Sistem Pakar Untuk Diagnosa Penyakit Vertigo Dengan Metode *Forward Chaining* (Studi Kasus: Klinik Bhayangkara Panaikang)

Erny Marlina*[1], Nur Salman Halim[2], Rudy Donny[3]
[1,2,3]Sistem Informasi STMIK Dipanegara Makassar
E-mail: *[1]ernikadang@dipanegara.ac.id, [2]nursalman.halim@dipanegara.ac.id,
[3]rudy.donny@dipanegara.ac.id

Abstrak

Klinik Bhayangkara Panaikang merupakan sebuah poliklinik umum yang berada dibawah naungan Rumah Sakit Bhayangkara. Pelayanan yang diberikan di klinik ini merupakan pelayanan untuk umum. Salah satu penyakit yang sering menjadi keluhan pasien adalah penyakit vertigo. Penyakit vertigo merupakan jenis penyakit yang menyerang pada bagian kepala. Gejala vertigo adalah mengalami muntah dan rasa mual dan pusing serta telinga yang berdengung. Untuk memudahkan pasien dalam mengenali penyakit vertigo maka dirancang sebuah sistem pakar sebagai alat bantu untuk mendiagnosa penyakit vertigo dengan. Sistem pakar merupakan sistem berbasis komputer yang mengandung pengetahuan dari satu atau lebih pakar atau manusia. Algoritma sistem pakar yang dipakai adalah Forward Chaining dan menggunakan metode pengujian Black Box. Penelitian ini akan menghasilkan pilihan gejala yang dapat dipilih oleh pasien, dimana setiap pilihan gejala yang dipilih pasien akan mendapatkan hasil akhir, sistem akan menampilkan pilihan gejala pasien, dan penyakit yang diderita. Adapun tujuan dari penelitian ini adalah untuk memberikan hasil berupa diagnosa penyakit yang dialami, anjuran pengobatan, dan terapi pengobatan dalam menjawab pertanyaan selama sesi konsultasi ketika menggunakan sistem ini.

Kata Kunci*—Klinik Bhayangkara, Sistem Pakar, Forward Chaining*

Abstract

Bhayangkara Panaikang clinic is a general polyclinic under of Bhayangkara Hospital. Servants given at this clinic are public services. One of diseases that often become a complaint of patients is vertigo disease. Vertigo disease is a type of disease that attacks the head. Symptoms of vertigo are experiencing vomiting and dizziness and buzzing ears. To facilitate patients in recognizing vertigo then in designing a expert system as tool to diagnose vertigo disease. Expert system are computer based system that contain knowledge of one or more expert humans. Expert system algorithm in used is forward chaining and using black box system testing method. This research will generate a selection of the symptons selected by the user who will get the final result. As for the purpose of this research is to give result of diagnosis of disease which experience treatment suggestion and treatment therapy inaswering question during the consultation session when using this system..

Keywords*—Bhayangkara Clinic, Expert System, Forward Chaining*

1. Pendahuluan

Teknologi semakin berkembang seiring dengan meningkatnya kebutuhan setiap individu di berbagai bidang, seperti di bidang bisnis, pendidikan, psikologi, dan tentu saja dibidang kesehatan [1]. Dengan perkembangan teknologi saat ini, semua orang dapat berkomunikasi serta memperoleh dan menyampaikan berbagai informasi yang dibutuhkan.

Kesehatan merupakan hal yang sangat penting dalam kehidupan. Namun terkadang kesadaran akan pola hidup yang sehat dan pola makan yang teratur masih terabaikan. Banyak penyakit dan gejala-gejala yang timbul akibat pola hidup dan pola makan yang tidak teratur, salah satunya adalah vertigo. Vertigo merupakan sakit kepala yang berbeda dari gejala sakit kepala biasa dan migren [2]. Vertigo adalah suatu sensasi gerakan atau rasa gerak dari tubuh atau lingkungan sekitarnya dengan gejala yang timbul, terutama dari jaringan otonomik yang disebabkan oleh gangguan alat keseimbangan tubuh yang ditandai dengan perasaan berputar, dunia serasa bergoyang, benda sekeliling berputar, rasa mau jatuh bahkan adakalanya benar-benar terjatuh, disertai dengan mual, muntah, keringat dingin serta merasa lebih baik jika berbaring.

Terkadang seorang penderita vertigo itu sendiri pada awalnya tidak mengetahui jenis gejala maupun penyakit yang dideritanya karena minimnya informasi yang mereka ketahui. Bila seorang penderita vertigo ingin mengetahui tentang gejala-gejala, penyebab serta cara penanggulangan yang baik maka mereka akan mendatangi dokter di klinik untuk berkonsultasi. Akan tetapi, hal tersebut tidak dapat dilakukan oleh semua orang mungkin karena faktor perekonomian yang kurang mencukupi ataupun karena tuntutan kesibukan dan aktifitas mereka yang padat sehingga tidak bisa menyesuaikan dengan jadwal konsultasi praktek dokter di klinik. Terkadang ketika berkonsultasi pun informasi hasil diagnosa yang disampaikan oleh dokter tidak sepenuhnya dipaparkan secara mendetail [2].

Klinik Bhayangkara merupakan sebuah poliklinik umum yang berada dibawah naungan Rumah Sakit Bhayangkara yang memberikan pelayanan tidak hanya kepada anggota Kepolisian saja tetapi juga pelayanan terbuka untuk masyarakat umum. Banyaknya masyarakat yang datang dengan gejala yang mirip dengan vertigo dan kurangnya informasi mengenai penyakit vertigo serta kurangnya tenaga medis menimbulkan permasalahan seperti banyaknya masyarakat yang harus menunggu waktu untuk berkonsultasi dengan tenaga medis berkaitan dengan penyakit vertigo, terkadang pasien penyakit vertigo awalnya tidak mengetahui penyakit yang dideritanya karena kurangnya informasi dan tidak semua penderita vertigo dapat berkonsultasi dengan dokter karena faktor ekonomi yang kurang atau karena aktivitas dan kesibukan masyarakat yang padat.

Salah satu solusi yang dapat digunakan oleh Klinik Bhayangkara Makassar adalah menggunakan menggunakan sistem pakar untuk mendiagnosa penyakit vertigo dengan algoritma *Forward Chaining*. *Forward Chaining* adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. Metode ini cocok digunakan untuk memcahkan masalah diagnosis dan merupakan cara yang efesien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur[3].

2. Metode Penelitian

2.1 Sistem Pakar (*Expert Sistem)*

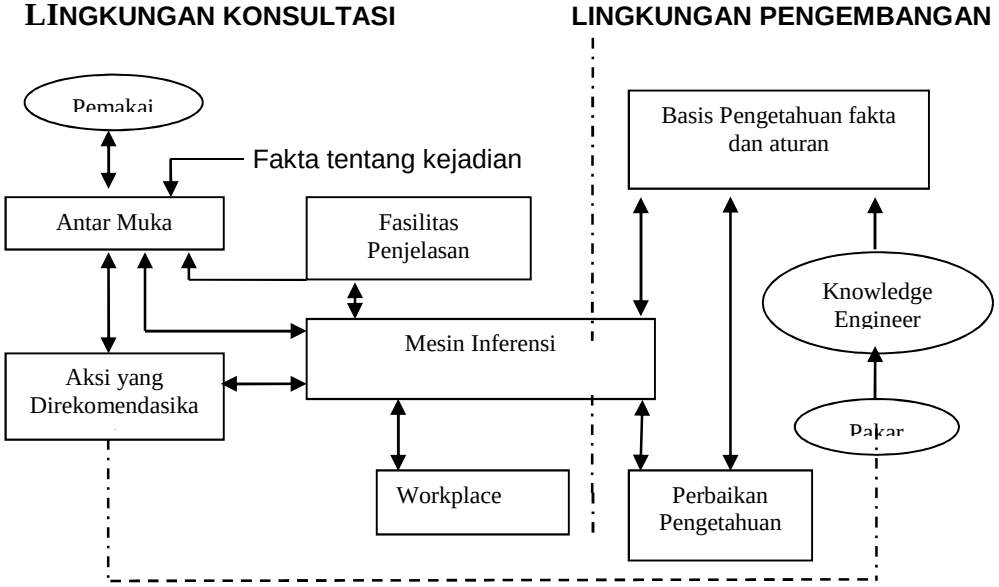
Sistem pakar (*expert system*) adalah Sistem yang dirancang untuk jenis – jenis aktivitas yang hanya mungkin dilakukan oleh pakar. Atau dengan kata lain sistem pakar adalah sistem yang didesain dan diimplementasikan dengan bantuan bahasa pemrograman tertentu untuk dapat menyelesaikan masalah seperti yang dilakukan para ahli. Diharapkan sistem ini, orang awam dapat menyelesaikan masalah tertentu baik sedikit rumit ataupun rumit sekalipun tanpa bantuan para ahli dalam bidang tersebut. Sedangkan bagi para ahli, sistem ini dapat digunakan sebagai asisten yang berpengalaman[4].

Secara umum dapat dikatakan baha sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Dengan sistem pakar orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan seorang ahli dibidangnya.

Selain itu sistem pakar juga dirancang untuk membantu aktivitas para pakar sebagai asisten yang berpengalaman dan memiliki kemampuan yang dibutuhkan. Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya dipergunakan dalam penyelasian masalah tertentu[4].

Sebuah sistem pakar yang baik harus memenuhi beberapa syarat yaitu: memiliki informasi yang handal, mudah dimodifikasi, dapat dipergunakan dalam berbagai jenis komputer dan memiliki kemampuan untuk belajar beradaptasi.

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment)* dan lingkungan konsultasi (*consulation environment)* lingkungan pengembangan pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar dalam memperoleh pengetahuan pakar. Komponen-komponen tersebut dapat digambarkan sebagai berikut:



Gambar 1. Arsitektur Sistem Pakar

Antarmuka pengguna merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya

kedalam bentuk yang dapat diterima oleh sistem. Selain antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai. Pada *user interface* terjadi dialog antara program dan pemakai, yang memungkinkan sistem pakar menerima instruksi dan informasi (*input*) dari pemakai, juga memberikan informasi (*output)* pada pemakai[4].

2.2 *Keuntungan Sistem Pakar*

Sistem Pakar (*expert system*) merupakan paket perangkat lunak atau aket program komputer yang ditujukan sebagai penyedia nasihat dan saran bantu dalam memecahkan masalah di bidang-bidang spesialisasi tertentu seperti sains, perekayasaan, matematika, kedokteran, pendidikan dan sebagainya. Sistem pakar merupakan subset dari *Artificial Intelegence*[4]. Ada beberapa keunggulan sistem pakar, yang di antaranya dapat :

a. Menghimpun data dalam jumlah yang sangat besar.
b. Menyimpan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu.
c. Menyimpan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu.
d. Mengerjakan perhitungan secara cepat dan tepat dan tanpa jenuh mencari kembali data yang tersimpan dengan kecepatan tinggi.

Sementara kemampuan sistem pakar[5] di antaranya adalah :

a. Menjawab berbagai pertanyaan yang menyangkut bidang keahliannya.
b. Bila diperlukan dapat menyajikan asumsi dan alur penalaran yang digunakan untuk sampai ke jawaban yang dikehendaki
c. Menambah fakta kaidah dan alur penalaran sahih yang baru ke dalam otaknya.

Bentuk-bentuk tersebut memungkinkan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik dari seorang yang bukan ahli.Seorang ahli adalah seorang yang mempunyai pengetahuan tertentu dan mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topic permasalahan (domain), menyusun kembali pengetahuan jika dipandang perlu, memilah aturan jika dibutuhkan, dan menentukan relevan atau tidaknya keahlian mereka. Pengalihan keahlian dari para ahli untuk kemudian dialihkan ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar[5].

Proses ini membutuhkan empat aktivitas, yaitu tambahan pengetahuan (dari para ahli atau sumber-sumber lainnya), representasi pengetahuan (ke komputer), inferensi pengetahuan, dan pengalihan pengetahuan ke pengguna. Pengetahuan yang disimpan di komputer dinamakan dengan nama basis pengetahuan (*knowledge base)*[5].

2.3 Vertigo

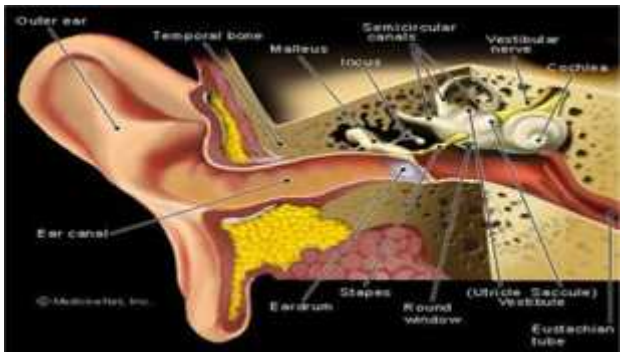
Vertigo berasal dari bahasa Latin “verto” = gerakan berputar adalah salah satu bentuk sakit kepala dimana penderita mengalami persepsi gerakan yang tidak semestinya. Biasanya gerakan berputar atau melayang ini disebabkan oleh gangguan pada sistem *vestibula.* Vertigo seringkali diawali dengan gejala mual dan muntah serta ketidakmampuan penderita menjaga keseimbangan badan, yang menyebabkan penderita mengalami kesulitan berdiri atau berjalan[6]. Rasa pusing pada vertigo disebabkan oleh gangguan alat keseimbangan tubuh yang mengakibatkan ketidakcocokan antara posisi tubuh yang sebenarnya dengan apa yang dipersepsikan oleh susunan saraf pusat.

Vertigo bukan terjadi karena faktor keturunan. Vertigo tidak boleh disepelekan karena bisa menjadi pertanda penyakit yang serius, seperti tumor otak, *hipertensi* (tekanan darah tinggi), *diabetes mellitus* (kencing manis), jantung, dan ginjal. Semakin dini vertigo ditangani akan semakin cepat dapat diatasi. Vertigo bisa terjadi karena :

1. Infeksi virus yang menyerang area *labirin*
2. Infeksi bakteri yang mengenai telinga tengah

3. Radang pada leher
4. Serangan migren
5. Sirkulasi darah yang berkurang yang dapat menyebabkan aliran darah ke otak berkurang
2. Mabuk kendaraan
3. Alkohol
4. Kelainan *Neurologis*
   a. *Sklerosis multiple*
   b. Patah tulang tengkorak yang disertai cedera pada *labirin*, persarafannya atau keduanya
   c. Tumor otak
   d. Tumor yang menekan saraf *vestibularis*.

Vertigo biasanya muncul karena adanya gangguan sistem *vestibular* (misalnya terdapat gangguan pada struktur telinga bagian dalam, saraf *vestibular*, batang otak, dan otak kecil/*cerebellum*). Sistem *vestibular* bertanggung jawab untuk mengintegrasikan rangsangan terhadap indera dan gerakan tubuh. Ketika kepala bergerak, sinyal ditransmisikan ke *labirin*, yang terdapat di telinga bagian dalam. *Labirin* kemudian membawa informasi ke saraf *vestibular* yang kemudian diteruskan ke batang otak dan otak kecil, yang berfungsi mengontrol keseimbangan, postur, dan kordinasi gerak.



Gambar 2. Gangguan Sistem *Vestibular*

2.4 Metode *Forward Chaining*

Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju.

Berikut adalah diagram *Forward Chaining* secara umum untuk menghasilkan sebuah *goal*. *Forward chaining* merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai *TRUE*), maka proses akan menyatakan konklusi. Forward chaining adalah *data-driven* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan *tree* yang lebar dan tidak dalam, maka gunakan *forward chaining* [7].
Tipe sistem yang dapat dicari dengan *Forward Chaining* :

1. Sistem yang dipersentasikan dengan satu atau beberapa kondisi.

2. Untuk setiap kondisi, sistem mecari *rule-rule* dalam *knowledge base* untuk *rule-rule* yang berkorespondensi dengan kondisi dalam bagian IF
3. Setiap *rule* dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian THEN. Kondisi baru ini ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi baru dari konklusi yang diminta, sistem akan kembali ke langkah 2 dan mencari *rule-rule* dalam *knowledge base* kembali. Jika tidak ada konklusi baru, sesi ini berakhir.

Contoh :

Terdapat 10 aturan yang tersimpan dalam basis pengetahuan yaitu :

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

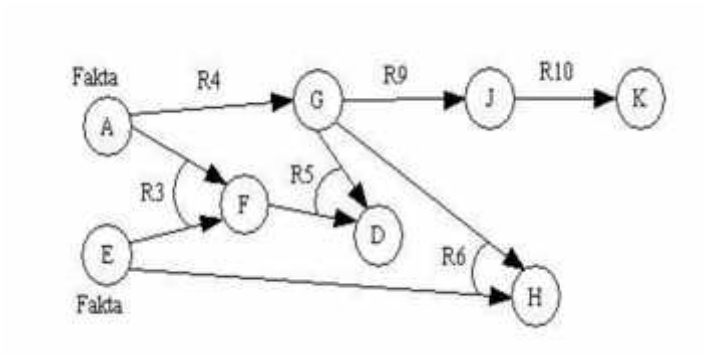
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar.
Proses penalaran forward chaining terlihat pada gambar 1 berikut ini :



Gambar 3. Penalaran *Forward Chaining*

Alasan penggunaan *forward channing* dalam proses pelacakan premis untuk menemukan solusi, karena sangat cocok digunakan untuk proses diagnosis, karena memiliki cara kerja yang mirip dengan seorang pakar, yaitu dimulai dengan pengumpulan data gelaja-gelaja terlebih dahulu menuju ke suatu konklusi. Hal ini juga dilakukan seorang pakar, yaitu melakukan pemeriksaan terlebih dahulu kemudian menyimpulkan hasilnya untuk pengambilan keputusan[7].

Adapun cara kerja *forward chaining* adalah sebagai berikut:

1. Sistem dipresentasikan dengan suatu atau lebih dari kondisi.

2. Untuk setiap kondisi system akan mencaru *rule* pada *knowledgebase* untuk *rule* tersebut yang cocok dengan kondisi IF.
3. Setiap *rule* dapat merubah suatu kondisi baru dari konklusi dari bagian THEN. Kondisi baru ini selanjutnya akan ditambahkan. Ada beberapa kondisi yang telah ditambahkan pada system akan diproses. Jika ada suatu kondisi maka system akan kembali pada langkah ke 2 dan akan mencari *rule* pada *knowledgebase* lagi. Jika tidak ada kondisi baru lagi maka sesi ini akan berakhir.

2.5 Analisis Dan Perancangan Berbasis Obyek

Analisis berorientasi objek (*Object Oriented Analysis* atau OOA) adalah teknik yang fokus pada objek-objek pemodelan yang merangkum pertimbangan terhadap data dan proses yang beraksi pada data tersebut. Model-model OOA adalah gambar-gambar yang mengilustrasikan objek-objek sistem dari berbagai macam perspektif (struktur, kelakuan dan interaksi objek-objek). Model OOA menggunakan *Unified Modeling Language* (UML). UML menyediakan sintaks grafis untuk keseluruhan rangkaian model-model objek juga mendefinisikan beberapa tipe diagram yang berbeda yang secara kolektif memodelkan sebuah sistem informasi atau aplikasi dalam artian objek-objek[8].

Unified Modeling Language (UML) adalah satu set peragaan konvensi yang digunakan untuk menetapkan atau menguraikan suatu sistem perangkat lunak dalam bentuk objek. Alat bantu yang digunakan dalam analisis berorientasi objek dengan UML antara lain adalah[8]:

1. *Use Case Diagram*

   Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuh *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakam sebuah pekerjaan tertentu. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. Sebuah *use case* dapat di include oleh lebih dari satu *use case* lain. Selama tahap desain *use case* diagram menetapkan perilaku sistem saat diimplementasikan.
2. Diagram Aktivitas (*Activity Diagram*)

   Activity Diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja. *Activity* diagram menyediakan gambaran visual dari aliran aktifitas, baik dalam sistem, bisnis dan alur kerja. Diagram ini berfokus pada kegiatan yang dilakukan dan siapa (atau apa) yang bertanggung jawab atas kinerja dari kegiatan tersebut.
3. *Class Diagram*

   Class Diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. *Class* Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. Sebuah class diagram digunakan untuk menunjukkan keberadaan dari kelas dan hubungannya di dalam pandangan logis dari sebuah sistem
4. *Statechart Diagram*

   Statechart Diagram mengganmbarkan transisi dan perubahan keadaan (dari satu *sta*te ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart* diagram menggambarakan *class* tertentu (satu *class* dapat memliki lebih dari satu *statechart* diagram).
5. *Sequence Diagram*

   Sequence Diagram menggambarkan interaksi antar objek didalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

3. Hasil dan Pembahasan

3.1 Halaman Login

Halaman login merupakan halaman untuk memasukkan nama pengguna dan *password*. Setelah semua data diinputkan, silahkan menekan tombol login dan akan masuk ke halaman utama untuk pengolahan data.

Gambar 4. Halaman Login

3.2 Halaman Input Data Pasien

Gambar 5 merupakan tampilan *interface input* untuk data pasien meliputi no.id, nama, jenis kelamin dan foto dari pasien, kemudian data tersebut disimpan ke database dengan mengklik tombol *save*.

Gambar 5. *Input* Data Pasien

3.3 Laporan Data Pasien

Setelah admin menginput data pasien pada gambar 5 diatas, maka dihasilkan keluaran atau *output* yang tampak pada Gambar 6.

Gambar *6. Output* Data Pasien

3.4 Input Data Penyakit

Interface pada Gambar 7 adalah *input* data penyakit yang meliputi nama penyakit, defenisi penyakit, anjuran pengobatan dan terapi penyakit.

Gambar 7. *Input* Data Penyakit

3.5 Output Data Penyakit

Hasil *output* dari input data penyakit pada gambar 7 akan tampak pada gambar 8.

Gambar 8. *Output* Penyakit

3.6 Input Gejala Penyakit

Interface pada Gambar 9 adalah *input* gejala penyakit yang meliputi kode gejala, nama gejala, pertanyaan dan keterangan.

Gambar 9. *Input* Gejala Penyakit

3.7 Output Gejala Penyakit

Output dari gambar 9 yaitu input gejala penyakit, dapat terlihat pada gambar 10 di bawah ini.

Gambar 10. *Output* Gejala Penyakit

4. Kesimpulan

Kesimpulan dari penelitian ini adalah dengan merancang sebuah sistem pakar dengan algoritma *forward chaining* maka dapat memudahkan *user untuk* mendapatkan informasi mengenai gejala dan jenis penyakit vertigo dan dapat memberikan fasilitas konsultasi bagi user guna mendiagnosa kemungkinan *user* terkena vertigo sedangkan pada pengujian perangkat lunak menggunakan *black box* dengan menguji beberapa kriteria form tampilan dengan hasil pengujian semua sistem dapat berjalan dengan baik dan sesuai dengan harapan.

Daftar Pustaka.

[1] George B., Ben B. (2015). Digital Planet. *Technologi and You.* 10th Edition. Pearson.
[2] Pustaka POLRI, 2014, Buku Petunjuk Klinik POLRI, Pustaka POLRI, Jakarta
[3] Leony Lydia. (2013). *Algoritma Dan Pemograman*. Gramedia.Jakarta.
[4] Agi Hindasah, 2012, Sistem Pakar Dengan Metode Forward Chaining, Penerbit Andi, Yogyakarta.
[5] Kusrini, 2015, Aplikasi Sistem Pakar, Andi Publisher, Yogyakarta.
[6] Sri Sutarni. (2015). Vertigo. Gadjah Mada University Press. Yogyakarta.
[7] Sustrani, 2014, Forward Chaining dan Backward Chaining, Elex Media Komputindo, Jakarta.
[8] Yuni Sugiarti, 2015, Analisis dan Perancangan Sistem dengan UML, Penerbit Graha Ilmu, Yogyakarta