

# Sistem Pakar Diagnosa Penyakit Anak Menggunakan Metode Forward Chaining Berbasis Mobile

Christian Ramba Pasalli<sup>(1)</sup>, Vecky. C Poekoel<sup>(2)</sup>, Xaverius Najoan<sup>(3)</sup>  
Teknik Informatika, Universitas Sam Ratulangi, Manado, Indonesia.  
Email: [chtapasalli@gmail.com](mailto:chtapasalli@gmail.com), [xnajoan@unsrat.ac.id](mailto:xnajoan@unsrat.ac.id)

**Abstrak** --- Kesehatan merupakan hal yang paling berharga bagi manusia, karena siapa saja rentan mengalami gangguan kesehatan. Anak sangat rentan terhadap kuman penyakit dan kepekaan terhadap gejala suatu penyakit merupakan ketakutan sendiri bagi orang tua. Namun dengan kemudahan adanya dokter ahli, terkadang terdapat pula kelemahan seperti jam kerja atau jam praktek dokter yang terbatas dan banyaknya pasien sehingga harus menunggu antrian. Karena hal itulah maka dibutuhkan suatu alat bantu yang dapat mendiagnosa penyakit anak berupa suatu sistem pakar.

Tujuan penelitian ini adalah membuat aplikasi sistem pakar berbasis mobile dalam mendiagnosis penyakit anak. Dengan menggunakan teknik penalaran forward chaining diagnosa dilakukan dengan memulai dari sekumpulan gejala-gejala, nantinya dapat melihat kesimpulan jenis penyakit pada anak. Metode yang digunakan sebagai tahapan penelitian ini adalah metode Extreme Programming (XP) yang merupakan metode rancang bangun perangkat lunak yang menekankan pada 4 tahap dalam pengembangan perangkat lunak.

Kata Kunci : Sistem Pakar, Forward Chaining, Extreme Programming, Mobile

## I. PENDAHULUAN

Perkembangan ilmu kedokteran mengalami kemajuan yang pesat ditandai dengan ditemukannya penyakit - penyakit tropis yang belum teridentifikasi sebelumnya. Para dokter ahli terus menemukan solusi untuk mengatasi penemuan terbaru dan selalu memberikan pelayanan terbaik kepada pasien. Kesehatan merupakan hal yang paling berharga bagi manusia, karena siapa saja rentan mengalami gangguan kesehatan. Anak sangat rentan terhadap kuman penyakit dan kepekaan terhadap gejala suatu penyakit merupakan ketakutan sendiri bagi orang tua. Orang tua merupakan orang awam terhadap dunia kesehatan. Apabila terjadi gangguan kesehatan terhadap anak maka mereka lebih mempercayakan kepada dokter ahli yang sudah mengetahui tentang dunia kesehatan tanpa memedulikan apakah gangguan tersebut masih tergolong tingkat rendah atau sudah kronis. Namun dengan kemudahan dengan adanya dokter ahli, terkadang terdapat pula kelemahan seperti jam kerja atau jam praktek dokter yang terbatas dan banyaknya pasien sehingga harus menunggu antrian. Sehingga dalam hal ini orang tua lebih membutuhkan seorang pakar yang bisa memudahkan mendiagnosa suatu penyakit lebih cepat agar orang tua dapat melakukan pencegahan lebih awal yang sekiranya bisa membutuhkan waktu lebih lama jika berkonsultasi dengan dokter ahli. Karena hal itulah maka dibutuhkan suatu alat

bantu yang dapat mendiagnosa penyakit anak berupa suatu sistem pakar. Sistem pakar adalah program berbasis pengetahuan yang menyediakan solusi-solusi untuk problema-problema dengan kualitas pakar. Sistem pakar merupakan program komputer yang meniru proses pemikiran dan pengetahuan pakar dalam menyelesaikan suatu masalah tertentu. Sistem pakar dibangun untuk mencoba menyerupai kemampuan manusia dalam menyelesaikan masalah tertentu dalam bentuk heuristik. Implementasi dari sistem pakar juga dapat diterapkan di segala bidang, termasuk dalam bidang kesehatan.

## II. LANDASAN TEORI

### 1. Sistem Pakar

Menurut Kusumadewi (2003), sistem pakar adalah suatu cabang dari *artificial intelligence* yang berusaha mengadopsi pengetahuan manusia ke komputer agar dapat menyelesaikan masalah seperti yang biasa diselesaikan oleh para pakar. Dengan adanya sistem ini, orang awam pun dapat menyelesaikan masalah yang rumit yang sebenarnya hanya dapat diselesaikan oleh para ahli. Sedangkan bagi para ahli, sistem pakar dapat membantu aktivitasnya sebagai asisten yang berpengetahuan. Ada beberapa dasar mengapa sistem pakar menggantikan seorang pakar diantaranya adalah : [1]

- a) Dapat menyediakan kepakaran setiap waktu diberbagai lokasi
- b) Secara rutin mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar
- c) Seorang pakar akan pensiun atau pergi
- d) Seorang pakar adalah mahal
- e) Kepakaran juga dibutuhkan pada lingkungan yang tidak bersahabat (*hostile environment*)

Dengan demikian seorang awam sekalipun bisa menggunakan sistem pakar itu untuk memecahkan berbagai persoalan yang ia hadapi dan bagi seorang ahli, sistem pakar dapat dijadikan alat untuk menunjang aktivitasnya yaitu sebagai asisten yang berpengalaman. Sistem pakar untuk melakukan diagnosa kesehatan telah dikembangkan sejak pertengahan tahun 1970 yang pertama kali dibuat oleh Bruce Buchanan dan Edward Shortliffe di Stanford University diberi nama MYCIN. MYCIN merupakan program interaktif yang melakukan diagnosa penyakit meningitis dan infeksi bacemia serta memberikan rekomendasi terapi antimikroba. MYCIN mampu memberikan penjelasan tentang penalarannya secara detail. Dalam uji coba, program ini mampu menunjukkan kemampuan seperti seorang spesialis.[1]

## 2. Mesin Inferensi

Menurut Turban menyebutkan mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace, dan untuk merumuskan kesimpulan. Terdapat dua pendekatan dalam mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu :

### 1) Pelacakan kedepan (forward chaining)

Pelacakan kedepan adalah pendekatan yang terkendali-data (*data-driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan mencari fakta yang sesuai dengan bagian *if* dari aturan *if-then*.

### 2) Pelacakan kebelakang (backward chaining)

Pelacakan kebelakang adalah pendekatan terkendali-tujuan (*goal driven*). Dalam pendekatan ini, pelacakan dimulai dari tujuan selanjutnya dicari aturan yang memiliki tujuan tersebut untuk tujuannya.[4]

## 3. Android

Android merupakan sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi linux, namun telah dimodifikasi. Android diambil alih oleh google pada tahun 2005 dari android. Ini sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja android termasuk tim yang mengembangkan android. Secara garis besar sistem operasi android menjadi 5 tingkatan:

### 1) Linux Kernel

Linux kernel adalah kernel dasar dari android. Tingkat ini berisi semua driver perangkat tingkat rendah untuk komponen-komponen perangkat android.

### 2) Libraries

Libraries berisi semua kode program yang menyediakan layanan-layanan utama sistem operasi android.

### 3) Android Runtime

Android runtime kedudukannya sama dengan libraries, android runtime menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi dengan bahasa pemrograman java.

### 4) Application Framework

Application Framework adalah semacam kumpulan class built in yang tertanam dalam sistem operasi android sehingga pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.[3]

## 4. Extreme Programming

Extreme Programming (XP) adalah metode pengembangan perangkat lunak yang ringan dan termasuk salah satu agile methods yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP merupakan agile methods yang paling banyak digunakan dan menjadi sebuah pendekatan yang sangat terkenal. Sasaran XP adalah tim yang dibentuk berukuran antara kecil sampai medium, tidak perlu menggunakan sebuah tim yang besar. Hal ini dimaksudkan untuk menghadapi requirements yang tidak jelas maupun terjadinya perubahan-perubahan requirements yang sangat cepat. XP sebagai sebuah metode yang dinamis diperlihatkan

dalam empat values yang dimilikinya dan keempatnya merupakan dasar-dasar yang diperlukan dalam XP. Kent Beck menyatakan bahwa tujuan jangka pendek individu sering berbenturan dengan tujuan sosial jangka panjang. Karena itu dibuatlah values yang menjadi aturan, hukuman, dan juga penghargaan. Keempat values tersebut adalah :[2]

1) Komunikasi (Communication)

2) Kesederhanaan (Simplicity)

3) Umpan Balik (Feedback)

4) Keberanian (Courage)

Menurut Kent Beck, XP ringan, efisien, resiko rendah, mudah disesuaikan, dapat diprediksi, ilmiah, dan mudah dikembangkan. Suatu model yang menekankan pada keterlibatan pengguna secara langsung, pengujian dan *pay-as-you-go design*. [2]

## III. METODE PENELITIAN

### 1. Metodologi Penelitian

Untuk membuat sistem pakar diagnosa penyakit anak akan digunakan metodologi extreme programming. Extreme programming merupakan salah satu metode pengembangan perangkat lunak. Berikut tahapan – tahapan dan gambar rincian pada Gambar.1 dari extreme programming :

#### 1) Planning.

Pada tahap ini penulis melakukan kegiatan :

a) Wawancara – Pada kegiatan ini penulis melakukan wawancara dengan para pakar di RSUP Prof Kandou dengan mendapatkan hasil data gejala dan data penyakit.

b) Analisis – Pada kegiatan ini penulis membuat tabel relasi gejala dan penyakit, beserta representasi pengetahuan. Kegiatan ini membantu untuk menyelesaikan masalah dalam ilmu pakar

#### 2) Design

Tahap ini semua hasil analisa kebutuhan sistem pakar didefinisikan dalam bentuk diagram UML dan CRC sedangkan interface sistem dimodelkan dengan desain interface.

#### 3) Coding

Tahap ini aplikasi akan dibuat menggunakan MIT app inventor. Alur aplikasi dimulai dengan ditampilkannya menu pertanyaan dan pilihan gejala. Rancangan sistem dibuat tidak digunakan secara online.

#### 4) Testing

Tahap terakhir yang dilakukan adalah tahap testing. Tahap ini dilakukan dengan menguji coba hasil perancangan sistem yang telah dibuat dengan menggunakan pengujian blacbox dan pengujian fungsional.

## IV. HASIL DAN PEMBAHASAN

### 1. Tahap Planning

Tahapan *planning* pada XP merupakan tahapan awal dari semua proses pengembangan sistem. Tahapan ini merupakan gabungan dari tahap analisis dan *planning* pada metode pengembangan ini.

#### 1) Data dan Gejala Penyakit

Dibawah ini adalah daftar penyakit dan gejala yang dimasukkan ke dalam sistem pakar :Pengidentifikasiian penyakit didasarkan pada gejala – gejala yang dirasakan

dan tanda klinis pada anak. Gejala yang timbul dari penyakit tropis dapat dilihat pada Tabel.2 :

2) Representasi Pengetahuan

Representasi pengetahuan yang akan digunakan dalam sistem pakar ini adalah dengan menggunakan jaringan semantik, pohon keputusan, dan kaidah produksi. Representasi pengetahuan dapat menemukan solusi atau memberikan penalaran yang tepat dalam pengambilan kesimpulan.

a) Jaringan Semantik

Suatu proses terhadap basis pengetahuan atau informasi yang didapat dari pakar, terlebih dahulu diubah ke dalam bentuk jaringan semantik. Jaringan semantik menunjukkan hubungan antara gejala dan penyakit.

b) Pohon Keputusan

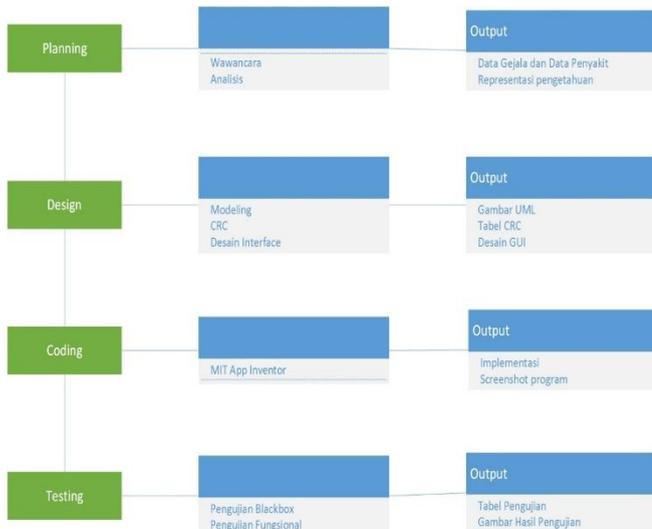
Setelah proses terhadap basis pengetahuan telah diubah kedalam bentuk jaringan semantik, selanjutnya diubah kedalam bentuk pohon keputusan, sehingga didalam penyelesaian masalah lebih mudah dilakukan penelusuran untuk mendapatkan solusi atau kesimpulan akhir yang terbaik. Adapun pohon keputusan pada sistem ini adalah

c) Kaidah Produksi

Basis aturan dalam permasalahan ini merupakan kumpulan kaidah-kaidah yang saling berhubungan satu sama lain. Kaidah-kaidah atau aturan-aturan ini direpresentasikan dalam bentuk persyaratan IF – Then berdasarkan pohon keputusan (gambar 4). Pernyataan ini menghubungkan bagi premios (IF) dan bagian kesimpulan (Then). Apabila premis dalam aturan produksi dapat memiliki lebih dari suatu proposisi, proposisi-proposisi tersebut dihubungkan dengan menggunakan operator logika AND.

2. Tahap Desain

Dalam mendesain aplikasi ini penulis menggunakan pendekatan yaitu membuat diagram UML, CRC, dan desain interface. Maksudnya adalah untuk mempermudah dan mempercepat penyelesaian proses desain. Penulis membuat desain ini supaya pada saat proses pengkodean lebih mudah karena gambaran fungsi-fungsinya sudah jelas.



Gambar 1. Metodologi Penelitian

Tabel.1 Nama Penyakit

Kode	Nama Penyakit
P01	Asma
P02	Bronchopneumonia
P03	Tifoid
P04	DHF (Dengue Haemorrhagic Fever) / DBD
P05	TBC
P06	Tonsilitis
P07	Leukemia
P08	Malaria
P09	Meningitis

Tabel.2 Nama Gejala

Kode	Gejala	Keterangan
G01	Batuk Kering	
G02	Gelisah	Perasaan Khawatir
G03	Sulit Berbicara	
G04	Tingkat Kesadaran Menurun	
G05	Sesak Napas	
G06	Batuk Produktif dan Kuat	
G07	Dada Sesak	
G08	Pernafasan Berbunyi	
G09	Demam 2-4 Hari	Suhu > 38°C
G10	Demam 5-7 Hari	Suhu > 38°C
G11	Mengigil	
G12	Nyeri Otot	
G13	Nyeri Perut Bagian Atas	
G14	Banyak Keringat	
G15	Sakit Kepala	
G16	Muntah	
G17	Diare	
G18	Kurang Nafsu Makan	
G19	Berat Badan Turun	
G20	Lidah Kotor dibagian Tengah	
G21	Ujung Lidah Berwarna Merah	
G22	Nyeri Tenggorokan	
G23	Lesu	
G24	Tangisan Merintih	
G25	Batuk >3 Minggu	Mengeluarkan Darah
G26	Gusi Berdarah	
G27	Kejang	
G28	Keringat Malam Hari	Tanpa Aktivitas
G29	Flu > 3 Minggu	
G30	Memar Tanpa Sebab	
G31	Pembesaran Getah Bening	
G32	Nyeri Tulang	
G33	Demam Turun Secara Tiba Tiba	
G34	Nyeri Menelan	
G35	Muncul Kemerahan Pada Kulit	
G36	Malas Minum	
G37	Pucat	
G38	Mulut Berbau	

1) Use Case

Use case ini menggambarkan tentang interaksi antara aktor (*user*) dengan sistem. Disini digambarkan aktor (*user*) melakukan beberapa kegiatan seperti melihat gejala, melihat hasil, melihat solusi, melihat tentang aplikasi dan keluar. Use case dapat dilihat pada Gambar 3.

2) Desain CRC Card

Proses desain pada tahap ini yaitu menggunakan *CRC Responsibilities, and Collaboration (CRC) card*. Penggunaan CRC card hanya dipakai jika pengembangan *software* berbasis objek. Setiap CRC card akan menjelaskan tiap objek yang dibutuhkan. Desain CRC card dapat dilihat pada Tabel.3.

3) Desain GUI

Pada tahap ini, keseluruhan perancangan secara grafis dengan tujuan untuk memudahkan user dalam menjalankan aplikasi dan dapat dengan mudah memahami informasi yang ditampilkan pada aplikasi tersebut. Dalam tahap ini digunakan prinsip – prinsip dari bidang ilmu Interaksi Manusia dan Komputer. Gambar 4 merupakan rancangan tampilan awal dari aplikasi sistem pakar diagnosa penyakit. Dalam rancangan terdapat 1 gambar dan 4 *button* yang memiliki fungsi masing – masing. Gambar.5 merupakan rancangan tampilan *screen* pertanyaan. Dalam rancangan tampilan *screen* pertanyaan terdiri dari label *text* pertanyaan dan *button* “Ya” *button* “Tidak”. Jika pengguna memilih salah satu *button* maka akan muncul pertanyaan selanjutnya dengan tampilan yang sama. Gambar.6 merupakan rancangan tampilan *screen* kesimpulan. Dalam rancangan ini terdiri dari gambar, label *text*, dan *button*. Tampilan kesimpulan akan muncul setelah pengguna menjawab beberapa pertanyaan dalam aplikasi ini.

3. Tahap Coding

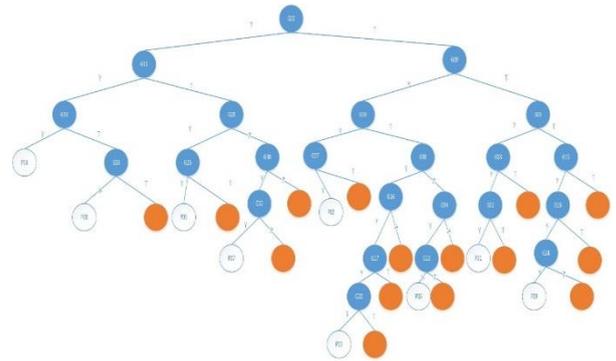
Pada tahap ini penyusunan blok dilakukan. Tahap ini merupakan tahapan yang paling utama dari model proses *extreme programming*. Tahap ini dapat berulang kali dilakukan sesuai dengan kebutuhan fungsional yang dapat berubah sewaktu-waktu.

4. Tahap Testing

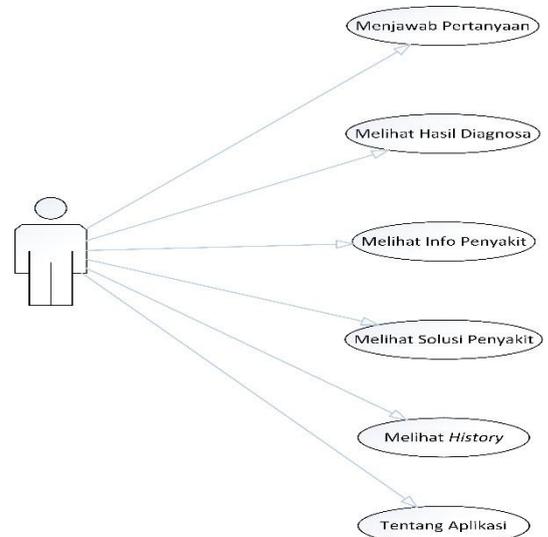
Pada tahap ini dilakukan pengujian program apakah sesuai dengan tugasnya. Pengtesan ini dilakukan dengan menggunakan metode *black box testing* yaitu pengtesan menggunakan metode pengujian logika program yaitu perkasus atau masalah yang diajukan. Pada tahap selanjutnya akan diujikan pada kondisi sebenarnya yaitu dengan pengujian fungsional dan evaluasi pengguna.

1) Blackbox Testing

Karena penulis menggunakan *extreme programming* sebagai model prosesnya maka sebenarnya pengujian hanya dilakukan dengan metode *unit testing*. Table dibawah merupakan hasil pengujian dengan *unit testing*.



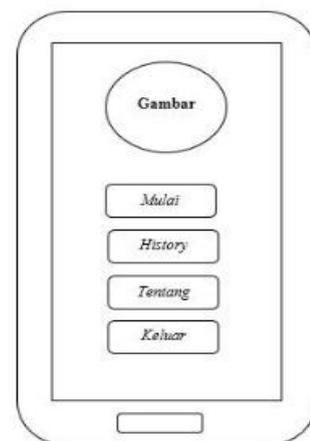
Gambar 2. Pohon Keputusan



Gambar 3. Use Case

Tabel.3 CRC Card

Class : Pertanyaan	
Responsibilities	Collaborators
Menampilkan Pertanyaan Menjawab Pertanyaan	Mesin Inferensi



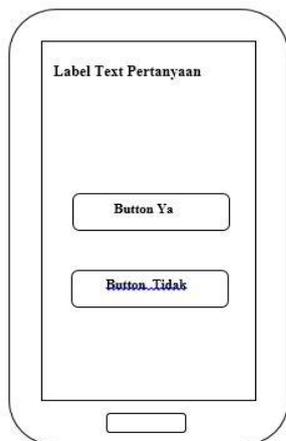
Gambar 4. Rancangan Tampilan Awal

2) Pengujian Fungsional

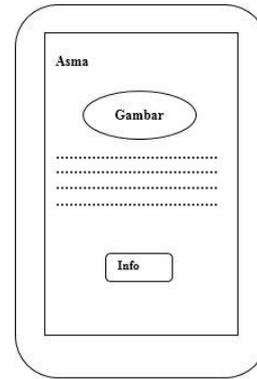
Pada tahap ini aplikasi akan diuji pada keadaan sebenarnya yaitu pada kondisi saat sistem sudah berjalan. Aplikasi dimulai dengan tampilan awal seperti pada gambar 7, terdapat 4 *button* menu yaitu “Mulai”, ”History”, “Tentang Aplikasi, dan *button* “Keluar”. selanjutnya pengguna menjawab pertanyaan sesuai gejala yang dialami dengan cara memilih salah satu *button* yang tersedia pada aplikasi seperti gambar 8. Mesin inferensi kemudian mengolah hasil input dari pengguna, kemudian menarik kesimpulan hasil analisa yaitu diagnosa penyakit anak. Hasil diagnosa akan muncul seperti pada gambar 9. Jika pengguna memilih *button* “info selengkapnya” maka aplikasi akan menampilkan informasi tentang penyakit dan pertolongan pertama seperti pada gambar 10. Terdapat juga *button* ”keluar” jika pengguna telah selesai menggunakan aplikasi.

Tabel 4. Hasil Pengujian dengan Unit Testing

No.	Nama Pengujian	Hasil Yang Diharapkan	Hasil
1.	Fungsi menu pada tampilan awal	Tiap menu dapat dijalankan tanpa error	Valid
2.	Proses analisa	Sistem mampu menampilkan hasil analisa berdasarkan input yang dimasukkan oleh user	Valid
3.	Menu info	Aplikasi menampilkan info penyakit berdasarkan hasil kesimpulan	Valid
4.	Menu pertolongan pertama	Aplikasi menampilkan solusi berdasarkan hasil kesimpulan	Valid



Gambar 5. Rancangan Pertanyaan



Gambar 6. Rancangan Kesimpulan



Gambar 7. Tampilan Awal



Gambar 8. Tampilan Pertanyaan



Gambar 9. Tampilan Kesimpulan



Gambar 10. Tampilan Info Penyakit

## V. KESIMPULAN DAN SARAN

### 1. Kesimpulan

Dengan ini aplikasi sistem pakar berbasis mobile dalam mendiagnosa penyakit anak telah selesai dibuat. Dengan menggunakan metode forward chaining, aplikasi ini mengolah input pengguna dengan aturan yang ada dalam basis pengetahuan sehingga menghasilkan kesimpulan yaitu hasil diagnosa penyakit anak. Pada aplikasi ini terdapat 9 penyakit dan 38 gejala yang didapat dari hasil wawancara dengan pakar.

Berdasarkan hasil pengujian semua fungsi pada aplikasi ini mendapatkan hasil yang valid. Dengan adanya aplikasi sistem pakar ini maka dapat lebih cepat mendapatkan penanganan pertama pada anak jika terserang penyakit. Namun demikian tidak sepenuhnya menggantikan peran pakar.

### 2. Saran

- Aplikasi ini dapat dikembangkan lagi dengan menambah fitur peta untuk mencari lokasi tempat praktek dokter spesialis terdekat.
- Ke depan aplikasi sistem pakar diagnosa penyakit anak dapat didesain dengan menggunakan metode backward chaining untuk data penyakit lebih kompleks

## DAFTAR PUSTAKA

- [1] Kusumadewi. 2003. Artificial Intelligent (Teknik dan aplikasinya). Edisi Pertama. Yogyakarta: Graha Ilmu.
- [2] Pressman, Roger S. 2010. Software Engineering : A Practitioner's Approach, Seventh Edition. Mc Graw Hill Higher Education
- [3] Safaat, Nazzarudin. 2011. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika. Bandung.
- [4] Turban, E. 2005. Decision Support System and Intelligence Systems. Yogyakarta: Penerbit Andi Yogyakarta.
- [5] Tyler, J. 2011. App Inventor for Android. Baker & Tayl, 2011.



Sekilas dari penulis dengan nama lengkap Christian Ramba Pasalli, lahir pada tanggal 11 Mei 1992 di Bitung. Dengan pendidikan pertama di Taman Kanak-Kanak Kristen Tabita, kemudian melanjutkan ke SD Negeri 20 Manado, melanjutkan ke SMP Negeri 1 Manado, dan pada tahun 2007 melanjutkan ke SMA Negeri 1 Manado.

Setelah lulus sekolah tingkat atas pada tahun 2010 penulis melanjutkan ke Perguruan Tinggi tepatnya di Universitas Sam Ratulangi Manado, Fakultas Teknik, Jurusan Elektro, Program Studi Teknik Informatika. Penulis membuat skripsi untuk memenuhi syarat sarjana (S1) dengan judul Sistem Pakar Diagnosa Penyakit Anak Menggunakan Metode Forward Chaining Berbasis Mobile yang di bimbing oleh dua dosen Teknik Informatika yaitu Dr.Eng. Vecky C. Poekoel, ST,MT dan Xaverius B. N. Najoan, ST, MT sehingga pada tanggal 29 Juni 2016 penulis resmi lulus di Teknik Informatika Universitas Sam Ratulangi Manado.