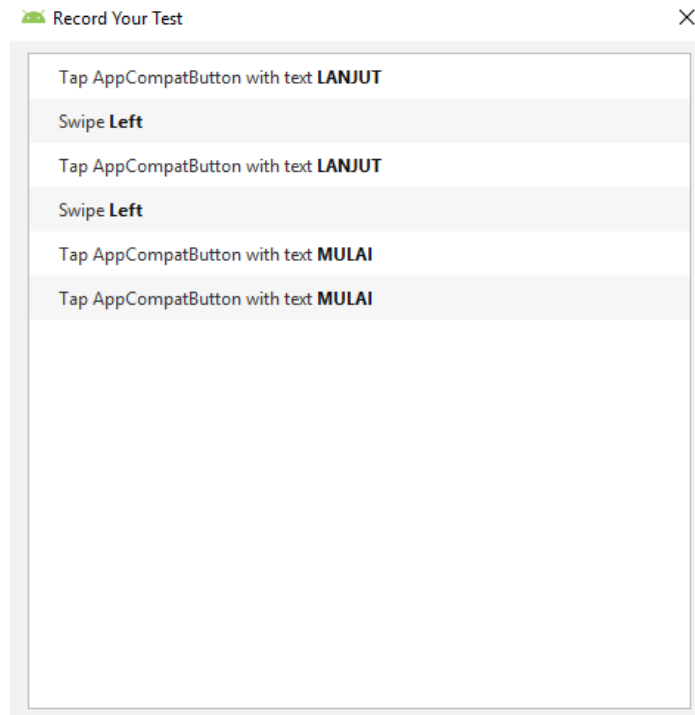
4.6.2 White Box Testing

White box testing disini akan menggunakan Tools yang disediakan di *Android Studio* yaitu, *Ekspresso unit testing* dari tools ini akan melakukan test terhadap sistem secara langsung saat digunakan, lalu menampilkan dalam sebuah window apa apa saja yang sudah bisa digunakan. Untuk melakukan testing ini cukup RUN program dengan Tools *Record Espresso Test* di Menu RUN



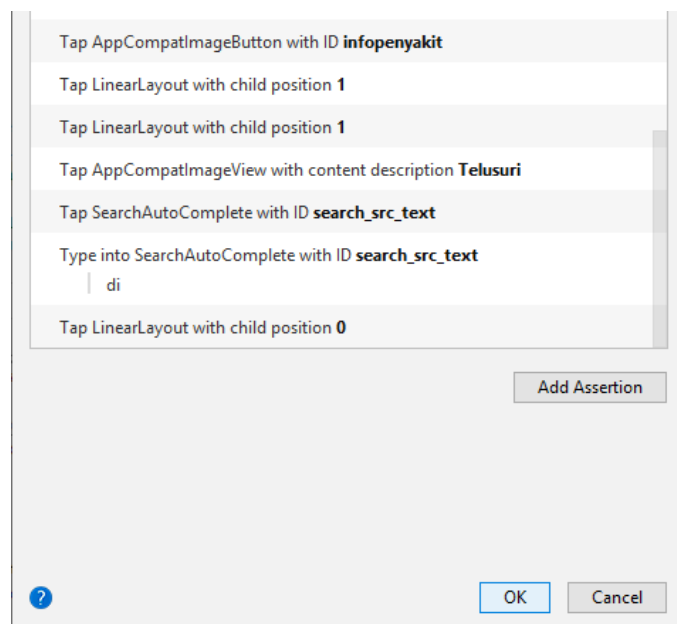
Gambar 4.14 *Run Record Espresso Test*

Hasil Testing dengan Espresso untuk pengujian Home Welcome, terlihat pada gambar dibawah ini bahwa Pengujian Tombol Lanjut, Swipe dan Mulai Berjalan



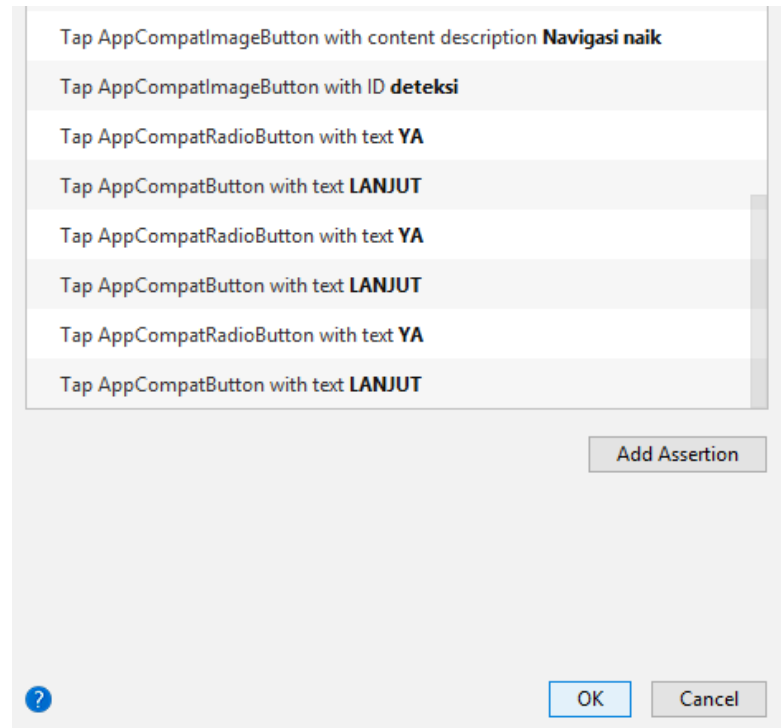
Gambar 4.15 Unit Testing Welcome Home

Pada Pengujian di Data penyakit Tombol Info penyakit, terlihat seperti gambar dibawah ini ImageButton info penyakit berjalan, tombol pencarian juga berjalan.



Gambar 4.16 Unit Testing Data Penyakit

Pada pengujian Ekspreso saat melakukan diagnosa. Terlihat ImageButtonNavigasiDiNaik berfungsi untuk melakukan kembali ke Page sebelumnya, lalu melakukan diagnosa dengan cara memilih YA di setiap Gejala, seperti terlihat pada gambar dibawah ini.



Gambar 4.17 Unit Testing Ekspreso Diagnosa Penyakit

4.6.3 Kesimpulan Hasil Pengujian

Dari Hasil Pengujian *Black box* dan *White box* Testing didapatkan hasil bahwa sistem pakar penyakit anak, berhasil berjalan dengan metode *Best First Search* , terlihat dari tak ditemukan error saat melakukan Pengujian dari sisi *Black box* dan *White box Testing*.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari Sistem Pakar Penyakit Anak menggunakan Metode *Best First Search* dapat ditarik kesimpulan sebagai berikut:

1. Sistem Pakar Penyakit Anak dibangun dengan merancang pohon keputusan, UML dan ERD
2. Implementasi Metode dilakukan dengan cara membangun Pohon keputusan lalu di susun kedalam database untuk menentukan node node lanjutannya
3. Sistem melakukan diagnosa dengan memperhatikan pohon keputusannya, dengan memilah berbagai node yang ditelusuri yang sudah di deklarasikan dalam database

5.2 Saran

Perancangan Sistem Pakar Penyakit anak Berbasis Android dengan Metode Best First Search tentu saja memiliki kekurangan-kekurangan yang diharapkan dapat diperbaiki dan dikembangkan kembali oleh para peneliti selanjutnya, saran pengembangan antara lain :

1. Data penyakit dan Gejala lebih banyak dan mencakup banyak penyakit hingga penyakit yang lebih spesifik
2. Bisa melakukan *Insert, Update, Select* dan *Delete* pada sistem sesuai keperluan yang akan digunakan
3. Bisa menampilkan dan memberikan solusi lebih dari 1 penyakit, karena 1 gejala bisa memiliki banyak penyakit

DAFTAR PUSTAKA

- [1] Huzainsyahnoor Aksad, Taufiqurrahman . 2014. *Sistem Pakar Untuk Mendiagnosa Penyakit Pada Balita Berbasis DFS* . Jurnal OJS STMIK Banjar Baru (JUTISI)
- [2] Sulthan Noor Ridho, Andi Farmadi, Dwi Kartini. 2017 . *SISTEM PAKAR PENYAKIT MATA GLAUKOMA DENGAN METODE BREADTH-FIRST SEARCH (BFS) DAN FUZZY TSUKAMOTO* . Jurnal Elektronik Nasional Teknologi dan Ilmu Komputer (JENTIK)
- [3] Yoyon Efendi.2016. *Diagnosis Kanker Darah pada Anak menggunakan Inferensi Forward Chaining* . SATIN – Sains dan Teknologi Informasi
- [4] Muhammad Syaifuddin, Anton Setiawan Honggowibowo. 2014. *SISTEM PAKAR DIAGNOSA PENYAKIT BAYI DAN BALITA BERBASIS ANDROID DENGAN MENGGUNAKAN Depth First Search* . E-JURNAL STTA
- [5] Tsai-ChiKuo^a, ShanaSmith^b, Gregory C.Smith^c, Samuel H.Huang^d. 2016. *A predictive product attribute driven eco-design process using depth-first search*. *Journal of Cleaner Production*
- [6] Dismar. 2011. *SISTEM PAKAR DIAGNOSA PENYAKIT MELALUI TELAPAK*.Pekanbaru. Skripsi FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
- [7] WIERFI, A. D. (2017). *PERANCANGAN DAN IMPLEMENTASI ALGORITMA EXTENDED WEIGHTED TREE SIMILARITY PADA LAYANAN INFORMASI PENCARIAN LOWONGAN KERJA BERBASIS WEBSITE*. Yogyakarta: Skripsi Universitas AMIKOM Yogyakarta.