

Sistem Pakar Untuk Mendiagnosa Penyakit Infeksi Saluran Pernafasan Akut (ISPA) Berbasis Web

Ardian Bagus Prabowo, *Member, IEEE*

Abstract— Acute Respiratory Infection (ARI) consists of several terms, namely Infectious Microorganisms are the entry into the human body and proliferate to cause disease. Respiratory tract is the organ from the nose to the alveoli along with its adnexal organs such as the sinuses, middle ear and pleural cavity. Acute infection is an infection that lasts up to 14 days. Acute Respiratory Infection can be caused by a variety of causes such as bacteria, viruses, mycoplasma, fungi and others. Acute Respiratory Infections upper part is generally caused by a virus, while the Acute Respiratory Infections bottom can be caused by bacteria, viruses and mycoplasma. Acute Respiratory Infections bottom caused by bacteria generally have severe clinical manifestations causing some problems in handling. Expert system is one of the artificial intelligence that contains the knowledge and experience by many experts to be included in a particular area of knowledge so that everyone used to use to solve a variety of problems that are specific, expert systems can be used in various fields of science, technology and so on.

Forward Chaining method is one method. processing that starts from a set of data for later performed in accordance with the rules of inference are applied to the conclusion that the optimum found. Inference engine will continue to loop in the decision process to achieve appropriate results. The advantages of this method is a forward chaining new data can be inserted into a database table inference and the possibility to make changes inference rules.

Index Terms— Expert System, Acute Respiratory Infection, Methods Chaining Forward

I. PENDAHULUAN

Infeksi Saluran Pernapasan Akut (ISPA) terdiri dari beberapa pengertian yaitu Infeksi adalah masuknya *Mikroorganisme* ke dalam tubuh manusia dan berkembang biak sehingga menimbulkan penyakit. Saluran pernapasan adalah organ mulai dari hidung hingga Alveoli beserta organ *Adneksanya* seperti sinus, rongga telinga tengah dan pleura. Infeksi akut adalah infeksi yang berlangsung sampai dengan 14 hari. Batas 14 hari diambil untuk menunjukkan proses akut meskipun untuk beberapa penyakit yang dapat digolongkan dalam ISPA, proses ini dapat berlangsung lebih dari 14 hari. Sedangkan *Pneumonia* adalah proses infeksi akut yang mengenai jaringan paru-paru (*Alveoli*). Terjadi *pneumonia* pada anak seringkali bersamaan dengan proses infeksi akut pada Bronkus disebut *Broncho pneumonia*. (Justin, 2007).

Berdasarkan pengertian di atas, maka ISPA adalah proses infeksi akut berlangsung selama 14 hari, yang disebabkan oleh mikroorganisme dan menyerang salah satu bagian, dan atau lebih dari saluran napas, mulai dari hidung (saluran atas) hingga *alveoli* (saluran bawah), termasuk jaringan *adneksanya*, seperti sinus, rongga telinga tengah dan pleura (Karna, 2006).

Penyakit ISPA dapat disebabkan oleh berbagai penyebab seperti bakteri, virus, mycoplasma, jamur dan lain-lain. ISPA bagian atas umumnya disebabkan oleh Virus, sedangkan ISPA bagian bawah dapat disebabkan oleh

bakteri, *virus* dan *mycoplasma*. ISPA bagian bawah yang disebabkan oleh bakteri umumnya mempunyai manifestasi klinis yang berat sehingga menimbulkan beberapa masalah dalam penanganannya.

II. METODE YANG DIUSULKAN

Metode yang dapat digunakan dalam mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu :

1. Penalaran Maju (*Forward Chaining*)

Pada penalaran maju, aturan-aturan diuji satu demi satu dalam urutan tertentu, urutan tersebut mungkin berupa aturan ke dalam perangkat aturan atau dapat juga urutan lain yang ditentukan oleh pemakai. Dalam pengujian tersebut sistem pakar berusaha mengevaluasi apakah kondisinya benar atau salah. Jika kondisi benar, aturan tersebut ditembakkan dan aturan berikutnya diuji. Jika kondisinya salah, aturan tersebut tidak ditembakkan dan aturan berikutnya diuji. Jika kondisinya salah, aturan tersebut tidak ditembakkan dan aturan berikutnya diuji.

Suatu aturan mungkin tidak dievaluasi sebagai benar atau salah. Mungkin kondisinya mencakup satu atau beberapa variabel dengan nilai yang tidak diketahui. Dalam hal itu, kondisi aturannya tidak diketahui. Jika kondisi aturan tidak diketahui, aturan tidak ditembakkan dan aturan berikutnya diuji.

Proses pengujian aturan satu demi satu berlanjut sampai putaran lengkap melalui seluruh perangkat aturan. Biasanya diperlukan lebih dari satu putaran untuk memberikan suatu nilai pada variabel sasaran. Mungkin informasi yang diperlukan untuk mengevaluasi satu aturan dihasilkan oleh aturan lain yang diuji

kemudian. Ketika tidak ada lagi aturan yang dapat ditembakkan, maka proses penalaran berhenti. Gambar 2.3 adalah contoh algoritma *forward chaining*

Forward chaining algorithm

```
function FOL-FC-ASK( $KB, \alpha$ ) returns a substitution or false
  repeat until new is empty
    new ← {}
    for each sentence r in KB do
      ( $p_1 \wedge \dots \wedge p_n \Rightarrow q$ ) ← STANDARDIZE-APART( $r$ )
      for each  $\theta$  such that  $(p_1 \wedge \dots \wedge p_n)\theta = (p'_1 \wedge \dots \wedge p'_n)\theta$ 
        for some  $p'_1, \dots, p'_n$  in KB
           $q' \leftarrow$  SUBST( $\theta, q$ )
          if  $q'$  is not a renaming of a sentence already in KB or new then do
            add  $q'$  to new
             $\phi \leftarrow$  UNIFY( $q', \alpha$ )
            if  $\phi$  is not fail then return  $\phi$ 
    add new to KB
  return false
```

Gambar 1 : Contoh algoritma *forward chaining*

2. Penalaran Mundur (*Backward Chaining*)

Pada penalaran mundur (*Backward Chaining*), inference engine memilih suatu aturan dan menganggapnya sebagai masalah yang harus diselesaikan. Dengan menggunakan perangkat aturan inference engine mulai mengevaluasi dari variabel sasaran, kemudian diikuti dengan pemilihan salah satu submasalah untuk dievaluasi, dan submasalah yang terpilih akan dievaluasi sebagai masalah baru. Inference engine terus mencari submasalah untuk menjadi masalah baru yang akan dievaluasi sampai dengan tidak ada lagi submasalah yang ditemui. Penalaran maju bergerak lebih cepat dari penalaran mundur karena penalaran mundur tidak harus mempertimbangkan semua aturan dan tidak membuat beberapa putaran melalui perangkat aturan. Penalaran mundur sangat sesuai digunakan jika :

- Terdapat variabel sasaran berganda (*multiple goal variable*)
- Terdapat banyak aturan.
- Semua atau hampir semua aturan tidak harus diuji dalam proses mencapai pemecahan

```
function FOL-BC-ASK( $KB, goals, \theta$ ) returns a set of substitutions
  inputs: KB, a knowledge base
         goals, a list of conjuncts forming a query
          $\theta$ , the current substitution, initially the empty substitution {}
  local variables: ans, a set of substitutions, initially empty

  if goals is empty then return { $\theta$ }
   $q' \leftarrow$  SUBST( $\theta, \text{FIRST}(goals)$ )
  for each r in KB where STANDARDIZE-APART( $r$ ) =  $(p_1 \wedge \dots \wedge p_n \Rightarrow q)$ 
    and  $\theta' \leftarrow$  UNIFY( $q, q'$ ) succeeds
      ans ← FOL-BC-ASK( $KB, [p_1, \dots, p_n, \text{REST}(goals)], \text{COMPOSE}(\theta', \theta)$ )  $\cup$  ans
  return ans
```

Gambar 2 : Contoh algoritma *backward chaining*

3. Metode Pengembangan Sistem

- Dalam perancangan Tugas Akhir ini penulis menggunakan metode *Prototype*. *Prototype Model* adalah salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan Metode Prototyping ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dibutuhkan, Pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang memperhatikan efisiensi algoritma. Kemampuan sistem operasi dan interface yang menghubungkan manusia dengan komputer. Pada Prototyping model kadang – kadang klien hanya memberikan beberapa kebutuhan umum software tanpa detail input, proses atau detail output dilain waktu mungkin tim pembangun (developer) tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form user interface. Ketika situasi seperti ini terjadi model prototyping sangat membantu proses pembangunan software. Proses pada prototyping bisa dijelaskan sebagai berikut :
 - Pengumpulan Kebutuhan : developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.
 - Perancangan : Perancangan dilakukan cepat dan rancangan mewakili aspek software yang diketahui. Dan rancangan ini menjadi dasar pembuatan prototype.
 - Evaluasi Prototype : klien mengevaluasi prototype yang dibuat dan dipergunakan untuk memperjelas kebutuhan software.



Gambar 3 : Model Prototype menurut Roger S. Pressman, Ph.D. 2005.

3.1 Tahapan-tahapan *Prototype*

Tahap-tahap pengembangan *Prototype* model menurut Roger S. Pressman, Ph.D. 2005 adalah :

1. Mendengarkan pelanggan Pada tahap ini dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

Kegiatan yang dilakukan pada tahap perencanaan sistem meliputi :

- a. Mendefinisikan masalah.

Berdasarkan kegiatan yang telah dilakukan pada tahap perencanaan aplikasi, maka definisi permasalahan yaitu:

- 1) Akibat penyakit ISPA yang menimbulkan kematian, dimana penyakit ISPA dapat menyerang semua usia dan virus penyebab penyakit ISPA semakin berkembang.
- 2) Penanganan dan pencegahan awal terhadap penyakit ISPA, bertujuan untuk mengurangi berkembangnya penyakit ISPA.
- 3) Penanganan dan pencegahan terhadap gejala-gejala yang timbul yang dapat menyebabkan penyakit ISPA.

- b. Menentukan tujuan Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web yang akan dirancang.

Tujuan Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web adalah implementasi aplikasi sistem pakar untuk diagnosa penyakit ISPA memberikan informasi tentang penyakit ISPA yang digunakan untuk pendiagnosaan awal untuk tindakan terhadap penyakit ISPA.

2. Merancang dan Membuat Prototype Pada tahap ini, dilakukan perancangan dan pembuatan prototype sistem. *Prototype* yang dibuat disesuaikan dengan kebutuhan sistem yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.

Kegiatan yang dilakukan pada tahap ini meliputi :

- a. Pembuatan pemodelan proses.

Pemodelan proses Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web menggunakan *Unified Modeling Language* (UML). UML yang akan dirancang yaitu: diagram proses diagnosa, diagram hasil diagnosa, diagram proses saran atau kritik, diagram proses login dan diagram proses administrasi.

- b. Pembuatan pemodelan data

Pemodelan proses Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web menggunakan *Class Diagram*. *Class Diagram* yang akan dirancang yaitu: pemodelan data admin, data gejala, data kerusakan, data rule dan data buku tamu.

- c. Pembuatan desain antarmuka (*interface*) Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web

Desain antarmuka yang akan dibuat yaitu: desain form login, form gejala, form kerusakan, form rule, form diagnosa, form hasil diagnosa penyakit, form buku tamu, dan form pengeditan data login admin.

3. Uji coba Pada tahap ini, *Prototype* dari system di uji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki *Prototype* yang ada.

Tahap implementasi dan pengujian merupakan kegiatan yang dilakukan setelah Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web selesai dibuat. Kegiatan yang dilakukan pada tahap ini yaitu:

- a. Menyiapkan perangkat keras.

Perangkat keras yang dibutuhkan yaitu:

- 1) Perangkat PC Klien dengan spesifikasi minimum yaitu: prosesor pentium dual core, kapasitas ram sebesar 1 GB, dan memiliki kartu vga tambahan.
- 2) Web Server, dengan spesifikasi minimum yaitu: prosesor pentium dual core, kapasitas ram sebesar 1 GB.
- 3) Printer yang digunakan untuk sarana mencetak laporan penulisan.

- b. Menyiapkan perangkat lunak.

Perangkat lunak yang dibutuhkan yaitu:

- 1) Sistem operasi Windows XP atau Windows 7.
- 2) AppServ 2.5.10.
- 3) Adobe Dreamweaver CS3.

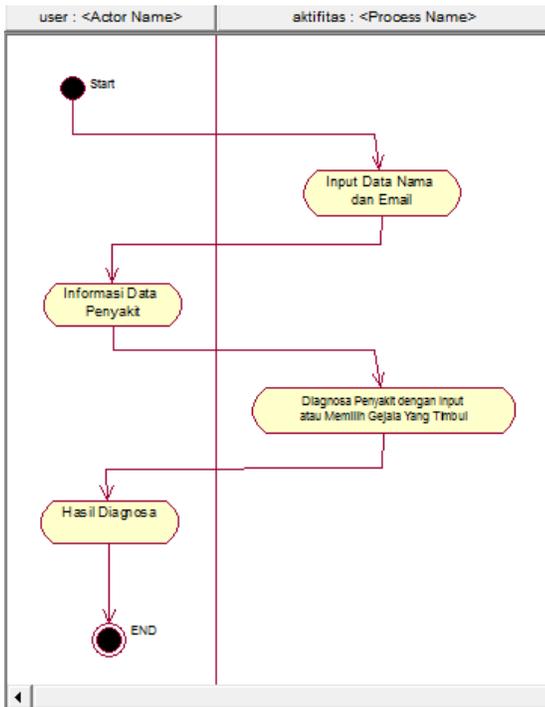
- c. Menerapkan Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web.

- d. Melakukan pengujian terhadap aplikasi yang telah dibuat.

III. IMPLEMENTASI

3. Activity Diagram

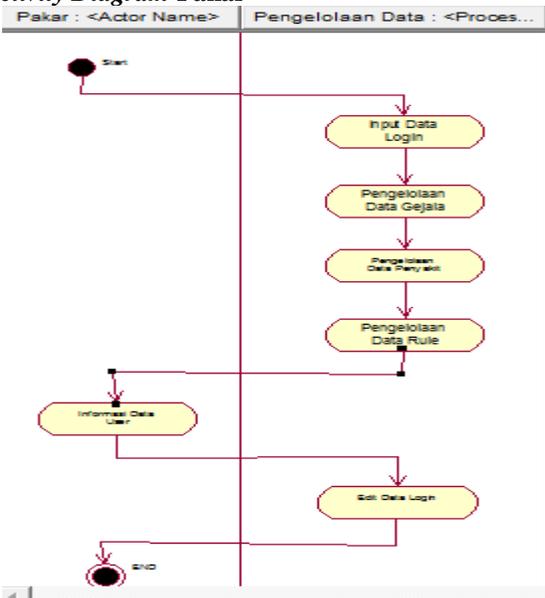
3.1 Activity Diagram User



Gambar 4 : Activity Diagram User

Activity diagram user diatas menerangkan bahwa arus data secara umum aktifitas yang dilakukan oleh user, yaitu user menginputkan data nama dan email sebelum dapat mengakses sistem pakar, setelah mengisikan data nama dan email user memperoleh informasi data penyakit, serta user dapat melakukan diagnosa penyakit dengan memilih atau input data gejala yang timbul atau dirasakan kemudian dari proses diagnosa user memperoleh hasil diagnosa.

1. Activity Diagram Pakar

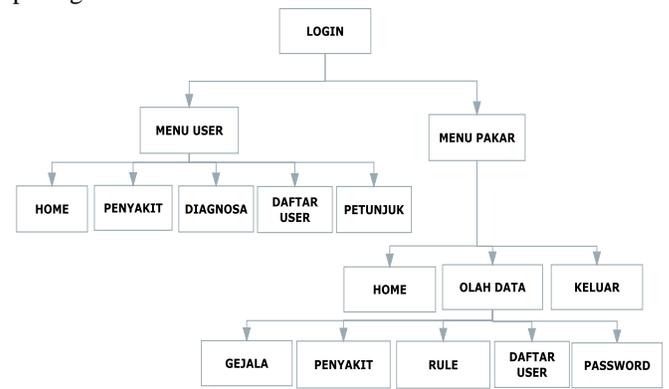


Gambar 5 : Activity Diagram Pakar

Activity diagram pakar diatas menerangkan bahwa arus data secara umum aktifitas yang dilakukan oleh pakar, yaitu pakar sebagai pengelola sistem pakar sebelum mengelola data harus melakukan login terlebih dahulu setelah berhasil login pakar dapat melakukan pengelolaan data meliputi data gejala, data penyakit dan data rule, pakar juga memperoleh informasi data user yang menggunakan sistem pakar, serta pakar juga dapat melakukan edit data login pakar.

4.2.4 Perancangan Struktur Program

Dalam perancangan sebuah aplikasi diperlukan adanya struktur program yaitu suatu aliran informasi yang dapat menjelaskan bagaimana urutan program itu akan beroperasi. Berikut ini adalah tampilan struktur program dari pembuatan program sistem pakar yang dapat dilihat pada gambar 6.



Gambar 6 : Struktur Program

IV. HASIL & PEMBAHASAN

4. Representasi Pengetahuan Pakar

Model representasi pengetahuan untuk sistem pakar diagnosa penyakit ISPA berbasis web ini menggunakan bentuk kaidah produksi yang dituliskan dalam bentuk jika-maka (*if-then*). Tabel penyakit menjelaskan tentang diagnosa penyakit ISPA berbasis web. Tabel penyakit ditunjukkan seperti pada Tabel 1.

No.	Kode Penyakit	Nama Penyakit
1.	P1	Infeksi saluran nafas atas akut
2.	P2	Hipertenansi esensial (primer)
3.	P3	Faringitis akut
4.	P4	Demam tifoid dan ISPA
5.	P5	Gastritis dan ISPA
6.	P6	Dermatitis dan ISPA

Tabel 1 : Tabel Penyakit

4.1 Aturan Kaidah Produksi Pakar

Kaidah produksi biasanya dituliskan dalam bentuk *IF-THEN*, kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian yaitu bagian premis (jika) dan bagian

konklusi (maka), apabila bagian premis dipenuhi, maka bagian konklusi juga akan bernilai benar.

Tabel keputusan sistem pakar diagnosa penyakit ISPA berbasis web ditunjukkan pada Tabel 2.

Tabel 2 : Tabel Keputusan

Kode	Gejala Penyakit	P1	P2	P3	P4	P5	P6
G1	Bapil (Batuk pilek)	v					V
G2	Greges	v		V			
G3	Tenggorokan sakit	v		V			V
G4	Berdahak	v					V
G5	Panas	v			V	v	
G6	Pusing		v		V	v	
G7	Lemes		v				
G8	Mual		v		V	v	
G9	Muntah		v			v	
G10	Pendarahan di Hidung		v				
G11	Nyeri telan			v			
G12	Susah tidur			v			
G13	Tonsil Membesar			v			
G14	Diare				v		
G15	Dingin				v		
G16	Perut Kembung					v	
G17	Gatal-Gatal						V
G18	Muncul Bercak Merah						V

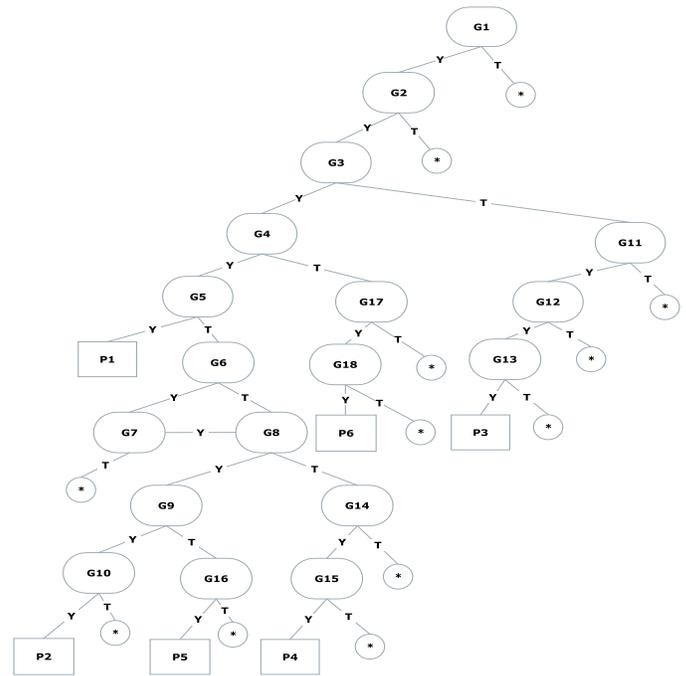
Untuk masing-masing area gejala, terdapat juga 6 aturan kaidah produksi gejala penyakit dalam bentuk *IF-THEN rules*. Tabel 3 berikut ini merupakan tabel pembentukan rule dari sistem pakar yang akan dibangun.

Tabel 3 : Tabel Pembentukan Rule

RULE	IF	THEN
R1	G1, G2, G3,G4,G5	P1
R2	G6, G7, G8, G9,G10	P2
R3	G2, G3, G11,G12,G13	P3
R4	G5, G6, G8,G14,G15	P4
R5	G5, G6,G8,G9,16	P5
R6	G1, G3,G4,G17,G18	P6

4.2 Perancangan Pohon Keputusan Diagnosa Penyakit ISPA

Diagram pohon keputusan merupakan suatu rancangan yang digunakan untuk membangun sebuah sistem pakar, di dalam diagram pohon keputusan tersebut akan dicari solusi hasil akhir dari setiap pemeriksaan. Diagram pohon keputusan akan mempermudah untuk menyusun basis pengetahuan dan aturan serta menentukan faktor kepastian dari setiap pelaksanaan Diagnosa gejala pada penyakit ISPA.



Gambar 7 : Pohon Keputusan diagnosa penyakit ISPA

Keterangan :

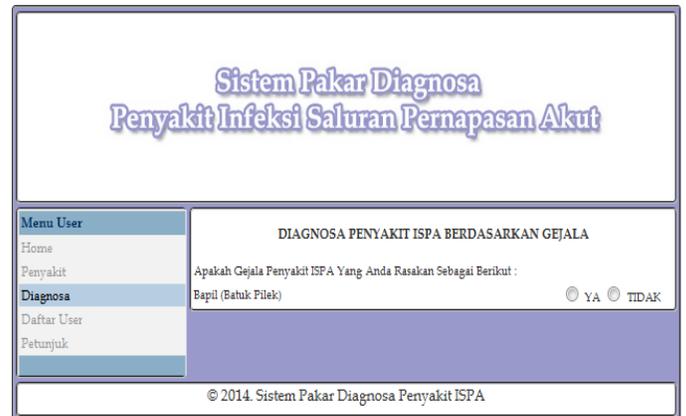
G1 – G18 = Gejala

P1 – P6 = Penyakit

* = Tidak Terdiagnosa

4.3 Diagnosa Penyakit

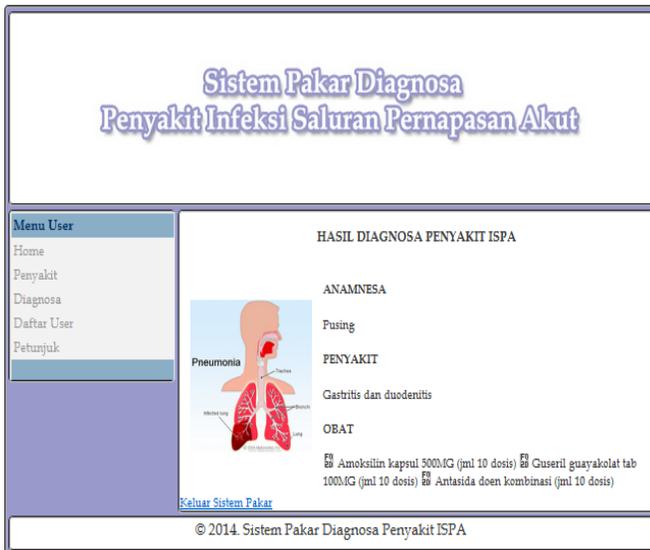
Halaman diagnosa penyakit merupakan halaman yang digunakan untuk pendiagnosaan penyakit penyakit ISPA yang dapat digunakan untuk user atau para pengunjung aplikasi



Gambar 8 : Halaman Diagnosa Penyakit

4.4 Hasil Diagnosa Penyakit

Halaman hasil diagnosa penyakit merupakan halaman yang menampilkan hasil pendiagnosaan penyakit penyakit ISPA yang dilakukan user atau para pengunjung aplikasi dengan memilih atau input data gejala penyakit yang dirasakan oleh user.



Gambar 9 : Hasil Diagnosa Penyakit

V. PENUTUP

5. Kesimpulan

Dari penjelasan Bab Pendahuluan sampai dengan Bab Perancangan dan Implementasi penulis dapat merangkum beberapa kesimpulan, yaitu sebagai berikut :

1. Perancangan suatu perangkat lunak sistem pakar untuk mendiagnosa penyakit ispa dengan metode *forward chaining* berbasis web dibangun dengan bahasa pemrograman PHP, database *MySQL* dan pengujian menggunakan server *Apache*.
2. Implementasi aplikasi sistem pakar untuk mendiagnosa penyakit ISPA ini memberikan informasi tentang pengenalan penyakit ISPA yang dibutuhkan oleh masyarakat sebagai sarana untuk mencegah atau tindakan awal terhadap penyakit ISPA.

REFERENCES

- [1] Amrullah, Afif. 2002. *Unified Modeling Language (UML)*. Pustaka, Bandung.
- [2] Anonim. 2002. *Program Pemberantasan Penyakit ISPA untuk penanggulangan Pnemonia pada Balita Dalam Pelita VI*. Jakarta.
- [3] Eriz, PW. 2013. *Sistem Pakar Diagnosa Infeksi Saluran Pernafasan Akut (ISPA) Menggunakan Logika Fuzzy*. Universitas Tanjungpura
- [4] Iskandar, Ed. 2007. *Sistem Pakar Untuk Diagnosa Penyakit ISPA Menggunakan Metode Faktor Kepastian*. STMIK EL RAHMA Yogyakarta
- [5] Justin. 2006. *Hubungan Sanitasi Rumah Tinggal Dengan Kejadian Penyakit Pneumonia*. Unhalu. Kendari.
- [6] Kadir, Abdul. 2002. *Pengenalan Sistem Informasi*. Andi. Yogyakarta.
- [7] Kristanto, Andi. 2003. *Perancangan Sistem Informasi dan Aplikasinya*. Gava Media. Jakarta.

- [8] Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta.
- [9] Nugroho, Adi. 2005. *Analisis dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Informatika. Bandung.
- [10] Nugroho, Bunafit. 2004. *Databases Relasional Dengan MySql*. Andi. Yogyakarta
- [11] Turban, Efraim. 2005. *Decision Support Systems and Intelligent Systems*, edisi Bahasa Indonesia jilid 1. Andi. Yogyakarta.
- [12] Pressman, R.S. 2005. *Software Engineering: A Practitioner's Approach*, Forth Edition, McGraw-Hill Book, Co.