

BAB II

LANDASAN TEORI

2.1. Perbandingan Penelitian

Pada penelitian sebelumnya yang dilakukan Melisa (2015), dengan judul “Perancangan Sistem Pakar Diagnosa Penyakit Leukimia dengan Metode *Forward Chaining*” menghasilkan sebuah sistem yang dapat membantu pasien dalam diagnosa tahapan awal terlebih dahulu, dan membantu dokter dalam melakukan pemeriksaan dan memberikan hasil diagnosa sesuai dengan hasil diagnosa dari sistem pakar. Penelitian ini menggunakan metode *Forward Chaining* dan pengembangan perangkat lunak menggunakan bahasa pemrograman *PHP*.

Kurniawan (2011), melakukan penelitian dengan judul “Aplikasi Sistem Pakar untuk Diagnosa Penyakit Gigi dan Mulut” menghasilkan sebuah sistem yang dapat membantu dokter dalam mendiagnosa dan menanggulangi penyakit gigi dan mulut serta mampu menentukan diagnosa banding dari kedekatan gejala yang menyertai penyakit gigi dan mulut. Penelitian ini dibuat berbasis *web* dengan *database MySQL* dan menggunakan metode *Forward Chaining*.

Angraini (2011), melakukan penelitian dengan judul “Sistem Pakar Diagnosa Penyakit Paru-Paru menggunakan *Variable-Centered Intelligence*” menghasilkan sebuah sistem yang dapat membantu dokter dalam mendiagnosa penyakit paru-paru berdasarkan gejala yang dirasakan serta memberikan solusi yang tepat. Sistem pakar ini dibangun menggunakan metode *Variable-Centered Intelligent Rule System* dan menggunakan teknik

forward chaining dengan sistem yang dibangun berbasis *web* dan menggunakan bahasa pemrograman *PHP* dan *database MySQL*.

Maria (2017), melakukan penelitian dengan judul “Sistem Pakar untuk Mendeteksi Penyakit dengan Gejala Demam pada Anak”. Menghasilkan sistem yang dapat membantu dokter dalam mendiagnosa penyakit demam pada anak sistem pakar yang dibangun berbasis *web* dengan metode *Forward Chaining*.

Pasaribu (2015), melakukan penelitian dengan judul “Implementasi Sistem Pakar untuk Diagnosa Penyakit Mata pada Manusia”. Menghasilkan sebuah sistem yang dapat membantu dokter dalam mendiagnosa penyakit mata dengan menjawab pertanyaan yang diberikan oleh sistem dengan penyimpanan pada *database*. Sistem ini menggunakan metode *forward chaining* dan pemrograman berbasis *web*. Penelitian-penelitian sebelumnya dapat dilihat pada tabel berikut:

Tabel 2.1 Perbandingan Penelitian

No	Nama	Judul Penelitian	Metode Yang Digunakan	Hasil
1	Melisa (2015)	Perancangan sistem pakar diagnosa penyakit Leukimia	<i>Forward chaining</i>	Sebuah sistem pakar yang dapat membantu dokter dalam mendiagnosa dan menaggulangi penderita penyakit

				leukemia.
2	Budi Kurniawan (2011)	Aplikasi sistem pakar untuk diagnosa penyakit gigi dan mulut	<i>Forward chaining</i>	Sebuah sistem pakar yang dapat membantu dokter dalam mendiagnosa dan menaggulangi penyakit gigi dan mulut serta mampu menentukan diagnosa banding dari kedekatan gejala yang menyertai.
3	Felvi Angraini (2011)	Sistem pakar diagnosa penyakit paru-paru	<i>Variable- centered intelligent</i>	Sistem pakar yang dapat membantu dokter dalam mendiagnosa penyakit paru-paru dan memberikan informasi pada pengguna.
4	Maria Soares (2017)	Sistem Pakar untuk mendeteksi penyakit dengan gejala	<i>Forward chaining</i>	Sistem yang dapat mendiagnosa penyakit pada anak

		demam pada anak		dengan menjawab pertanyaan berupa gejala yang dialami yang disediakan oleh sistem.
5	Johani Pasaribu (2015)	Implementasi sistem pakar untuk mendiagnosa penyakit mata pada manusia	<i>Forward chaining</i>	Sistem yang dapat mendiagnosa penyakit mata dengan menjawab pertanyaan dari sistem dan menampilkan hasil diagnosa dengan beberapa jenis penyakit mata.

Penelitian ini merujuk pada penelitian yang dilakukan oleh Pasaribu (2015). Penelitian tersebut menggunakan beberapa jenis penyakit mata dan menggunakan metode *forward chaining* dengan program sistem pakar berbasis *web*. Perbandingan dengan penelitian terdahulu adalah penelitian sekarang membahas masalah rancang bangun Sistem Pakar Diagnosis Penyakit Hepatitis dengan menggunakan metode *forward chaining* dan kelebihan dari sistem ini adalah sistem ini dapat memberikan informasi kepada pengguna berupa jenis penyakit serta cara pencegahan pada pasien.

Basis pengetahuan yang dipakai terdiri dari 5 jenis penyakit hepatitis yaitu : jenis hepatitis A, B, C, D dan E serta tersedia penjelasan sebagaimana dokter memberikan penanganan kepada pasien yang mendapat pemeriksaan langsung dengan dokter bagi pengguna sistem yang dibangun.

2.2. Sistem Pakar (*Expert System*)

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Kusrini, 2006).

Sistem pakar adalah salah satu cabang dari AI (*Artificial Intelligence*) yang membuat pengguna secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya (Arhami, 2005).

Sistem pakar yang diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain: pembuatan keputusan (*desicion making*), pemaduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar (Kusrini, 2006).

2.2.1. Ciri-ciri Sistem Pakar

Adapun ciri-ciri sistem pakar adalah sebagai berikut (Kusrini, 2006).

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Dirancang untuk dikembangkan secara bertahap.
6. *Output*-nya bersifat nasihat atau anjuran.
7. *Output* tergantung dari dialog dengan *user*.

2.2.2. Tujuan Sistem Pakar

Sistem pakar bertujuan untuk mentransfer kepakaran yang dimiliki seorang pakar kedalam komputer, dan kemudian kepada orang lain (*nonexpert*). Aktivitas yang dilakukan untuk memindahkan kepakaran adalah (Arhami, 2005):

1. *Knowledge Acquisition* (dari pakar atau sumber lainnya)
2. *Knowledge Representation* (ke dalam komputer)
3. *Knowledge Inferencing*
4. *Knowledge Transferring*

2.2.3. Komponen-Komponen Sistem Pakar

Komponen-komponen yang terdapat dalam struktur sistem pakar adalah sebagai berikut (Arhami, 2005):

1. *User Interface*

User interface atau antarmuka pengguna merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

2. *Knowledge Base*

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Pengetahuan itu dapat berasal dari ahli, buku, basis data, penelitian dan gambar.

Ada 2 bentuk pendekatan basis pengetahuan:

a. Penalaran berbasis aturan (*rule-based reasoning*)

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *IF-THEN*. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan pakar dapat menyelesaikan masalah tersebut secara berurutan. Di samping itu, bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) pencapaian solusi.

b. Penalaran berbasis kasus (*case-based reasoning*)

Berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan

apabila *user* menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu bentuk ini juga digunakan bila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

3. *Knowledge Acquisition*

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan yang diakuisisi adalah pengetahuan *procedural* (apa yang harus dilakukan, berupa aturan, prosedur, metode, dan lain-lain). Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai. Ada beberapa tantangan dalam melakukan akuisisi yaitu pengetahuan yang tidak lengkap, pengetahuan yang salah, kemampuan menjelaskan pengetahuan dan pandangan yang berbeda dari beberapa pakar.

Metode akuisisi pengetahuan sebagai berikut:

- a. Wawancara : metode yang paling banyak digunakan, yang melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara.
- b. Analisis protokol : dalam metode ini pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Pekerjaan tersebut direkam, dituliskan, dan dianalisis.

4. *Inference Engine*

Mesin Inferensi (*Inference Engine*), merupakan otak dari sistem pakar, juga dikenal sebagai penerjemah aturan (*rule interpreter*). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan. Kerja mesin inferensi meliputi:

- a. Menentukan aturan mana yang akan dipakai.
- b. Menyajikan pertanyaan kepada pemakai, ketika diperlukan.
- c. Menambahkan jawaban ke dalam memori Sistem Pakar.
- d. Menyimpulkan fakta baru dari sebuah aturan.
- e. Menambahkan fakta tadi ke dalam memori.

Ada 2 cara dalam melakukan inferensi :

- a. *Forward Chaining*: pencocokkan fakta atau pernyataan dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.
- b. *Backward Chaining*: pencocokkan fakta atau pernyataan dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.

5. *Workplace / Black board*

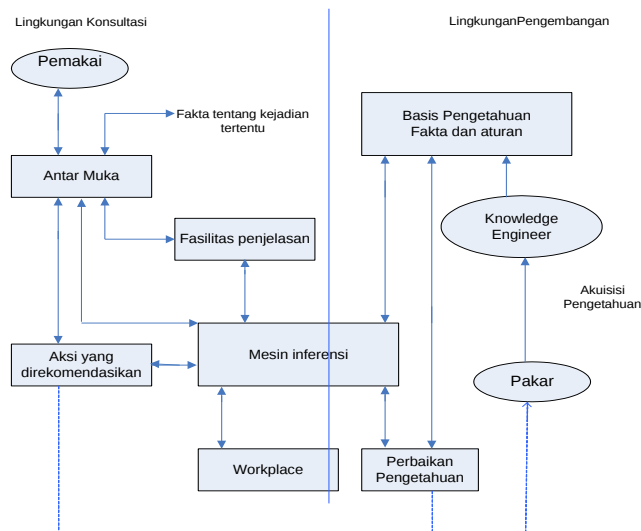
Workplace merupakan area dari sekumpulan memori kerja (*working memory*), digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara.

6. *Explanation Facility*

Kemampuan untuk menjejak (*tracing*) bagaimana suatu kesimpulan dapat diambil merupakan hal yang sangat penting untuk transfer pengetahuan dan pemecahan masalah.

7. *Repair Of Knowledge*

Menurut (Arhammi, 2005), Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya. Berikut arsitektur dari sistem pakar terdapat pada gambar berikut:



Gambar 2.1 Arsitektur Sistem Pakar (Turban, 1995).

Komponen-komponen yang terdapat dalam sistem pakar seperti yaitu *User Interface* (antarmuka pengguna), basis pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan.

7.1. Antarmuka Pengguna (*User Interface*)

User Interface merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

7.2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

7.3. Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan diperoleh dari pakar, dilengkapi

dengan buku, basis data, laporan penelitian dan pengalaman pemakai.

7.4. Mesin Inferensi

Komponen ini mengandung pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan.

7.5. *Workplace*

Workplace merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil antara dan kesimpulan yang dicapai.

7.6. Fasilitas Penjelasan

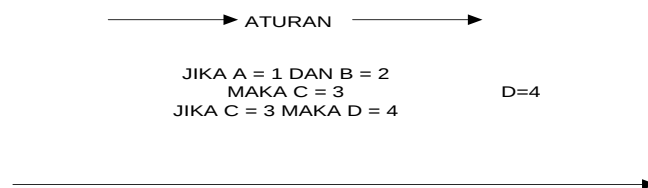
Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai.

7.7. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

2.3. Metode Runut Maju (*Forward Chaining*)

Menurut Kusrini (2006), 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 menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Wilson, 1998). Cara kerja metode inferensi runut maju dapat dilihat pada gambar berikut:



Gambar 2.2 Metode Runut Maju (Kusrini, 2006).

Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*).

Contoh inferensi dengan menggunakan runut maju sebagai berikut:

JIKA penderita terkena penyakit epilepsi adapatik dengan CF antara
0,4 s/d 0,6

MAKA berikan obat *carbamazepine* (Kusrini, 2006).

2.4. Web

Web adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser* (Yuhefizar, 2009).

2.5. *PHP*

PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses di sisi *server*. Fungsi utama *PHP* dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. Data *website* akan dimasukkan ke *database*, diedit, dihapus, dan ditampilkan pada *website* yang diatur oleh *PHP* (Abdulloh, 2015).

2.6. *MySQL*

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrograman aplikasi *web*. Contoh DBMS lainnya adalah: *Postgre SQL (freeware)*, *SQL Server*, *MS Access* dari *Microsoft*, *DB2* dari *IBM*, *Oracle* dan *Oracle Corp*, *Dbase*, *FoxPro*, dan sebagainya.

Kelebihan dari *MySQL* adalah gratis, handal, selalu di-*update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah (Hidayatullah, 2015).

2.7. *Macromedia Dreamweaver 8*

Macromedia dreamweaver merupakan aplikasi desain pengembangan *web* yang menyediakan editor WYSIWYG visual (bahasa sehari-hari yang disebut sebagai *Design View*) dan kode editor dengan fitur standar seperti *syntax highlighting*, *code completion*, dan *code collapsing* serta fitur lebih canggih seperti *real time syntax cheking* dan *code introspection* untuk menghasilkan petunjuk kode, untuk membantu pengguna dalam menulis kode (Nugroho, 2014).

2.8. Basis Data

Dalam mengembangkan aplikasi pasti dibutuhkan media sebagai penyimpanan data. Salah satu metode untuk penyimpanan data adalah dengan menggunakan basis data. Basis data merupakan sekumpulan data yang saling berhubungan satu dengan yang lainnya, yang didesain untuk menyimpan berbagai informasi. Sehubungan dengan basis data ini terdapat suatu *software* yang memungkinkan kita untuk mendefinisikan, membaaur, memelihara dan mengontrol basis data tersebut. *Software* tersebut disebut *DBMS (Database Management Sysem)* (Fathansyah, 2015).

Dengan menggunakan basis data, ada beberapa keuntungan yang didapat yaitu:

- a. Penggunaan basis data memungkinkan untuk menghindari duplikasi data.
- b. Dapat meningkatkan konsistensi data sehingga data yang disimpan lebih dapat dipercaya.
- c. Dapat mengolah data tersebut menjadi informasi yang berguna.
- d. Data dapat diakses oleh beberapa sistem secara bersamaan tanpa harus meng-*copy* data tersebut ke dalam sistem-sistem tersebut secara terpisah.
- e. Oleh karena data yang disimpan lebih konsisten maka integritas data dapat ditingkatkan sehingga informasi yang didapatkan dari data tersebut lebih terpercaya.
- f. Meningkatkan kecepatan akses data.

2.9. Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer kedalam basis pengetahuan (Arhami, 2005).

2.10. Representasi Pengetahuan

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Representasi yang dimaksudkan untuk menangkap sifat-sifat penting problema dan membuat informasi itu dapat diakses oleh prosedur pemecahan problema.

Bahasa representasi harus dapat membuat seorang *programmer* mampu mengekspresikan pengetahuan yang diperlukan untuk mendapatkan solusi problema, dapat diterjemahkan ke dalam bahasa pemrograman dan dapat disimpan (Kusrini, 2006).

2.11. Tabel Keputusan

Tabel keputusan adalah tabel yang digunakan sebagai alat bantu untuk menyelesaikan logika di dalam program. Algoritma yang berisi keputusan bertingkat yang banyak sekali sangat sulit untuk digambarkan langsung dengan *structured english* atau *pseudocode*. Untuk hal tersebut dapat menggunakan tabel keputusan (Kursini, 2011).

2.12. Pohon Keputusan

Pohon keputusan adalah metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang sangat merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami, dan mereka juga dapat diekspresikan dalam bentuk bahasa basis data seperti *structured query language* untuk mencari *record* pada kategori tertentu. Pohon keputusan berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah variabel target. Karena pohon keputusan memadukan antara eksplorasi data dan pemodelan, maka pohon keputusan sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai model akhir dari beberapa teknik lain (Kusrini, 2011).

2.13. Penyakit Hepatitis

Penyakit hepatitis memiliki 5 jenis yaitu hepatitis A, B, C, D dan E, yang memiliki berbagai gejala dari gejala yang umum hingga gejala yang spesifik. Gejala dari kelima jenis penyakit hepatitis masing-masing memiliki gejala yang spesifik atau gejala yang mencirikan dari jenis penyakit hepatitis.

2.13.1 Prosedur Pemeriksaan

Prosedur pemeriksaan ini dilakukan untuk mengumpulkan informasi berupa gejala awal yang dialami oleh pasien.

a. Kumpul keterangan dari pasien (saat pasien melakukan konsultasi)

1. Nama Pasien, Jenis Kelamin, umur.

2. Gejala yang diderita pasien.
3. Melakukan tensi darah untuk mengetahui kondisi pasien.
4. Memeriksa kondisi pasien dari gejala penyakit yang terlihat seperti terdapat pembengkakan dan terlihat kuning pada bagian tubuh dan kelopak mata.

2.13.2 Penyebab Penyakit

Berbagai macam virus, alkohol dan penggunaan obat-obatan telah diketahui sebagai penyebab penyakit yang menyerang manusia sehingga menyebabkan manusia jatuh sakit. Penyakit yang disebabkan oleh infeksi virus, alkohol dan penggunaan obat-obatan adalah penyakit hepatitis. Berikut penjelasan tentang penyebab penyakit hepatitis:

1. Virus

- a. Hepatitis A, jenis penyakit ini disebabkan oleh buruknya tingkat kebersihan dan dapat menyebar melalui air dan makanan yang tidak sehat.
- b. Hepatitis B, jenis penyakit ini disebabkan oleh penggunaan jarum suntik yang tidak steril dan dapat menular melalui pertukaran cairan tubuh atau kontak dengan darah dari orang yang terinfeksi hepatitis B dan penularan dari ibu hamil kepada bayi-nya saat melahirkan, hubungan suami istri, dan transfusi darah.
- c. Hepatitis C, jenis penyakit ini disebabkan oleh infeksi virus yang ditularkan melalui kontak darah, transfusi, jarum suntik

(terkontaminasi), serangga yang menggigit penderita lalu menggigit orang lain disekitarnya.

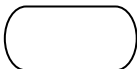
d. Hepatitis D, jenis penyakit ini disebabkan oleh virus yang ditularkan melalui cairan tubuh atau kontak langsung dengan penderita.

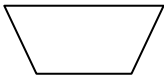
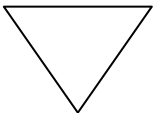
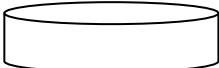
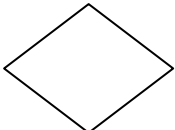
e. Hepatitis E, jenis penyakit ini disebabkan oleh virus yang ditularkan melalui makanan yang telah dikonsumsi oleh penderita penyakit hepatitis E.

2.14. *Flowchart*

Flowchart adalah bagan yang menjelaskan urutan dari prosedur-prosedur yang ada dalam sistem. Pada bagian ini akan digambarkan *flowchart* sistem yang akan dibangun. Pasien memasukkan data kemudian disimpan (proses rekam) ke dalam *database* dan juga mengalami proses rekam pada *database*. Tujuan adanya *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, dengan menggunakan simbol-simbol standar (Fathansyah, 2015). Simbol-simbol *Flowchart* terdapat pada tabel berikut:

Tabel 2.2 Simbol *Flowchart*

No	Simbol	Keterangan
1.		Mencetak keluaran dalam bentuk dokumen (melalui printer)
2.		Menyatakan permulaan atau akhir suatu program
		Memasukkan data secara manual dengan

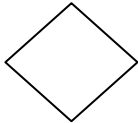
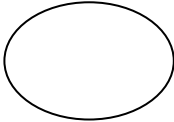
3. menggunakan *online keyboard*
4.  Menyatakan suatu tindakan (proses) yang dilakukan oleh Komputer
5.  Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh computer
6.  Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
7.  Menggambarkan informasi yang ditampilkan oleh perangkat *output*
8.  Data penyimpanan *database*
9.  Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/tidak
10.  Multi dokumen (Laporan Aktif)

2.15. *Entity Relationship Diagram (ERD)*

ERD dibuat untuk menunjukan obyek-obyek (himpunan entitas) apa saja yang ingin dilibatkan dalam sebuah *database* dan bagaimana hubungan terjadi diantara obyek-obyek tersebut. Dalam membentuk ERD ada dua komponen utama pembentuk model tersebut yaitu entitas (*entity*) dan relasi (*relation*). Entitas merupakan individu yang mewakili suatu yang nyata (eksistensinya)

dan yang dibedakan dari suatu yang lain (Fathansyah, 2015). Simbol-simbol ERD terdapat pada tabel berikut:

Tabel 2.3 Simbol ERD

SIMBOL	KETERANGAN
	Kumpulan entitas atau sesuatu yang dapat dibedakan atau didefinisikan
	Hubungan yang terjadi antara suatu entitas atau lebih entitas
	Atribut yaitu karakteristik dari entitas atau <i>relationship</i> yang menyediakan penjelasan <i>detail</i> tentang entitas atau <i>relationship</i>
	<i>Link</i>

2.16. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan asal data dan tujuan data yang keluar dari sistem, tempat penyimpanan data, proses apa yang menghasilkan data tersebut, serta interaksi antar data yang tersimpan dan proses yang dikenakan pada data tersebut (Fathansyah, 2015). Simbol-simbol yang ada pada *DFD* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol DFD

SIMBOL	KETERANGAN
--------	------------



Terminator: entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya *terminator* dikenal dengan nama entitas luar (*external entity*)



Komponen proses menggambarkan bagian dari sistem yang mentransformasikan *input* menjadi *output*



Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama kata benda jamak misalnya mahasiswa



Komponen data *flow* (alur data) digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses
