

Sistem Pakar Diagnosa Penyakit Menular Pada Anak Menggunakan Metode *Forward Chaining* dan *Backward Chaining*

Suci Hardianti¹⁾, Andi Tenriawaru²⁾, Natalis Ransi³⁾

^{1,2,3)}Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Halu Oleo, Kendari Kampus Hijau Bumi Tridharma, Anduonohu, Kec. Kambu, Kota Kendari, Sulawesi Tenggara 93232
Email: sucihardianti75@gmail.com¹⁾, atenriawaru36@gmail.com²⁾, Natalis.ransi@gmail.com³⁾

Abstrak - Sistem pakar merupakan pengetahuan seorang ahli yang dimasukkan kedalam sistem komputer agar dapat memecahkan masalah tertentu. Sebab terbatasnya pengetahuan dan informasi mengenai penyakit menular, khususnya yang menyerang anak-anak menjadi permasalahan umum yang didapatkan di masyarakat. Para orang tua kesulitan untuk memprediksi penyakit yang diderita oleh anak mereka, dan bingung untuk mengobatinya. Orang tua lebih memilih merawat anak mereka sendiri sesuai kemampuan mereka karena terkendala biaya yang bagi sebagian orang tua merupakan biaya yang cukup mahal. Hal itu sering kali terjadi, akibat keterlambatan penanganan menyebabkan penyakit tersebut bertambah parah dan sulit untuk disembuhkan. Tujuan dibuatnya sistem pakar diagnosa penyakit menular pada anak adalah untuk mempermudah orang tua dalam mendiagnosa penyakit yang dialami oleh anak dan dapat segera mendapatkan penanganan sesuai penyakit yang dialami oleh anak sehingga dapat Menghemat waktu dalam pengambilan keputusan. Bagi pakar diharapkan dapat menyimpan pengetahuan dan keahlian para pakar. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *waterfall* karena metode ini lebih mudah dipahami untuk perancangan sistem pakar. Sistem ini dibuat menggunakan bahasa pemrograman PHP dan MySQL sebagai *database*. Dari hasil pengujian menggunakan metode *blackbox testing* diperoleh sistem pakar diagnosa penyakit menular pada anak yang dibangun memiliki fungsionalitas yang baik dan berjalan sesuai dengan yang diharapkan.

Kata kunci - Sistem Pakar, *Forward Chaining*, *Backward Chaining*

1. PENDAHULUAN

Semakin meningkatnya kebutuhan serta permasalahan manusia sehingga peran komputer penting dalam menyelesaikan permasalahan manusia. Permasalahan umum yang biasa dirasakan oleh manusia salah satunya mengenai kesehatan. Sistem kekebalan tubuh atau sistem imun merupakan sistem yang bekerja untuk melindungi tubuh dari berbagai hal yang bisa menyebabkan tubuh mengalami sakit. Sistem kekebalan tubuh pada anak kecil masih belum terlalu kuat, sehingga lebih mudah terkena penyakit, terutama penyakit menular. Penyakit menular merupakan penyakit yang disebabkan oleh mikroorganisme, seperti virus, bakteri, parasit atau jamur dan dapat berpindah ke orang lain yang sehat. Terbatasnya pengetahuan dan informasi mengenai penyakit menular, khususnya yang menyerang anak-anak menjadi permasalahan umum yang didapatkan di masyarakat. Para orang tua kesulitan untuk memprediksi penyakit yang diderita oleh anak mereka dan bingung untuk mengatasinya. Orang tua lebih memilih mepercayakan diagnosis penyakit kepada pakar atau dokter yang ahli dibidangnya. Namun untuk melakukan pemeriksaan kepada pakar atau dokter memerlukan biaya, untuk sebagian orang tua, biaya untuk memeriksakan kondisi anak tidaklah murah sehingga banyak orang tua yang lebih memilih untuk merawat anak mereka sendiri. Akibat keterlambatan

penanganan terhadap penyakit menyebabkan penyakit tersebut bertambah parah dan sulit untuk disembuhkan.

Oleh karena itu, agar tidak ada kesalahan diagnosa serta mempermudah orang tua mengetahui secara dini penyakit yang diderita anak sehingga tidak terlambat mendapatkan pengobatan dikarenakan seorang dokter atau pakar memiliki keterbatasan waktu. Maka penulis tertarik membuat “sistem pakar diagnosa penyakit menular pada anak” yang dapat membantu dan mempermudah pasien dalam mendiagnosa penyakit. Dalam sistem pakar diperlukan suatu *Inference Mechanism* (mekanisme inferensi). Maka pada sistem pakar ini menggunakan algoritma dengan metode *forward chaining* dan *backward chaining*.

2. METODE

2.1 Sistem Pakar

Sistem pakar merupakan bagian dari kecerdasan buatan (*artificial intelligence*) yang terdiri dari pengetahuan dan pengalaman dari banyak pakar yang dimasukkan ke dalam suatu basis pengetahuan. Sistem pakar dapat membantu seseorang yang mungkin bukanlah seorang pakar untuk menyelesaikan persoalan tertentu[1].

2.2 Diagnosis

Diagnosis merupakan penentuan jenis penyakit dengan cara meneliti atau memeriksa gejala-gejalanya [2].

2.3 Penyakit Menular

Penyakit menular yang juga dikenal sebagai penyakit infeksi adalah sebuah penyakit yang disebabkan oleh sebuah agen biologi (seperti virus, bakteri atau parasit), bukan disebabkan faktor fisik (seperti luka bakar dan trauma benturan) atau kimia (seperti keracunan) yang bisa ditularkan atau menular kepada orang lain melalui media tertentu seperti udara, tempat makan dan minum yang kurang bersih pencuciannya, jarum suntik dan transfusi darah [3].

2.4 Inferensi

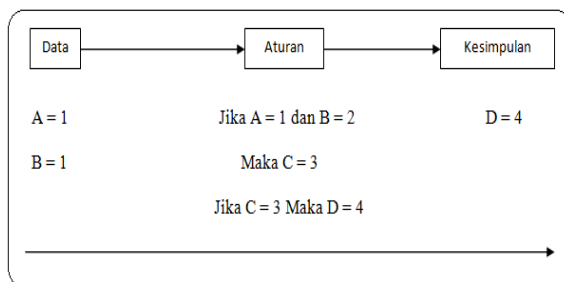
Inferensi merupakan suatu proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (logical conclusion) atau implikasi berdasarkan informasi yang tersedia[4].

Ada dua metode inferensi yang penting dalam sistem pakar, yaitu:

- 1). Runut maju (*forward chaining*)
- 2). Runut balik (*backward chaining*)

2.5 Forward chaining

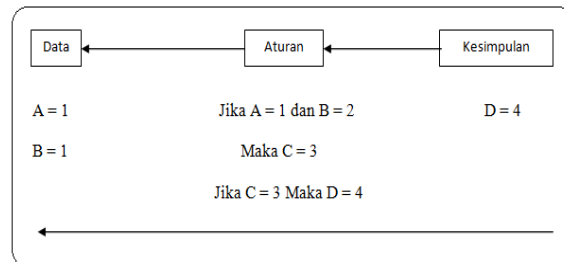
Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang-ulang sampai ditemukan hasil, gambar 1 menunjukkan bagaimana cara kerja metode inferensi runut maju.



Gambar 1 Alur Forward Chaining

2.6 Backward Chaining

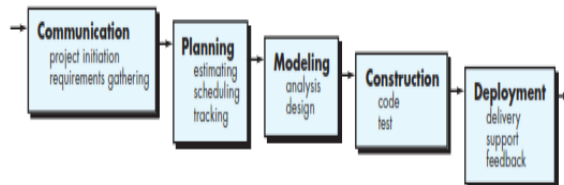
Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik, penalaran dimulai dengan tujuan menurut balik ke jalur yang akan mengarahkan ke tujuan tersebut. Gambar 2 menunjukkan bagaimana cara kerja metode inferensi runut balik.



Gambar 2 Alur Backward Chaining

2.7 Model Waterfall

Model *waterfall* adalah suatu proses pengembangan perangkat lunak, dimana kemajuan dipandang sebagai terus mengalir kebawah seperti air terjun. Umumnya dikenal dengan siklus hidup klasik, menyarankan pendekatan sistematis dan berurutan untuk pengembangan perangkat lunak yang dimulai dengan spesifikasi kebutuhan pelanggan dan kemajuan melalui perencanaan, pemodelan, konstruksi, dan penyebaran yang memuncak dalam dukungan yang berkelanjutan dari perangkat lunak [5]. Model waterfall dapat dilihat pada Gambar 3.



Gambar 3 Model Waterfall

2.8 Black Box Testing

Black Box Testing sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *black box* menitikberatkan pada fungsi sistem. Pengujian pada sistem menggunakan metode *Blackbox*, tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah benar menampilkan pesan-pesan kesalahan jika terjadi kesalahan dalam penginputan pesan. *Blackboxtesting* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Blackbox testing* memungkinkan pengembangan *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program [6].

3. HASIL DAN PEMBAHASAN

3.1 Metode Pengumpulan Data

Metode pengumpulan data yang dibutuhkan dalam penelitian ini ada dua yaitu data primer dan data sekunder sebagai berikut:

1). DataPrimer

Data yang dikumpulkan langsung dari tempat penelitian, melalui wawancara kepada pakar atau dokter spesialis anak dan kepada karyawan-karyawan yang dapat memberikan keterangan yang diperlukan. Wawancara dilakukan untuk memperoleh data dan informasi yang lebih lengkap dan benar dilakukan dengan mengadakan wawancara terhadap dokter spesialis anak yang mempunyai informasi dan wewenang untuk memberi data

2). Data Sekunder

Data yang didapat dan digunakan berupa pengetahuan teoritis yang didapat penulis oleh dokter spesialis anak langsung, mendapatkan referensi dari buku-buku yang relevan serta dari hasil penjelajahan (*browsing*) di *internet* yang berhubungan dengan penelitian ini.

3.2 Metode Pengembangan Sistem

Modeling

Pada proses *modeling* ini dilakukan penerjemahan syarat analisis kebutuhan ke sebuah perancangan perangkat lunak dan juga berfokus pada desain sistem. Berikut analisis kebutuhan sistem dan desain sistem.

- 1) Analisis kebutuhan sistem

Analisis kebutuhan dilakukan dengan observasi dan wawancara dengan pihak dokter spesialis anak di Rumah Sakit Umum Daerah Abunawas Kota Kendari yang berwenang memberikan data. Berdasarkan hasil wawancara dan observasi, maka dapat disimpulkan bahwa sistem yang akan dikembangkan memiliki kebutuhan sebagai berikut:

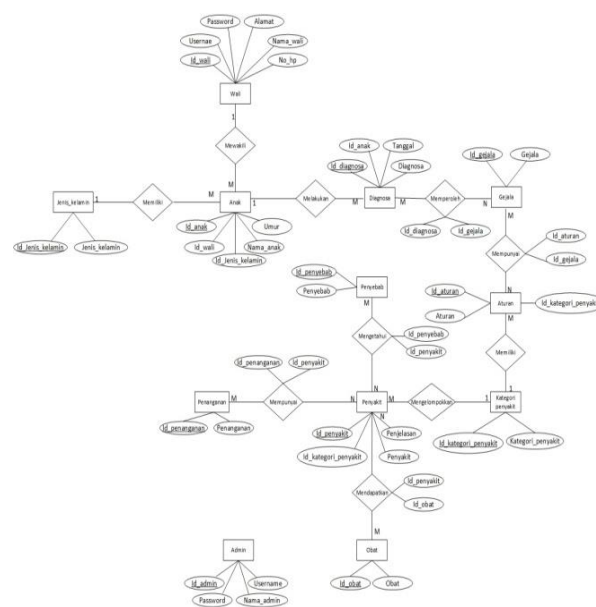
- (1). Data tentang jenis-jenis penyakit menular pada anak serta penjelasan, gejala, penanganan, penyebab dan obat.
- (2). Hasil rekamedik pasien.

2) Desain Sistem

Di tahap desain sistem, dilakukan perancangan sesuai spesifikasi kebutuhan sebelumnya, desain sistem membantu dalam menentukan perangkat keras (*hardware*), serta membantu dalam mendefinisikan arsitektur sistem secara keseluruhan. Perancangan suatu sistem membutuhkan adanya teknik-teknik penyusunan sistem untuk menganalisa dan mendokumentasikan data yang mengalir di dalam sistem tersebut. Pada tahap ini, terdapat perancangan *Entity Relationship Diagram* (ERD), normalisasi, perancangan desain *Unified Modeling Language* (UML) dan desain *Interface*.

(1) Desain *Entity Relationship Diagram (ERD)*

Desain ERD menggambarkan data atau aspek informasi dalam sistem yang akan diimplementasikan dalam basis data. Komponen utama dalam ERD yaitu *entity* dan hubungannya atau *relationship*. *Entity* merupakan objek fisik yaitu anak, wali, dan admin. Dalam entitas lainnya mencakup 6 jenis yaitu gejala, kategori penyakit, jenis penyakit, penyebab, obat penanganan diagnosa dan aturan. Desain ERD dari sistem pakar diagnosa penyakit menular pada anak menggunakan metode *forward chaining* dan *backwark chaining* pada Gambar 4.



Gambar 4 *Entity Relationship Diagram*

(2) *Bussines rules*

a. Wali

- Fungsi wali melakukan registrasi untuk dapat menggunakan sistem dalam mendiagnosa penyakit menular pada anak
- Fungsi wali memilih gejala yang dialami pasien yaitu anak
- Wali memilih kategori penyakit yang akan diproses oleh sistem sehingga mendapatkan penanganan yang sesuai jenis penyakit yang dialami.

b. Admin

- a) Admin bertugas menginput data wali dan anak agar dapat mengakses sistem

- b) Admin bertugas menginput data-data jenis penyakit menular yang akan dimasukkan ke dalam sistem
- c) Admin menginput data-data gejala yang dapat menimbulkan penyakit menular
- d) Menginputkan data jenis-jenis penanganan yang dapat digunakan dalam pengobatan penyakit menular.

(3) Aturan Desain *entity relationship diagram*

a. Identifikasi entitas dasar

Berdasarkan hasil wawancara dengan dr.Yeni Haryani,M.Kes.,Sp.A yang dilakukan di Rumah Sakit Abunawas Kendari diperoleh beberapa entitas yang bisa dimasukkan kedalam sistem, diantaranya entitas wali, anak, jenis kelamin, diagnosa, gejala, aturan, kategori penyakit, penyakit, penyebab, obat, dan entitas penanganan.

b. Identifikasi relasi

Setelah menentukan entitas-entitas dasar selanjutnya proses analisis relasi antar entitas, yaitu entitas wali memiliki relasi mewakili, entitas jenis kelamin memiliki relasi memiliki, entitas anak memiliki relasi melakukan, entitas diagnosa memiliki relasi memperoleh, entitas gejala memiliki relasi mempunyai, entitas aturan memiliki relasi memiliki, entitas kategori penyakit memiliki relasi mengelompokkan, entitas penyakit ke entitas penyebab diperoleh relasi mengetahui, entitas penyakit ke entitas penanganan memiliki relasi mempunyai dan entitas penyakit ke entitas obat memiliki relasi mendapatkan.

c. Menentukan nilai maksimal kardinalitas

Dari relasi yang dijabarkan, maka diperoleh kardinalitas setiap entitas. Pada entitas wali dengan entitas anak memiliki kardinalitas 1:M. Artinya satu wali bisa mewakili banyak anak. Entitas anak dengan entitas jenis kelamin memiliki kardinalitas M:1. Artinya banyak anak hanya dapat memiliki satu jenis kelamin. Entitas anak dengan entitas diagnosa memiliki kardinalitas 1:M. Artinya satu anak bisa melakukan banyak kemungkinan diagnosa. Pada entitas diagnosa dengan entitas gejala memiliki kardinalitas N:M. Artinya banyak diagnosa dapat memperoleh kemungkinan banyak gejala. Entitas gejala dengan entitas aturan memiliki kardinalitas N:M. Artinya banyak gejala dapat mempunyai banyak aturan. Entitas aturan dengan entitas kategori penyakit memiliki kardinalitas M:1. Artinya banyak aturan memiliki satu kategori penyakit. Entitas kategori penyakit dengan entitas penyakit memiliki kardinalitas 1:M. Artinya satu kategori penyakit dapat mengelompokkan banyak penyakit. Entitas penyakit dengan entitas penyebab memiliki

kardinalitas N:M. Artinya banyak penyakit dapat mengetahui banyak penyebab. Pada entitas penyakit dengan entitas penanganan memiliki nilai kardinalitas N:M. Artinya banyak penyakit mempunyai banyak kemungkinan penyebab. Dan pada entitas penyakit dengan entitas obat memiliki kardinalitas N:M. Artinya banyak penyakit mendapatkan kemungkinan banyak obat.

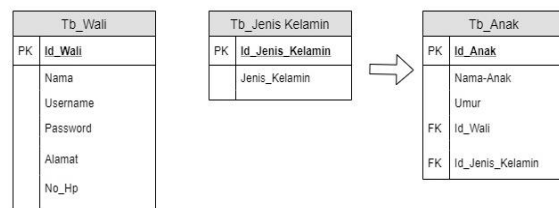
(4) Normalisasi

Berdasarkan *entity relationship diagram* yang telah dijabarkan sebelumnya, maka akan dilakukan normalisasi. Pada tahap ini normalisasi dilakukan untuk menghilangkan dan mengurangi redundansi atau duplikat data dan menghindari suatu proses update data pada suatu entity logik yang mengakibatkan perubahan pada lebih dari satu tempat dalam satu relasi (*problem update anomalies*) terhadap data yang disimpan dalam database sehingga terhindar dari *inkonsistensi* data atau tidak konsisten. Sebelum melakukan normalisasi tentunya ada tabel yang tidak normal, sebagai contoh penulis akan mengambil tabel anak seperti yang terlihat pada Gambar 5.

| Tb_Anak | |
|---------|----------------|
| PK | <u>Id_Anak</u> |
| | Nama_Anak |
| | Umur |
| FK | Nama_Wali |
| FK | Jenis_Kelamin |

Gambar 5 Tabel Anak Tidak Normal

Setelah menentukan tabel yang tidak normal selanjutnya dilakukan pengecekan normalisasi pada setiap tabel yang terbentuk. Sebagai tahap awal pengecekan normalisasi, pengecekan dilakukan pada tabel anak dan ditunjukkan pada diagram dependensi fungsionalnya.



Gambar 6 Tabel Anak Bentuk Normal

Diagram dependensi fungsional (DDF) tabel anak ditunjukkan pada Gambar 7.



Gambar 7 DDF Tabel Anak

Berdasarkan gambar 7 diperoleh bahwa tabel anak telah memenuhi 1NF karena setiap atribut memiliki nilai tunggal. Pada tabel anak diperoleh bahwa id_anak merupakan kunci. Setiap atribut yang bukan kunci yaitu id_jenis kelamin bergantung secara fungsional terhadap id_anak, sehingga dari tabel tersebut, tabel anak telah memenuhi 2NF. Selanjutnya dari DDF tabel anak, diketahui tidak terdapat dependensi fungsional transitif terhadap id_jenis_kelamin. Oleh karena itu tabel pengajuan servis telah memenuhi 3NF. Selain itu, dari DDF tabel anak diketahui bahwa yang merupakan determinan adalah id_anak yang merupakan kunci. Akibatnya, tabel pengajuan servis telah memenuhi BCNF.

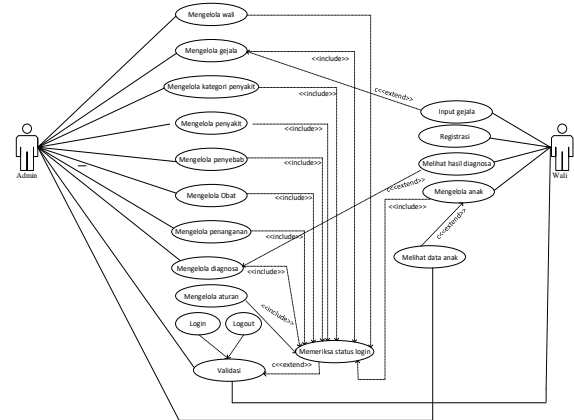
(5) Desain Unified Modeling Language (UML)

Berikut adalah perancangan desain UML Sistem Pakar Diagnosa Penyakit Menular Pada Anak Menggunakan Metode Forward Chaining dan Backward Chaining.

a. Use Case Diagram

Use case diagram menjelaskan antara aktor dengan sistem serta fungsi yang tersedia untuk setiap aktor. Pada Sistem Pakar Diagnosa Penyakit Menular Pada Anak Menggunakan Metode Forward Chaining dan Backward Chaining mempunyai dua aktor yaitu admin dengan wali. Aktor wali memiliki fungsi yaitu dapat mengelola wali, mengelola gejala, mengelola kategori penyakit, mengelola penyakit, mengelola penyebab, mengelola obat, mengelola penanganan, mengelola diagnosa, mengelola aturan, mengelola validasi dan juga dapat melihat data anak. Sedangkan aktor wali memiliki fungsi yaitu melakukan registrasi, menginput gejala, mengelola anak dan juga melihat hasil diagnosa. *Use case diagram* ditunjukkan pada Gambar 8.

Deskripsi mengenai aktor yang digunakan dalam *use case diagram* Sistem Pakar Diagnosa Penyakit Menular Pada Anak Menggunakan Metode Forward Chaining dan Backward Chaining dapat dilihat pada Tabel 1.



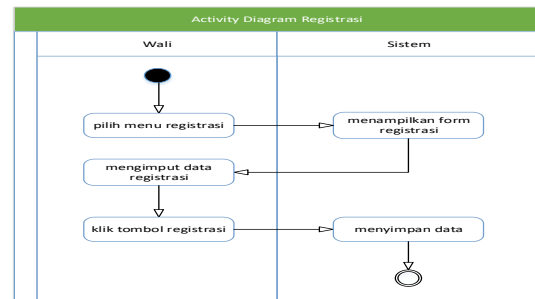
Gambar 8 Use Case Diagram

Tabel 1. Deskripsi Use Case Diagram

| No | Aktor | Deskripsi |
|----|-------|---|
| 1. | Wali | Orang tua pasien atau wali anak merupakan orang yang melakukan konsultasi masalah kesehatan untuk memperoleh pelayanan kesehatan yang diperlukan. Wali hanya bisa melihat informasi, melakukan konsultasi kesehatan dan mendapatkan jawaban dan solusi dari hasil konsultasi. |
| 2. | Admin | Admin merupakan aktor yang memiliki tugas untuk melakukan administrasi terhadap sistem, melakukan pemeliharaan sistem, memiliki kewenangan mengatur hak akses terhadap sistem, serta hal-hal lain berhubungan dengan pengaturturan operasional sistem |

b. Activity Diagram

Activity Diagram berikut menggambarkan aliran kerja Sistem Pakar yang akan dirancang. Ada beberapa *activity diagram* pada sistem pakar.



Gambar 9 Activity Diagram Registrasi

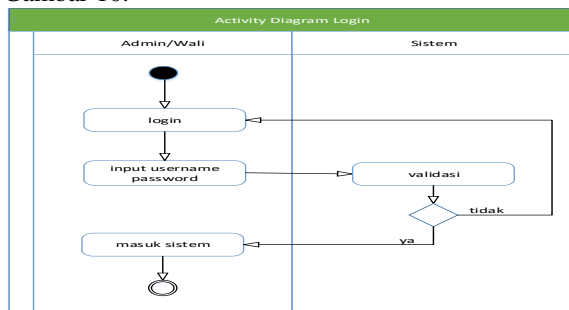
a). Activity Diagram Registrasi

Menggambarkan alur registrasi dari wali. Alur yang dilakukan oleh wali untuk registrasi adalah dengan memilih menu registrasi, kemudian sistem akan menampilkan form untuk registrasi, selanjutnya wali menginput data, adapun data-data

yang dimasukan oleh wali berupa data anak dan data diri wali. Setelah wali selesai memasukkan data kemudian wali mengklik tombol registrasi dan kemudian sistem menyimpan data yang telah dibuat oleh wali. *Activity diagram* registrasi dapat dilihat pada Gambar 9:

b). Activity Diagram Login

Menggambarkan alur kerja proses *login* dari *user* yaitu admin dan wali. Proses *login* dimulai dari node status awal kemudian admin wali memasukkan *username* dan *password* ke dalam *form login*. kemudian sistem akan melakukan pengecekan ke *database* berdasarkan *username* dan *password* yang dimasukkan oleh *user*. Jika valid maka admin dan wali akan masuk ke dalam sistem namun jika tidak valid maka akan dikembalikan ke halaman *login*. *Activity diagram login* dapat dilihat pada Gambar 10.



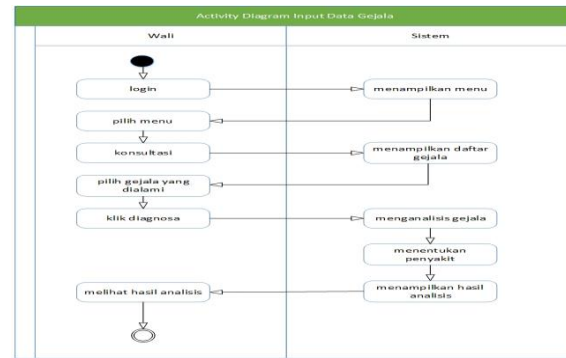
Gambar 10 Activity Diagram Login

c). Activity Diagram Input Data Gejala

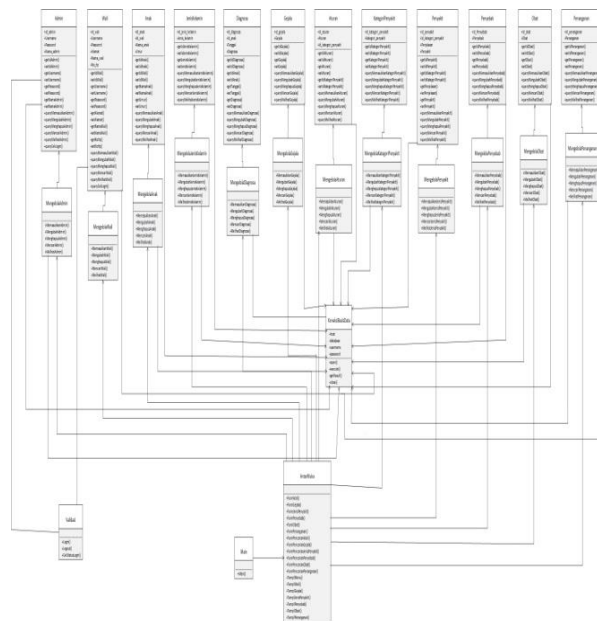
Menggambarkan alur kerja proses input data gejala dari anak. Setelah wali atau user login, sistem akan menampilkan menu. User atau wali masuk ke dalam menu dan memilih menu gejala. Setelah wali memilih menu gejala, sistem menampilkan daftar gejala selanjutnya wali menginput data gejala berdasarkan gejala yang dialami oleh anak. Setelah itu, makasistem akan melakukan diagnosa terhadap gejala yang dimasukkan. Dan wali dapat melihat hasil diagnosa. *Activity diagram* input data gejala dapat dilihat pada Gambar 11.

c. Diagram Kelas (Class Diagram)

Menggambarkan bagaimana struktur dari perancangan sistem. *Class Diagram* menunjukan bentuk validasi dalam pembuatan sistem. Masing-masing *class* memiliki atribut dan metode/fungsi sesuai dengan proses yang terjadi adapun bentuk *Class Diagram* dari “sistem pakar diagnosa penyakit menular pada anak dengan menggunakan metode *forward chaining* dan *backward chaining*” dapat dilihat pada Gambar 12.



Gambar 11 Activity Diagram Input Data Gejala



Gambar 12 Class Diagram

Class diagram menggambarkan struktur sistem dari Sistem Pakar yang akan dirancang. Pada Gambar 12 terdapat beberapa kelas, diantaranya kelas *main* sebagai fungsi awal eksekusi ketika sistem dijalankan, kelas *view* berupa kelas antarmuka, kelas yang diambil dari pendefinisian *use case* yaitu mengelola wali, mengelola anak, mengelola wali, mengelola jenis kelamin, mengelola diagnosa, mengelola gejala, mengelola aturan, mengelola kategori penyakit, mengelola penyakit, mengelola penyebab, mengelola obat, mengelola penanganan, mengelola admin dan validasi.

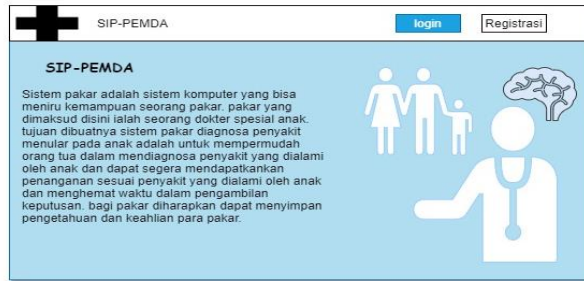
(6) Desain Antarmuka (Desain Interface)

Desain *Interface* yang dibuat meliputi desain *dashboard*, desain *login*, desain registrasi, desain konsultasi, desain hasil diagnosa, desain

daftar penyakit, desain daftar gejala, desain kategori penyakit, dan desain informasi pakar.

a. *Desain Interface Dashboard*

Merupakan halaman yang pertama kali ditampilkan ketika *user* mengakses *website*. Desain *interface dashboard* ditunjukkan pada Gambar 13.



Gambar 13 Desain *Interface Dashboard*

Pada Gambar 13 Desain *Interface Dashboard* merupakan tampilan utama yang ditampilkan ketika pengguna mengakses sistem pakar ini. Pada tampilan *dashboard* hanya ditampilkan menu *dashboard* yang berisi kata-kata mengenai sistem pakar, menu yang terdapat pada *dashboard* yaitu *login* dan *registrasi*.

b. *Desain interface halaman login*

Merupakan halaman setelah *user* berhasil masuk kedalam sistem. Desain *interface* menu *login* yang ditunjukkan pada Gambar 14.



Gambar 14 Desain *Interface Menu Login*

Pada Gambar 14 terdapat menu *login* yang memiliki *form* *username* dan *password*, dalam sistem pakar diagnosa penyakit menular pada anak menu *login* terdiri dari beberapa hak akses yaitu sebagai admin dan user.

c. *Desain Interface Registrasi*

Merupakan tampilan yang ditampilkan ketika user mengakses halaman registrasi. Desain *interface* registrasi dapat dilihat pada Gambar 15.

Halaman ini digunakan untuk melakukan registrasi pasien yang berisi *form* isian biodata wali dan biodata anak, kontak dan akun yang akan digunakan sebagai akun aktif untuk mengakses sistem pakar. Halaman registrasi berlaku bagi semua

user/pasien saat pertamakali mengakses sistem pakar ini. Data yang di isi harus sesuai dengan dokumen asli atau data yang di miliki wali dan anak.

Gambar 15 Desain *Interface Registrasi*

d. *Desain Interface Menu Konsultasi*

Merupakan tampilan yang ditampilkan ketika user telah melakukan registrasi dan memilih menu konsultasi. Desain *Interface* Menu konsultasi dapat dilihat pada Gambar 16.

Gambar 16 Desain *Interface Menu Konsultasi*

Pada Gambar 16 desain *interface* menu konsultasi terdapat *form* pilihan gejala-gejala. Pada halaman ini, user dalam hal ini wali di perintahkan untuk mengisi *form* pilihan gejala yang di alami anak untuk melakukan diagnosa.

e. *Desain Interface Menu Hasil Diagnosa*

Merupakan halaman setelah *user* berhasil memasukkan gejala yang di alami anak. Desain *interface* menu hasil diagnosa yang ditunjukkan pada Gambar 17.

Gambar 17 Desain Interface Menu Hasil Diagnosa

Pada Gambar 17 terdapat list gejala pilihan pasien kemudian menentukan kategori penyakit berdasarkan gejala, hasil diagnosa penyakit anak berupa nama penyakit, penjelasan tentang penyakit, penyebab, penanganan dan obat yang sesuai penyakit yang dialami anak.

f. Desain Interface Menu Daftar Penyakit

Merupakan halaman ketika user/admin membuka master data dan memilih menu daftar penyakit. Desain *interface* menu daftar penyakit yang ditunjukkan pada Gambar 18.

Gambar 18 Desain Interface Menu Daftar Penyakit

Pada halaman ini terdapat semua nama-nama penyakit yang ada dalam sistem pakar ini beserta menu tambah penyakit, menu penjelasan, edit dan hapus. Desain yang di tampilkan merupakan tampilan khusus admin dan pada tampilan untuk user hanya terdapat menu penjelasan maksudnya tidak dapat menambah penyakit, mengedit dan hapus.

g. Desain Interface Menu Daftar Gejala

Merupakan halaman ketika user/admin membuka master data dan memilih menu daftar gejala. Desain *interface* menu daftar gejala yang ditunjukkan pada Gambar 19.

Gambar 19 Desain Interface Menu Daftar Gejala

Pada Gambar 19 terdapat semua gejala yang ada dalam sistem pakar ini beserta menu tambah gejala, menu edit dan hapus. Desain yang di tampilkan merupakan tampilan khusus admin dan pada tampilan untuk user hanya terdapat gejala-gejala dan tidak terdapat menu tambah gejala, edit dan hapus. Maksudnya user hanya dapat melihat gejala-gejala namun tidak dapat menambah, mengedit dan menghapus gejala.

h. Desain Interface Menu Kategori Penyakit

Merupakan halaman ketika user/admin membuka master data dan memilih menu kategori penyakit. Desain *interface* menu kategori penyakit yang ditunjukkan pada Gambar 20.

Gambar 20 Desain Interface Menu Kategori Penyakit







Pada Halaman ini terdapat semua kategori penyakit yang ada dalam sistem pakar ini, beserta menu tambah kategori penyakit, edit dan hapus. Desain yang di tampilkan merupakan tampilan khusus admin dan pada tampilan untuk user hanya terdapat beberapa kategori penyakit dan tidak terdapat menu tambah kategori penyakit, edit dan

hapus. Maksudnya user hanya dapat melihat kategori penyakit namun tidak dapat menambah, mengedit dan menghapus.

- i. Desain *Interface* Menu Informasi Pakar

Merupakan halaman ketika *user* memilih menu informasi pakar. Desain *interface* menu informasi pakar yang ditunjukkan pada Gambar 21.

Pada Gambar 21 terdapat biodata diri, riwayat pendidikan dan tempat praktek dokter yang menjadi pakar dalam penelitian ini.

| SIP-PEMDA | | LOGOUT |
|---|---|--|
|  | Administrator | <div>Informasi Pakar</div> <div> <p>Nama : dr. Yeni Haryani M.Kes., Sp. A</p> <p>Tempat/Tanggal Lahir: Bulukumba 17 Oktober 1981</p> <p>Agama: Islam</p> <p>Pendidikan:</p> <ol style="list-style-type: none"> SD - Bulukumba SMP: Bulukumba SMA: Makassar S1: UNHAS 2005 S2: UNHAS 2013 <p>Tempat Praktik:</p> <ol style="list-style-type: none"> RSUD Abunawas Kota Kendari </div> |
|  | Konsultasi | |
| <input type="radio"/> | Master Data  | |
|  | Daftar Penyakit | |
|  | Daftar Gejala | |
|  | Kategori Penyakit | |
|  | Informasi Pakar | |

Gambar 21 Desain *Interface* Menu Informasi Pakar

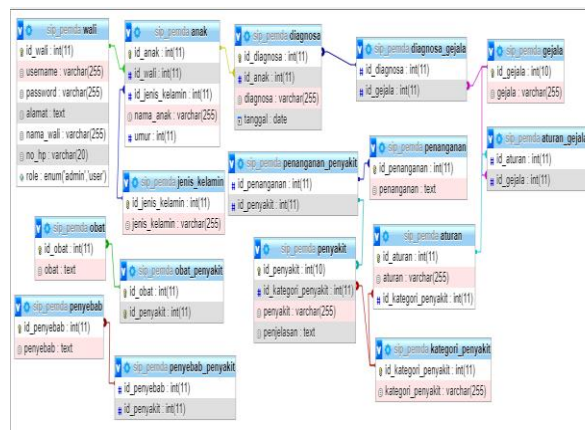
Constructions

Construction merupakan proses membuat *code*. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh *user*. Tahap ini yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Proses pengujian pada penelitian ini menggunakan *Blackbox Testing*. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

Deployment

Tahap ini bisa dikatakan tahap *final* dalam pembuatan *software* atau sistem. Setelah melakukan analisis, desain, pengkodean dan *testing* maka sistem yang sudah jadi akan diimplementasikan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan untuk melihat sejauh mana kehandalan sistem setelah diterapkan, proses pemeliharaan akan dicek pula apakah sistem sudah bekerja sesuai dengan fungsinya.

3.3 Implementasi Basis Data



Gambar 22 Implementasi Basis Data

Pengembangan sistem pakar diagnosa penyakit menular pada anak menggunakan DBMS (*Database Management System*). Di dalam basis data pengembangan sistem ini, terdapat enam belas tabel yang saling berelasi yaitu: tabel wali, anak, jenis kelamin, diagnosa, diagnosa gejala, gejala, aturan gejala, aturan, kategori penyakit, penanganan, penanganan penyakit, penyakit, obat penyakit, obat, penyebab penyakit, dan tabel penyebab. Implementasi basis data sistem pakar diagnosa penyakit menular pada anak menggunakan metode *forward chaining* dan *backward chaining* dapat dilihat pada Gambar 22.

3.4 Pengujian (*Testing*)

| Kasus dan hasil uji benar (data benar) | | | | |
|--|--|---|---|------------|
| No | Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Admin atau wali mengisi <i>form</i> biodata wali yaitu <i>username</i> dan <i>password</i> untuk mendaftarkan akun serta biodata anak lalu menekan tombol register | Jika biodata wali, <i>username</i> , <i>password</i> dan biodata anak valid, maka wali akan diarahkan ke halaman <i>login</i> | Data registrasi valid, sistem mengarahkan ke halaman <i>login</i> | Berhasil |
| Kasus dan hasil uji kesalahan (data salah) | | | | |
| No | Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1. | Admin atau wali mengisi data registrasi dengan memasukkan data tidak lengkap atau tidak mengisi salah satu <i>form</i> registrasi | Jika data yang dimasukkan tidak lengkap atau tidak mengisi salah satu <i>form</i> registrasi, maka sistem akan menampilkan informasi untuk mengisi <i>form</i> yang belum diisi | Sistem menampilkan pesan kesalahan yaitu " <i>please fill out this field</i> " yang artinya harap isi bagian yang belum terisi. | Berhasil |

Gambar 23 Pengujian Halaman Registrasi

Hasil pengujian sistem menggunakan metode *black box testing* yang terdiri dari pengujian halaman registrasi, *login* dan konsultasi. Hasil pengujian dapat ditunjukkan pada Gambar 23 hingga Gambar 25.

| Kasus dan hasil uji benar (data benar) | | | | |
|--|---|--|---|------------|
| No | Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Admin atau wali mengisi form login dengan memasukkan username dan password dan menekan tombol login | Jika data login valid, maka admin atau wali akan diarahkan ke halaman dashboard | Data login valid, sistem mengarahkan ke halaman dashboard | Berhasil |
| Kasus dan hasil uji kesalahan (data salah) | | | | |
| No | Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Admin atau wali mengisi data login dengan memasukkan username dan password salah atau belum terdaftar | Jika data yang dimasukkan salah maka sistem akan menampilkan pesan kesalahan dan data tidak diproses | Sistem menampilkan pesan kesalahan yaitu username/ password salah, login gagal. | Berhasil |

Gambar 24 Pengujian Halaman Login

| Kasus dan hasil uji benar (data benar) | | | | |
|--|---|---|---|------------|
| No | Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Admin atau wali memasukkan gejala yang di alami oleh anak lalu mengklik tombol diagnosa | Jika gejala yang di inputkan sesuai, maka akan di arahkan ke halaman hasil diagnosa | Gejala valid maka sistem menampilkan hasil diagnosa berdasarkan gejala yang dialami anak. | Berhasil |
| Kasus dan hasil uji kesalahan (data salah) | | | | |
| No | Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Admin atau wali memasukkan gejala hanya satu atau dua lalu mengklik tombol diagnosa | Jika gejala yang di input hanya satu atau dua maka menampilkan pesan kesalahan. | Sistem menampilkan pesan kesalahan yaitu "Gejala yang anda pilih kurang dari 3" | Berhasil |

Gambar 25 Pengujian Halaman Konsultasi

- 1). Sistem pakar ini menggunakan metode *forward chaining* dan *backward chaining* sebaiknya diperbanyak lagi jenis penyakitnya sehingga sistem mampu untuk menentukan banyak jenis penyakit dan solusi
- 2). Sistem pakar ini perlu dibuat versi android agar lebih mudah dalam melakukan konsultasi sehingga mempercepat proses diagnosa.

REFERENSI

- [1] Solichin.A. 2011. Sistem Pakar Berbasis Mobile Untuk Mendeteksi Penyakit Pada Ginjal. 4(3), 249-254.
- [2] Diagnosis. (Def.1) (n.d). Dalam Kamus Besar Bahasa Indonesia (KBBI) Online. Diakses Melalui https://kbbi.web.id/diagnosa_diagnosis, 20 Desember 2020
- [3] Khomsah. 2007, Desember. Penyakit Menular dan Penyakit tidak menular. Diakses 2 April, 2011, dari Web Site Penyakit dan Pengobatannya: <http://www.infopenyakit.com/2007/12/penyakit-menular-dan-tidak-menular.html>
- [4] Kusri. 2006. Sistem Pakar Teori dan Aplikasi, Yogyakarta: Andi Yogyakarta
- [5] Pressman. R. S. MAXIM, B. R. (2015). *Software Engineering A PRACTITIONER'S APPROACH* (V. Bradshaw (ed.); Eight). McGraw-Hill Education.
- [6] Hanifah.U, Alit.R, Sugiarto. Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk. SCAN VOL. XI NOMOR 2 JUNI 2016

4. KESIMPULAN

Kesimpulan dari penelitian pembuatan sistem pakar diagnosa penyakit menular pada anak menggunakan metode *forward chaining* dan *backward chaining* adalah sebagai berikut :

- 1). Dalam penelitian sistem pakar ini menggunakan bahasa pemrograman HTML, CSS, Javascript dan PHP. Sistem ini juga menggunakan *Mozilla Firefox* atau *Google Chrome* sebagai media untuk membuka sistem dan MySQL sebagai penyimpanan data.
- 2). Sistem pakar ini telah diuji menggunakan metode pengujian *Black-box* dan hasil pengujian menunjukkan bahwa sistem pakar diagnosa penyakit menular pada anak menggunakan metode *Forward Chaining* dan *Backward Chaining* dapat berjalan dengan normal dan bekerja sesuai dengan rancangan, hal ini ditandai dengan respon sistem yang sudah tepat sesuai dengan input yang diberikan. Dengan hasil tersebut maka dapat dikatakan bahwa tujuan pembuatan sistem ini sudah tercapai.

Berdasarkan pembuatan sistem pakar yang dibuat, maka disarankan sebagai berikut :