

BAB II **KAJIAN PUSTAKA**

2.1 Teori Dasar

Deskripsi teori dalam suatu penelitian merupakan uraian terstruktur mengenai teori (bukan hanya pendapat seorang pakar atau buku) dan hasil dari penelitian yang sesuai dengan variabel penelitian. Deskripsi teori memberikan penjelasan terhadap variabel yang diteliti melalui pendefinisian, dan uraian yang lengkap dan mendalam dari berbagai referensi, sehingga ruang lingkup, kedudukan dan prediksi terhadap hubungan antar variabel yang akan diteliti sehinggamenjadi lebih jelas dan terarah (Sugiyono, 2014) .

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence* (AI)

Istilah Kecerdasan Buatan (*Artificial Intelligence*) diusulkan pada tahun 1956 di konferensi Dartmouth. Pengembangan kecerdasan buatan terus dilakukan oleh berbagai penelitian yang membahas teori-teori dan prinsip-prinsipnya sejak saat itu. Walaupun istilah kecerdasan buatan baru muncul tahun 1956, tetapi konsep yang mengarah ke kecerdasan dimulai dari tahun 1941 (Suyanto, 2014).

Kecerdasan buatan asalnya dari bahasa Inggris “*Artificial Intelligence*” umumnya disebut dengan *AI*, yaitu *intelligence* merupakan suatu bentuk dari kata

sifat yang berarti cerdas, *artificial* itu sendiri artinya buatan. *AI* yang dimaksud yaitu merujuk kepada suatu mesin yang dapat berpikir, mengolah tindakan yang akan diambil, lalu dapat mengambil keputusan seperti yang bisa dilakukan oleh manusia. Seorang ahli matematika yang dijuluki dengan sebutan bapak komputer modern dan seorang pembongkar sandi Nazi pada Perang II tahun 1950 bernama Alan Turing, memutuskan definisi dari *Artificial Intelligence* “Jika komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai intelegensi”(Sutojo, Mulyanto, & Suhartono, 2011)

Berdasarkan definisi tersebut, kecerdasan buatanpun menawarkan media dan pengujian konsep mengenai kecerdasan. Konsep kecerdasan buatan kemudian dapat diolah kedalam suatu bahasa pemrograman dan eksekusinya dapat dibuktikan pada suatu aplikasi. Dapat disimpulkan cerdas: memiliki ilmu pengetahuan, memiliki pengalaman, dan memiliki penalaran dalam membuat keputusan dan mengambil tindakan. Jadi, supaya mesin yang diciptakan bisa cerdas (bertindak seperti manusia) maka harus dibekali dengan pengetahuan dan kemampuan untuk menalar. (Sutojo et al., 2011)

Kecerdasan buatan memiliki beberapa keuntungan dan kelebihan dibandingkan dengan kecerdasan yang dimiliki oleh manusia (kecerdasan alami), antara lain: (Sutojo et al., 2011)

1. Kecerdasan buatan memiliki sifat konstan atau permanen
2. Mudah diduplikasi dan disebarluaskan.

3. Biaya yang lebih murah jika dibandingkan dengan kecerdasan alami
4. Bersifat konsisten (tidak berubah-ubah)
5. Dapat diarsipkan
6. Efisien waktu karena mampu melakukan suatu pekerjaan dengan cepat.
7. Mengerjakan suatu hal atau jenis pekerjaan dengan baik dibandingkan dengan kecerdasan alami.

Permasalahan yang diatasi oleh kecerdasan buatan seiring dengan berjalannya waktu semakin meluas sehingga memungkinkan untuk memasuki bidang ilmu yang lain. Hal ini disebabkan karakteristik dari kecerdasan buatan sudah mulai dibutuhkan serta dikembangkan oleh berbagai disiplin ilmu pengetahuan dan teknologi (Sutojo et al., 2011).

Definisi *artificial intelligence* atau kecerdasan buatan yang paling tepat untuk saat ini adalah *acting rationally* dengan pendekatan *rational agent*. Hal yang mendasari pemikiran ini adalah penalaran logis dapat dilakukan oleh komputer dan mampu melakukan aksi yang masukan dengan hasil penalaran tersebut.(Suyanto, 2014)

2.1.1.1 Logika Fuzzy (*Fuzzy Logic*)

Untuk permasalahan yang mengandung ketidakpastian, *fuzzy logic* adalah pilihan yang tepat. *Fuzzy logic* menunjukkan performansi yang bagus untuk berbagai masalah, khususnya optimasi dan kontrol otomatis.

Logika *fuzzy* adalah peningkatan dari logika Boolean yang berhadapan dengan konsep kebenaran sebagian. Saat logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah biner (0 atau 1, hitam atau putih, ya atau tidak) logika *fuzzy* menggantikan kebenaran Boolean dengan tingkat kebenaran (Sutojo et al., 2011).

2.1.1.2 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Jaringan syaraf tiruan merupakan salah satu usaha manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan dan pengorganisasian jaringan syaraf tiruan berdasarkan kemampuan otak manusia yang disebut *neuron*, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi. Jaringan syaraf tiruan adalah prosesor tersebar paralel (*parallel distributed processor*) yang sangat besar yang memiliki kemampuan dalam menyimpan pengetahuan berdasarkan pengalaman sehingga siap untuk digunakan. Jaringan syaraf tiruan serupa dengan otak yang dimiliki manusia dalam dua hal, yaitu: Pengetahuan yang diperoleh jaringan syaraf tiruan melalui pembelajaran; Keuatan hubungan antar sel syaraf (*neuron*) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan (Suyanto, 2014).

2.1.2 Sistem Pakar (*Expert System*)

Salah satu cabang dari *Artificial Intelligence (AI)* yang sudah cukup lama adalah sistem pakar (*expert system*), mulai dikembangkan dipertengahan tahun 1960. Newell dan Simon berhasil mengembangkan dan memunculkan sistem pakar untuk pertama kali yaitu *General purpose problem solver (GPS)*. Hingga saat ini sistem pakar sudah banyak dibuat dan dikembangkan, diantaranya adalah *Prospector* digunakan untuk membantu, mencari dan menemukan deposit dalam bidang geologi, *MYCIN* untuk diagnosis penyakit, *SOPHIE* untuk analisis sirkuit elektronik, *FOLIO* digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, *DELTA*, *XCON* & *XSEL* untuk membantu sistem komputer besar dalam melakukan konfigurasi, dipakai untuk pemeliharaan lokomotif listrik disel, dan sebagainya, serta *DENDRAL* untuk mengenali struktur molekul campuran yang belum teridentifikasi. (Sutojo et al., 2011)

Sistem pakar merupakan sebuah program yang dibuat sebagai peniru pakar manusia untuk melakukan hal-hal yang dikerjakan oleh pakar. Dalam membangun sistem komponen yang harus ada yaitu sebagai berikut: (Hartati & Iswanti, 2008):

a. *User Interface (Antarmuka Pengguna)*

Sistem pakar berkemampuan menggantikan seorang pakar dengan keadaan tertentu, oleh karena itu sistem pakar harus mudah digunakan untuk pengguna yang awam dengan masalah teknis. Suatu sistem pakar harus menyediakan media komunikasi yang berfungsi sebagai penghubung antara sistem dan penggunanya atau

biasa disebut antarmuka. Antarmuka yang tersedia bagi pemakai yang tidak ahli ataupun awam dengan sistem pakar haruslah efektif dan *user friendly* (ramah pengguna).

b. *Knowledge Base* (Basis Pengetahuan)

Pengetahuan suatu bidang ilmu tertentu pada setiap tingkatan pakar dan memiliki format tertentu disebut dengan basis pengetahuan. Pengetahuan yang didapatkan dari berbagai sumber dan akumulasi pengetahuan dari seorang pakar. *Knowledge Base* memiliki sifat yang tidak tetap, hal ini dikarenakan basis pengetahuan selalu mengikuti perkembangan pengetahuan dari waktu ke waktu yang selalu bertambah dan mengikuti pengetahuan *ter-update*. Sistem pakar dengan basis pengetahuan yang terpisah dari mesin inferensinya bertujuan agar dalam melakukan pengembangan (penambahan atau pengurangan) sistem dapat dilakukan secara leluasa serta dapat mengikuti perkembangan suatu bidang ilmu pengetahuan sehingga tidak menggagu fungsi mesin inferensi.

c. *Inference Machine* (Mesin Inferensi)

Mesin inferensi adalah otak atau sumber dari sistem pakar itu sendiri, memiliki prinsip sebagai pencari solusi dalam memecahkan masalah. Mesin inferensi yaitu suatu program komputer yang melakukan penalaran mengenai informasi pengetahuan pada *working memory*, dan merumuskan kesimpulan.

d. *Explanation Facility* (Fasilitas Penjelasan)

Dalam sesi konsultasi dan kemudian melakukan penalaran layaknya seorang pakar sehingga membuat proses penentuan keputusan tersebut dilakukan oleh mesin

inferensi. Pengguna mesin inferensi umumnya adalah orang awam, sehingga dibuatlah fasilitas penjelasan. Dengan fasilitas ini dapat memberikan informasi kepada pengguna mesin inferensi, informasi penjelasan yang diberikan mengenai proses penalaran sehingga dapat menghasilkan keputusan. Penjelasan yang diberikan berupa pertanyaan kepada pengguna, yakni penjelasan pertanyaan mengapa atau pertanyaan bagaimana dalam sistem mencapai kesimpulan.

Tujuannya adanya fasilitas penjelasan dalam sistem pakar diantaranya adalah adanya proses analisa yang menunjukkan tidak kalah penting dalam memuaskan psikologi pengguna, sehingga membuat sistem menjadi lebih cerdas.

e. *Knowledge Acquisition Facility* (Fasilitas Akuisisi Pengetahuan)

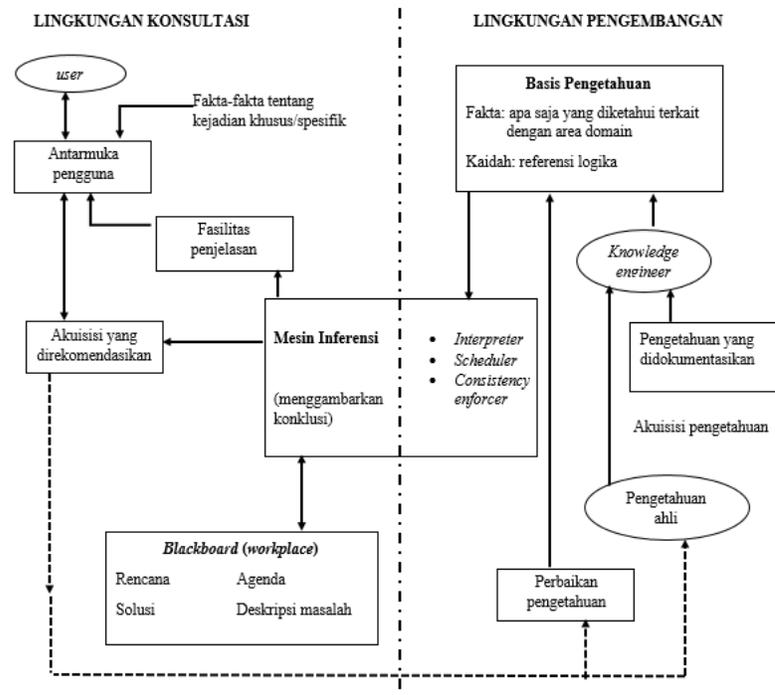
Fasilitas ini memudahkan seorang pakar dalam menambahkan pengetahuan ataupun kaidah. Pengguna biasa tidak memiliki hak akses untuk memakai fasilitas akuisisi ini

f. *Working Memory* (Memori Kerja)

Adalah salah satu bagian yang dimiliki sistem pakar berfungsi sebagai penyimpanan fakta yang didapatkan pada saat proses konsultasi dilaksanakan kemudian disimpan ke dalam basis pengetahuan. Nantinya digunakan dalam menentukan suatu keputusan masalah. Kesimpulannya dapat berupa tindakan mapupun hasil diagnosa.

Sistem pakar juga memiliki sudut pandang lingkungan (*environmwnt*). Ada dua jeni lingkungan yaitu konsultasi dan pengembangan. Konsultasi dibuat untuk pengguna biasa (bukan pakar) melakukan konsultasi dengan tujuan mendapatkan nasehat atau

hasil dari seorang pakar. Lingkungan pengembangan digunakan sebagai meng-*input* perolehan pengetahuan ke dalam basis data. Struktur lingkungan sistem pakar dapat dilihat pada gambar 2.1:



Gambar 2. 1 Struktur Sistem Pakar

Pengetahuan diolah ke dalam bentuk dan format tertentu agar mudah dipahami oleh komputer maka harus melakukan representasi pengetahuan. Pemilihan pengolahan pengetahuan dengan sesuai dapat menghasilkan suatu sistem pakar yang efektif. Salah satu bentuk penting dari pengolahan pengetahuan adalah aturan produksi (*production rule*). (Hartati & Iswanti, 2008)

Masing-masing rule yang telah dibandingkan oleh mesin inferensi tersimpan menjadi manajemen pengetahuan beserta dengan fakta yang ada pada basis data. Jika adalah *IF* (kondisi) dari *rule* cocok dan sesuai dengan fakta maka *rule* akan dieksekusi

dan *THEN* (aksi) disimpan dalam *database* sebagai penambahan fakta baru. (Sutojo et al., 2011)

2.1.2.1 Penalaran Maju (*Forward Chaining*)

Teknik pencarian yang diawali dengan fakta yang ada, lalu fakta-fakta tersebut dicocokkan dengan *rules IF-THEN* disebut dengan *forward chaining*. Jika menemukan ada fakta yang sesuai dengan *IF*, maka *rule* tersebut segera diproses. Jika *rule* yang telah selesai diproses, maka fakta baru (*THEN*) disimpan ke dalam *database*. Saat melakukan pencocokan, dimulai dari *rule* teratas. Masing-masing *rule* hanya boleh diproses sekali saja. Proses pencocokan akan berhenti apabila *rule* tidak ada lagi yang dapat diproses. Konsep ini disebut juga *data driven search* (pencarian yang dimotori oleh data). (Sutojo et al., 2011)

Penalaran runut maju melakukan penalaran terlebih dahulu dimulai dari asumsi-asumsi (premis) atau informasi masukan (*IF*) kemudian menuju kesimpulan atau *drived information (THEN)*. Dapat dituliskan sebagai berikut:

IF (premis)

THEN (kesimpulan)

Premis dapat berupa pengamatan, sedangkan kesimpulan merupakan diagnosa sehingga disebut sebagai jalan penalaran *forward chaining* dimulai dari pengamatan dan menuju ke diagnosa. Dengan metode *forward chaining*, sistem tidak melakukan praduga apapun dengan kesimpulan, namun sistem akan menerima semua diagnosis

berdasarkan gejala yang pengguna berikan kemudian sistem akan memeriksa gejala-gejala tersebut dan dicocokkan dengan kesimpulan yang paling sesuai (Sutojo et al., 2011)

2.1.2.2 Penalaran Mundur (*Backward Chaining*)

Penalaran yang diawali berdasarkan hasil dan bekerja mundur kembali ke arah kondisi awal disebut dengan metode inferensi *backward chaining*. Proses penalaran diawali dengan *THEN* dari aturan *IF-THEN* yang selanjutnya pencarian diteruskan dengan melakukan pencocokan fakta yang ada sesuai dengan premis *THEN* yang ada di *database* sebagai fakta yang baru. Apabila tidak sesuai, premis tersimpan dibagian *IF* ke dalam tumpukan (*stack*) sebagai *subGoal*. Pencocokan selesai apabila *Goal* ditemukan atau kebenaran tidak dapat dibuktikan dari *subGoal* atau *Goal*. (Sutojo et al., 2011)

2.1.2.3 Pohon Keputusan

Cara formal yang disediakan untuk merepresentasi arahan atau strategi dituliskan dalam bentuk jika-maka (*IF-THEN*). Kaidah *IF-THEN* menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Contoh dari struktur penulisan

kaidah *IF-THEN* yang menghubungkan objek adalah sebagai berikut: (Hartati & Iswanti, 2008)

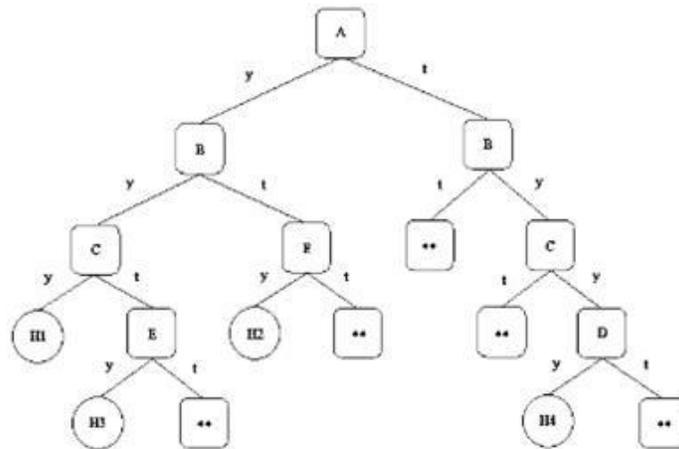
1. JIKA premis MAKA kesimpulan
2. JIKA masukan MAKA keluaran
3. JIKA kondisi MAKA tindakan
4. JIKA antesenden MAKA konsekuen
5. JIKA data MAKA hasil
6. JIKA tindakan MAKA tujuan
7. JIKA aksi MAKA reaksi
8. JIKA gejala MAKA diagnosa

Gejala mengarah kepada keadaan yang menyebabkan adanya kerusakan atau kondisi tertentu yang mendorong adanya pemeriksaan (diagnosa). Aksi berdasarkan pada tindakan yang menyebabkan timbulnya efek dari tindakan yang dibuat. Sebelum memberikan hasil yang diharapkan. Sebelum mendapatkan hasil yang diharapkan pemberian tindakan harus dilakukan dan harus berdasarkan pada kegiatan berlangsung. Data berdasarkan informasi harus tersedia sehingga sebuah hasil dapat diperoleh. Antesenden berdasarkan situasi yang terjadi sebelum konsekuensi dapat diamati. Kondisi berdasarkan dengan keadaan dan sebelum mengambil tindakan. Masukan berdasarkan pada data yang tersedia sebelum keluaran dapat diperoleh. Premis berdasarkan pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. (Hartati & Iswanti, 2008).

Pengetahuan yang telah diperoleh dari *domain* tertentu ditampilkan berupa tabel keputusan dan selanjutnya dituangkan menjadi sebuah pohon keputusan. Contoh penyajian tabel keputusan beserta pohon keputusannya dapat dilihat sebagai berikut ini: (Hartati & Iswanti, 2008)

Tabel 2. 1 Bentuk Tabel Keputusan

Hipotesis <i>Evidence</i>	Hipotesis 1	Hipotesis 2	Hipotesis 3	Hipotesis 4
<i>Evidence A</i>	Ya	Ya	Ya	Tidak
<i>Evidence B</i>	Ya	Tidak	Ya	Ya
<i>Evidence C</i>	Ya	Tidak	Tidak	Ya
<i>Evidence D</i>	Tidak	Tidak	Tidak	Ya
<i>Evidence E</i>	Tidak	Ya	Ya	Tidak



Keterangan:

A = *evidence A*,
B = *evidence B*,
C = *evidence C*,
D = *evidence D*,

H1 = hipotesis 1,
H2 = hipotesis 2,
H3 = hipotesis 3,
H4 = hipotesis 4

y = ya
t = tidak
** = tidak menghasilkan hipotesa tertentu

Gambar 2. 2 Pohon Keputusan

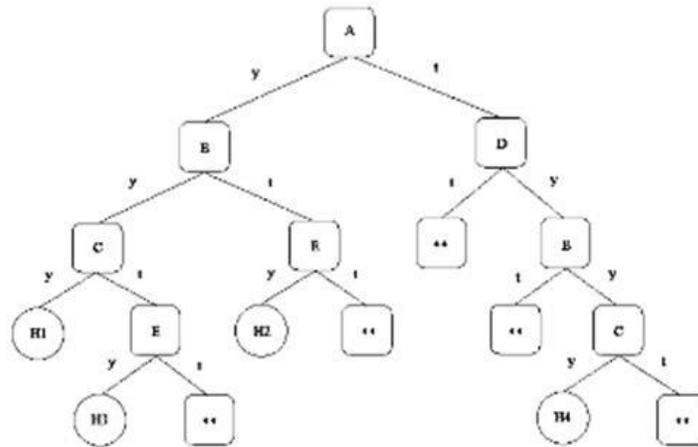
Pada gambar 2.2 dapat diketahui bahwa hipotesis H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesis H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesis H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesis H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node (evidence)* di atasnya, notasi “t” artinya tidak memenuhi (Hartati & Iswanti, 2008).

Pada saat melakukan konsultasi pada sistem pakar, pertanyaan-pertanyaan yang diajukan oleh sistem dilakukan oleh *node-node* yang mewakili *evidence*. Dengan melihat pohon keputusan pada gambar 2.2 permasalahan terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menampilkan pertanyaan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban yang diberikan oleh pengguna baik “ya” atau “tidak” maka sistem akan meneruskan ke pertanyaan *evidence* B. Jawaban pengguna ini berarti tidak akan mempengaruhi sistem. Cara yang dapat dilakukan untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan terlihat pada tabel 2.2 (Hartati & Iswanti, 2008)

Tabel 2. 2 Tabel Keputusan Alternatif

Hipotesis <i>Evidence</i>	Hipotesis 1	Hipotesis 2	Hipotesis 3	Hipotesis 4
<i>Evidence</i> A	Ya	Ya	Ya	Tidak
<i>Evidence</i> D	Tidak	Tidak	Tidak	Ya
<i>Evidence</i> B	Ya	Tidak	Ya	Ya
<i>Evidence</i> C	Ya	Tidak	Tidak	Ya
<i>Evidence</i> E	Tidak	Ya	Ya	Tidak

Dari tabel 2.2 yang dihasilkan, maka diperoleh pohon keputusan sebagai berikut:



Keterangan:

A = *evidence* A,

B = *evidence* B,

C = *evidence* C,

H1 = hipotesis 1,

H2 = hipotesis 2,

H3 = hipotesis 3,

y = ya

t = tidak

** = tidak menghasilkan hipotesa tertentu

Gambar 2. 3 Pohon Keputusan Alternatif

Setiap *node* yang mewakili *evidence* tertentu dapat dilihat dari gambar 2.3, untuk kondisi “y: dan “t” sudah tidak mengarah pada *evidence* yang sama. Maka dengan ini jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula. (Hartati & Iswanti, 2008)

Berdasarkan pohon keputusan yang ada pada gambar 2.3 maka kaidah yang dapat dihasilkan adalah sebagai berikut: (Hartati & Iswanti, 2008)

1. Kaidah 1: *IF A AND B AND C THEN H1*
2. Kaidah 2: *IF A AND B AND E THEN H3*
3. Kaidah 3: *IF A AND E THEN H2*
4. Kaidah 4: *IF D AND B AND C THEN H4*

Model representasi ini mudah dipahami dan bersifat ringkas dan jelas sesuai dengan jalan pemikiran manusia dalam penyelesaian suatu masalah sehingga model basis pengetahuan kaidah dalam produksi banyak digunakan dalam aplikasi pembuatan sistem pakar.

Sistem pakar mempunyai beberapa kelebihan yang bisa digunakan, tetapi juga mempunyai beberapa kekurangan, diantaranya adalah: (Sutojo et al., 2011)

1. Dalam membuat dan memelihara sistem pakar membutuhkan biaya yang sangat mahal.
2. Pengembangan sistem pakar sulit dilakukan karena keterbatasan keahlian yang dimiliki dan pakar yang minim.
3. Sistem pakar tidak 100% akurat dan bernilai benar

2.1.3 Web

World Wide Web (WWW) adalah suatu program ditemukan oleh tim Berner-Lee tahun 1991. Bermula dari mencari cara yang lebih efisien dalam melakukan penyusunan arsip risetnya. Maka dari itu, dilakukanlah pengembangan suatu sistem sebagai keperluan pribadi. Sistem tersebut bernama *Enquire*, suatu program yang dapat digunakan diperangkat lunak. Program tersebut berhasil menciptakan suatu jaringan yang dapat menghubungkan berbagai arsip sehingga menjadi lebih mudah dalam melakukan pencarian informasi yang dibutuhkan. Hal ini yang sekarang menjadi dasar dari perkembangan pesat dari *WWW*. (Hidayatullah & Kawistara, 2014)

Dengan *web* sangat memudahkan bagi pengguna komputer yang terhubung dengan pengguna internet lainnya dan mencari banyak informasi di *internet*. Karena beberapa alasan, *Web* diadopsi banyak perusahaan sebagai bagian dalam menyusuauni strategi teknologi informasi diantaranya akses dan pendistribusian informasi yang sangat mudah, *set-up server* lebih mudah, dapat ditampilkan oleh *web browser* di sistem operasi apapun karena *web* memiliki standar dokumen dapat digunakan diberbagai tipe data untuk ditampilkan (Sidik & Pohan, 2009)

Web merupakan layanan yang berhubungan dengan internet memiliki ruang informasi berupa *web*, dan *user* dapat mengakses *hyperlink* untuk menemukan dan mendapatkan informasi yang dibutuhkan, informasi yang diberikan berupa dokumen yang disajikan dengan *web browser*. Keuntungan dari penggunaan *web* adalah sebagai berikut: (Raharjo, 2011)

1. Penyaluran informasi menjadi sangat mudah, sehingga bisa diakses oleh seluruh pengguna di penjuru dunia.
2. Pengaturan *server* menjadi sangat mudah.
3. Cukup sekali dalam pemasangan aplikasi disetiap komputer *pengguna*.
4. Informasi yang dibutuhkan dapat dengan mudah diakses melalui komputer manapun tanpa harus melihat sistem operasi yang dimiliki.

2.1.4 Basis Data (*Database*)

Database yaitu sistem yang memiliki tujuan utama dalam pemeliharaan data dan informasi yang terkomputerisasi serta telah diolah ketika informasi tersebut dibutuhkan. *Database* digunakan sebagai media penyimpanan data agar dapat diakses dengan cepat dan mudah. Kebutuhan basis data diantaranya adalah menyimpan, memasukkan, mengambil data dan menyusun laporan yang mengacu pada data tersimpan sebelumnya. *Database Management System (DBMS)* merupakan salah satu bentuk basis data yang dibutuhkan dalam sebuah sistem. *DBMS* digunakan sebagai media penyimpanan, mengolah, dan memunculkan data. Syarat yang harus dimiliki *DBMS* adalah: (S. & Shalahuddin, 2013):

1. Tersedianya kemudahan yang dapat digunakan dalam mengelola akses data
2. Dapat menangani keutuhan data
3. Dapat menangani data yang diakses secara serentak dalam satu waktu
4. Dapat melakukan pencadangan data

Beberapa jenis *DBMS* yang saat ini paling banyak digunakan: (S. & Shalahuddin, 2013)

1. *DBMS* berbayar atau komersial, yaitu *Microsoft SQL Server, Oracle, Microsoft Access, dan IBM DB2,*
2. *DBMS* dengan sumber terbuka yaitu *MySQL, SQLite, Postgre SQL, dan Firebird*

Terdapat beberapa tahapan yang dinamakan *physical database design* dalam alur hidup basis data (*Database Life Cycle*). Tahap ini dimulai dengan membuat rancangan

fisik *database* yaitu *Physical Data Model* (PDM). *PDM* merupakan pemodelan menggunakan tabel dalam menggambarkan dan menghubungkan antar data. Tabel memiliki beberapa kolom dengan nama unik dan tipe data yang digunakan dalam *database*. *PDM* adalah konsep detail yang kegunaannya menjelaskan cara penyimpanan data ke dalam *database*. *PDM* merupakan bentuk nyata dari perancangan *database* yang akan diolah dengan *DBMS* (S. & Shalahuddin, 2013)

2.1.5 Validasi Sistem

Validasi berpacu dengan beberapa Aktivitas berbeda yang menyatakan pembuatan perangkat lunak atau aplikasi telah dibangun dan sesuai dengan harapan. Berikut adalah beberapa pendekatan yang dilakukan dalam pengujian validasi sistem (S. & Shalahuddin, 2013):

1. *Black-Box Testing* (Pengujian Kotak Hitam)

Pendekatan *black box testing* dilakukan dengan cara perangkat lunak atau sistem diuji melalui segi kegunaan tanpa pengujian rancangan ataupun kode-kode pembuatan program. Tujuannya adalah memastikan bahwa fungsi dari *input* dan *output* dari aplikasi atau sistem tersebut sudah sesuai. Pengujian kasus benar dan salah merupakan salah satu cara uji yang wajib dilakukan pada *black box testing*.

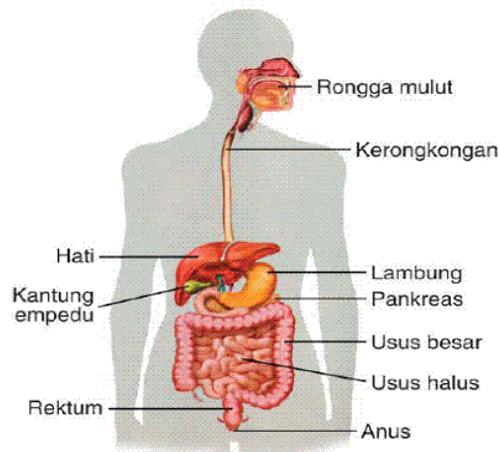
2. *White-Box Testing* (pengujian kotak putih)

White box testing dilakukan dengan melakukan pengujian perangkat lunak dari berdasarkan bait kode dan desain program apakah telah menghasilkan suatu program atau aplikasi yang sesuai dengan logika yang ada didalamnya. Cara melakukan pengujian adalah dengan memeriksa kode dan logika program. Kasus uji dapat dibuat sesuai standar pengujian program yang ada.

2.2 Variabel Penelitian

Variabel penelitian dapat berupa bermacam hal yang kemudian dipilih oleh peneliti sebagai bahan yang diolah dan dipelajari untuk memperoleh informasi untuk disimpulkan. Karena memiliki variatif oleh karena itu dinamakan dengan variabel. (Sugiyono, 2014). Variabel pada penelitian ini adalah penyakit saluran pencernaan, dimana jenis penyakit tersebut cukup banyak.

Pencernaan adalah proses ketika makanan dihancurkan dalam kanal pencernaan menjadi zat yang dapat diserap oleh tubuh. Dengan katalain “pencernaan” adalah istilah yang menggambarkan bagaimana tubuh kita mengubah makanan menjadi bahan kimia, sehingga bahan-bahan itu dapat mengisi, membangun, dan memperbaiki jaringan tubuh kita (Atkins, 2016)



Gambar 2. 4 Saluran Pencernaan Manusia

Pencernaan manusia memiliki dua jenis dalam memproses zat yang masuk, yaitu (Shanty, 2011):

1. Makanan dengan bentuk yang kasar diolah menggunakan gigi (mengunyah) agar menjadi bentuk yang halus, proses ini dinamakan dengan pencernaan mekanik.
2. Zat makanan yang telah diolah melalui pencernaan mekanik, kemudian disederhanakan lagi dengan enzim yang diproses oleh mulut, diteruskan ke lambung, dan diolah oleh usus, proses tersebut dinamakan dengan pencernaan kimiawi.

Mulut, tenggorokan (*faring*), kerongkongan, lambung, usus halus, usus besar, *rectum*, dan anus adalah *anatomi* pencernaan manusia. Sistem pencernaan manusia meliputi juga organ-organ yang berada di luar saluran pencernaan, yaitu kantung empedu, hati dan *pankreas* (Shanty, 2011). Dengan indikator-indikator sebagai berikut:

1. Diare

Diare adalah penyakit yang menyerang proses pencernaan yakni terjadinya peningkatan frekuensi dalam buang air besar biasanya terjadi secara berulang-ulang virus (lebih dari 50%) dan kuman yang menyerang usus atau bakteri *shigella*, *salmonella*, dan *escherichia coli*. Saat terserang diare, Aktivitas yang dilakukan oleh usus menjadi meningkat sehingga ingin selalu buang air besar. Penumpukan carian di usus mengakibatkan mekanisme penyerapan air menjadi terganggu. (Saydam, 2011)

Gejala-gejala yang bisa timbul dalam penyakit diare ini adalah sebagai berikut (Saydam, 2011):

1. Perut terasa sakit dan mulas
2. Perut terasa kembung
3. Tinja sedikit berlendir
4. Badan terasa lemas
5. Saat ingin buang air, sekitaran anus menjadi panas.

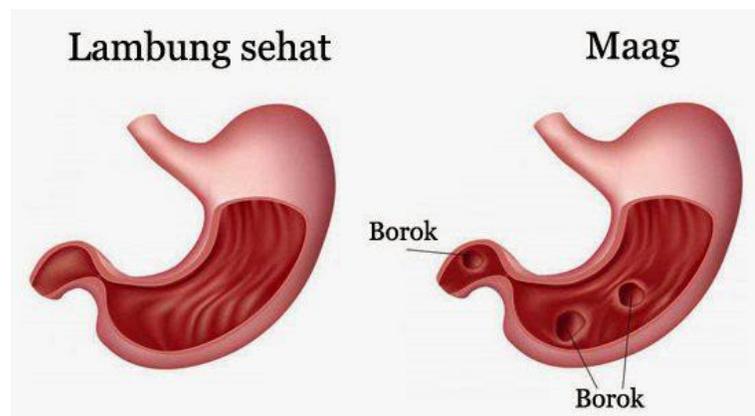
Jika tidak diberikan penanganan atau perawatan bagi penderita, diare akan menjadi penyakit yang sangat membahayakan karena bisa menimbulkan hilangnya nyawa. Penyebabnya adalah terlalu banyak mineral dan kandungan cairan pada tubuh yang dikeluarkan sehingga bisa memicu dehidrasi. Dalam menggantikan cairan yang hilang, penderita harus selalu diberikan air putih yang cukup. (Saydam, 2011).

Untuk menolong penderita diare, maka harus segera diberikan solusi atau penanganan dini diantaranya adalah (Saydam, 2011):

1. Mengganti cairan yang keluar dengan memberikan air putih secara cukup kepada penderita.
2. Memberikan larutan oralit pada penderita
3. Untuk menghindari terserang diare maka lakukan pencegahan berikut:
 - a. Tangan harus selalu bersih pada saat sebelum dan sesudah makan, setelah buang air besar ataupun setelah melakukan aktivitas lainnya, yaitu dengan cara tangan selalu dicuci dengan sabun dan air bersih.
 - b. Makanan selalu disimpan ditempat yang aman dari debu dan jangkauan lalat.

2. Tukak Lambung (*Maag*)

Dinding lambung yang rusak dikarenakan karena selimut lambung (*mucus*) telah rusak menjadi penyebab dari penyakit *maag*. *Mucus* menghasilkan enzim yang memakan permukaan lambung. Yang ditimbulkan dari penyakit ini membuat dinding lambung menjadi berlubang dan isinya jatuh ke dalam rongga perut. (Shanty, 2011).



Gambar 2. 5 Lambung Manusia

Penyakit tukak lambung disebabkan oleh gangguan lambung yang dikarenakan oleh pola makan yang berantakan dan tidak sesuai dengan pola umum orang Indonesia kebanyakan dan dapat juga disebabkan oleh kelainan gangguan pada organ lambung (Saydam, 2011).

Gejala-gejala yang muncul pada penderita tukak lambung ini yaitu (Saydam, 2011):

1. Timbulnya rasa sakit kepala (pusing)
2. Perut terasa kembung
3. Rasa mual dan ingin muntah
4. Sering bersendawa dan buang gas (kentut)
5. Nyeri lambung yang berulang

Penanganan dan pencegahan dini yang dapat dilakukan pada saat terserang penyakit tukak lambung yaitu sebagai berikut (Saydam, 2011):

1. Pengaturan gaya hidup dan pola makan yang sehat
2. Memberikan obat yang dapat meredakan asam lambung sekaligus membunuh kuman pada lambung yang terinfeksi.
3. Mengonsumsi obat antasida atau tetral asam lambung sesuai dengan keterangan dosis tertera untuk mengatasi serangan nyeri lambung.

Supaya terhindar dari *maag* yang menyerang penderita dapat melakukan pengaturan pola makan (d disesuaikan dengan kebutuhan dan jenis makanan yang beragam) dengan jadwal yang teratur, lambung tidak dalam keadaan kosong yang

cukup lama karena asam berlebih pada lambung akan mudah naik dan pastinya akan merusak dinding lambung (Saydam, 2011).

3. Tipus (Demam *Typhoid*)

Tipus adalah penyakit yang terdapat pada pencernaan manusia, merupakan infeksi akut pada usus. Yang dimaksudkan akut disini adalah seluruh tubuh akan turut menderita bila seseorang terserang penyakit tipus ini (Saydam, 2011)

Demam tipus disebabkan oleh bakteri *salmonella typhi* dari makanan yang terinfeksi dan air yang dikonsumsi. Demam tipus terdiagnosa ketika tinja terdeteksi bakteri *salmonella*. Penyakit tipus ini berdurasi sekitar empat hingga enam minggu. Biasanya penderita tipus mengalami gejala-gejala berikut: (Shanty, 2011).

1. Timbulnya rasa sakit kepala (pusing)
2. Nafsu makan menurun
3. Badan terasa lemas
4. Lidah tampak berselaput putih susu dan kemerahan dibagian tepi.
5. Demam yang muncul pada sore dan malam hari

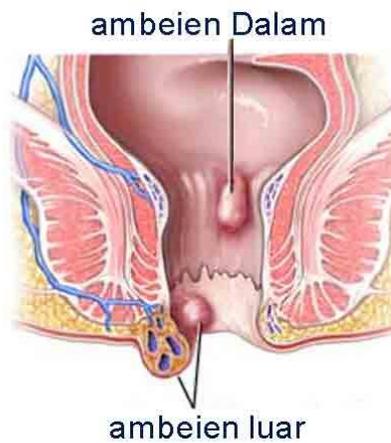
Obat yang dipakai untuk penyakit tipus ini adalah *chloromycetin*. Obat ini dapat membantu menurunkan suhu tubuh, memperpendek lama penyakit tipus, dan mencegah timbulnya gejala-gejala penyakit setelah sembuh. (Saydam, 2011).

Pencegahan dini agar terhindar dari penyakit tipus ini adalah dengan memperhatikan kebersihan dan tempat tinggal, memperhatikan pola makan, kebersihan

dalam memasak makanan, tidur cukup dan selalu menjaga stamina tubuh (Saydam, 2011).

4. Wasir (*Ambeien*)

Wasir atau *ambeien* merupakan pembuluh darah yang berada di bawah selaput lendir anus melebar dan kemudian menjadi benjolan. Penyakit wasir cukup membuat repot si penderita, karena benjolan tersebut muncul di dalam bagian tubuh yang sedikit rumit (Saydam, 2011)



Gambar 2. 6 Penyakit Wasir

Aliran darah menuju jantung dari sekitaran anus terhalang, dan mengakibatkan pembuluh darah menyebar sehingga menimbulkan *ambeien*. Pada daerah anus memiliki otot lingkar yang selama 24 jam selalu mencengkram, dan pada saat anus akan mengeluarkan kotoran otot tersebut akan berelaksasi. (Saydam, 2011).

Faktor lain yang meningkatkan kemungkinan terserang wasir adalah sembelit berkepanjangan, sering diare, mengejan untuk buang air besar, mengalami obesitas,

sering mengangkat benda berat, hamil (akibat peningkatan tekanan di dalam perut karena bayi yang tumbuh), dan riwayat keluarga. Adapun gejala-gejala dari penyakit wasir ini adalah sebagai berikut (Atkins, 2016):

1. Pendarahan dengan rasa gatal di bagian anus
2. Terkadang terasa nyeri pada saat ingin buang air besar
3. Munculnya benjolan disekitar anus

Pengobatan yang dapat diberikan untuk penderita wasir ini adalah sebagai berikut (Atkins, 2016):

- 1) Banyak mengkonsumsi buah dan serat yang cukup agar terhindar dari sembelit dan mengejan
- 2) Mengkonsumsi obat-obat pereda wasir dari apotek seperti *laktulosa* akan menjaga feses tetap lunak.
- 3) Mengkonsumsi *parasetamol* sesuai anjuran dosis untuk mengurangi rasa ketidak nyamanan.

Untuk mencegahnya penderita dapat melakukan perubahan pola makan dan pola hidup. Memilih makanan yang harus mengikutsertakan sayur dan buah-buahan, menghindari makanan cepat saji, serta olahraga yang teratur yang sesuai dengan daya kemampuan (Saydam, 2011).

4. Sindrom Iritasi Usus

Sindrom iritasi usus atau disebut juga *Irritable Bowel Syndrome* (IBS) adalah gangguan dalam jangka panjang pada sistem pencernaan; penyakit yang dapat terjadi berulang kali, yang lebih sering menyerang wanita daripada laki-laki, dan dapat dialami oleh segala usia, termasuk anak-anak. Adapun gejala yang sering dialami dengan gangguan tersebut yaitu (Atkins, 2016)

1. Nyeri dibagian perut
2. Perut terasa kembung
3. Perut terasa keram
4. Nyeri punggung
5. Perasaan lelah sepanjang waktu

Penanganan dan pengobatan sederhana dalam mengatasi sindrom iritasi usu ini adalah sebagai berikut (Atkins, 2016)(Atkins, 2016)

- 1) Melakukan diet sederhana, makan secara teratur, mengurangi mengkonsumsi minuman berkafein dan bersoda, minum air putih secukupnya (disarankan 1,5-2 liter setiap hari), mengkonsumsi makanan dengan serat yang cukup
- 2) Pemberian obat pada sindrom iritasi usus ini adalah *Lopiramede* (Imodium), obat *antidepresan*, simetikon (*Wund-eze*)

2.3 Software Pendukung

Software pendukung adalah aplikasi yang mendukung dan membantu dalam pembuatan sistem pakar pada penelitian ini. Aplikasi yang digunakan adalah *Star UML*, *HTML5*, *CSS3*, *PHP*, *MySQL*, *XAMPP*, *PHPMyAdmin*, *Notepad++*, dan *Bootstrap*.

2.3.1 Star UML

Star UML adalah perangkat pembantu yang telah berbasis komputer dalam merekayasa dan mendukung alur hidup perangkat lunak (*life cycle support*) sering juga disebut dengan *CASE (Computer Aided Software Engineering) tools*. *StarUML* dikelompokkan pada *upper CASE tools* yang mendukung perencanaan dan dalam pembangunan perangkat lunak (S. & Shalahuddin, 2013)



Gambar 2. 7 Logo *StarUML*

Pemodelan yang paling banyak digunakan saat ini salah satunya adalah *UML (Unified Modeling Language)*. *UML* menjadi bahasa yang standar digunakan oleh dunia industri untuk mendefinisikan kebutuhan, memuat analisis, desain dan menggambarkan struktur pemrograman yang berorientasi pada objek. (S. & Shalahuddin, 2013)

Banyaknya kebutuhan pemodelan bentuk visual dalam membuat spesifikasi, membuat gambaran, membangun dan mengarsipkan segala bentuk dari sistem perangkat lunak, sehingga *UML* ada dan digunakan oleh banyak kalangan. *UML* tidak memberikan batas bagi jenis pemrograman tertentu, meskipun faktanya *UML* paling banyak digunakan pada jenis pemrograman yang berorientasi pada objek. *UML* memiliki 13 macam diagram yang terbagi menjadi 3 kategori yaitu: (S. & Shalahuddin, 2013)

1. *Structure diagrams* yaitu diagram-diagram yang digunakan untuk menggambarkan suatu alur kerja perangkat lunak.
2. *Behavior diagram* merupakan salah satu kategori diagram yang digunakan untuk menggambarkan penentu sifat dari sistem atau perubahan-perubahan yang terangkai pada sebuah sistem.
3. *Interaction diagram* yaitu diagram-diagram yang digunakan untuk menggambarkan aspek dinamis dalam *UML* dan menggambarkan interaksi atau komunikasi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

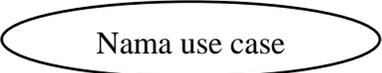
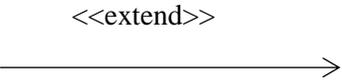
Berikut diagram yang digunakan dalam membuat rancangan desain sistem pakar yaitu:

1. *Use case diagram*

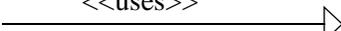
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah

interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu *use case* dan aktor Berikut ini adalah simbol-simbol yang digunakan pada *use case diagram* (S. & Shalahuddin, 2013):

Tabel 2. 3 Simbol-Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Kegunaan yang disediakan oleh sistem sebagai unit yang saling bertukar pesan antar unit atau aktor; biasanya dituliskan dengan memberi nama sesuai dengan <i>goal</i> yang dibutuhkan oleh sistem</p>
<p>Aktor/<i>actor</i></p> 	<p>Menggambarkan orang, peralatan sistem yang berinteraksi dengan aplikasi yang sedang di bangun. Biasanya dinyatakan dengan menggunakan kata benda.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Penghubung komunikasi antar aktor dan <i>use case</i> yang menjadi partisipan atau yang memiliki interaksi dengan aktor <i>usecase</i>.</p>
<p>Ekstensi/<i>extend</i></p> 	<p><i>Use case</i> yang memiliki hubungan umum-khusus antara 2 <i>use case</i>, dimana salah satu <i>usecase</i> memiliki fungsi lebih umum dari fungsi lainnya. Generalisasi arah panah mengarah pada <i>use case</i>.</p>

Tabel 2. 3 Lanjutan

<p>Menggunakan/<i>include/uses</i></p> <p style="text-align: center;"><<include>></p>  <p style="text-align: center;"><<uses>></p> 	<p>Relasi tambahan dimana <i>use case</i> memerlukan relasi ini sebagai syarat dijalankannya <i>use case</i>. Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>
---	---

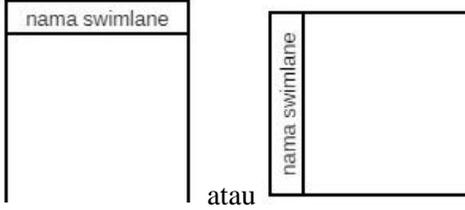
2. *Activity Diagram*

Activity diagram atau diagram Aktivitas merupakan gambaran aliran kerja (*workflow*) atau kegiatan dari suatu sistem yang memproses menu pada perangkat lunak. *Activity diagram* menggambarkan Aktivitas pada sistem bukan apa yang dilakukan oleh aktor. *Activity diagram* memiliki simbol-simbol yang digunakan pada antara lain adalah sebagai berikut:

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Deskripsi
<p>Status awal</p> 	<p>Suatu status awal dalam Aktivitas sistem sebuah diagram Aktivitas memiliki sebuah status awal</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dinyatakan dengan kata kerja. Aktivitas yang dilakukan sistem</p>
<p>Percabangan/<i>decision</i></p> 	<p>Penghubung jika Aktivitas memiliki pilihan lebih dari satu.</p>
<p>Penggabungan/<i>join</i></p> 	<p>Penghubung untuk menggabungkan Aktivitas lebih dari satu.</p>

Tabel 2. 4 Lanjutan

Status akhir 	Status akhir diagram Aktivitas yang dilakukan oleh sistem.
<i>Swimlane</i> 	Sebagai pemisah organisasi terhadap Aktivitas yang terjadi.

2.3.2 HTML (*Hyper Text Markup Language*)

Hyper Text Markup Language atau yang disingkat *HTML* merupakan bahasa penting dan paling dasar dalam menampilkan atau mengelola tampilan halaman *website*. Sebuah penjelajah *web internet* menggunakan *HTML* untuk menampilkan berbagai informasi dengan format *hypertext* sederhana yang dituangkan dalam format berkas *ASCII* sehingga menghasilkan wujud yang terintegrasi .(Saputra, 2012)



Gambar 2. 8 Logo *HTML5*

Struktur dokumen *HTML* memiliki aturan dalam pembuatannya. Elemen wajib *file HTML* dalam membangun kerangka *website* adalah sebagai berikut: (Saputra, 2012)

1. Elemen *HTML*

Merupakan dasar dalam mengawali membuat dokumen *HTML*. Tanda ini berfungsi sebagai pendefinisian bahwa ini merupakan dokumen *HTML*. Contoh penulisannya adalah:

`<html>` kemudian diakhiri dengan `</html>` pada akhir dokumen.

2. Elemen *Head*

Tag elemen *Head* berfungsi sebagai penulisan keterangan dokumen *web* yang akan ditampilkan. Penutupan elemen ini diakhiri dengan tanda penutup `</head>`.

Secara urut dituliskan dengan format berikut:

```
<html>
<head>
</head>
</html>
```

3. Elemen *Title*

Pemberian informasi judul pada *caption browser web* mengenai topik atau tema dari dokumen *web* yang nantinya akan tampil pada *browser* yaitu menggunakan elemen *title*. Berikut ini adalah struktur penggunaan elemen *title*:

```
<html>
<head>
  <title> ..JUDUL WEB.. </title>
</head>
</html>
```

4. Elemen *Body*

Bagian ini merupakan bagian utama dalam *web*. Elemen *body* berfungsi sebagai tanda untuk menampilkan teks atau informasi atau yang sering disebut konten, dan diletakkan di elemen *body*. Struktur pembuatan elemen *body* dapat dilihat di bawah ini:

```
<html>  
<head>  
    <tittle> ..JUDUL WEB </tittle>  
</head>  
<body>  
    ..isi konten web  
</body>  
</html>
```

2.3.3 CSS3 (*Cascading Style Sheet*)

Cascading Style Sheet (CSS) adalah suatu bahasa yang bekerja sama dengan dokumen *HTML* untuk mendefinisikan cara bagaimana suatu isi halaman *web* yang ditampilkan atau dipresentasikan. Presentasi ini meliputi *style* atau gaya teks, *link*, maupun tata letak (*layout*) halaman (Raharjo, 2011).



Gambar 2. 9 Logo CSS3

2.3.4 *PHP (Hypertext Preprocessor)*

Kepanjangan dari *PHP* yaitu *Hypertext Preprocessor* yang berupa bahasa pemrograman untuk ditambahkan pada bahasa *HTML*. Penggunaan *PHP* dipakai dalam pembuatan program *web* yang dinamis serta membuat *Content Management System (CMS)* (Madcoms, 2016).



Gambar 2. 10 Logo *Hypertext Processor (PHP)*

Saat ini *PHP* telah menjadi pemrograman *web* umum yang banyak dipakai oleh pengembang. Mempunyai banyak kelebihan menjadi alasan utama kenapa *PHP* lebih dipilih sebagai basis umum dalam membuat sebuah *web*. (Hidayatullah & Kawistara, 2014)

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Disebut bahasa pemrograman *server-side* karena *PHP* diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client)*. *PHP* dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. *PHP* dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi *GNU General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*. (Madcoms, 2016)

PHP dapat melakukan tugas-tugas yang dilakukan dengan mekanisme *CGI* (*Common Gateway Interface*) seperti mengambil, mengumpulkan data dari *database*, meng-*generate* halaman dinamis, atau bahkan menerima dan mengirim *cookie*. (Hidayatullah & Kawistara, 2014)

Berikut ini adalah beberapa kelebihan lain dari *PHP* (Madcoms, 2016):

1. Bisa membuat *web* menjadi dinamis.
2. *PHP* bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.
3. Program yang dibuat dengan *PHP* bisa dijalankan oleh semua Sistem Operasi (*OS*) karena *PHP* berjalan secara *web base* yang artinya semua Sistem Operasi bahkan *handphone* yang mempunyai *web browser* dapat menggunakan program *PHP*.
4. Aplikasi *PHP* lebih cepat dibandingkan dengan *ASP* maupun *Java*.
5. Mendukung banyak paket *database* seperti *MySQL*, *Oracle*, *PostgrSQL*, dan lain-lain.
6. Bahasa pemrograman *PHP* tidak memerlukan kompilasi (*compile*) dalam penggunaannya.
7. Banyak *web server* yang mendukung *PHP* seperti *Apache*, *Lifhttpd*, *IIS*, dan lain-lain.
8. Pengembangan aplikasi *PHP* tidak memerlukan banyak dokumentasi, referensi, dan *developer* yang membantu dalam pengembangannya.

9. Banyak bertebaran aplikasi dan program *PHP* yang gratis dan siap pakai seperti *WordPress*, *PrestaShop* dan lain-lain.

2.3.5 *PHPMysqlAdmin dan MySQL*

Sebuah aplikasi *open source* yang memiliki fungsi sebagai mempermudah dalam manajemen *MySQL*, membuat *database* dengan membuat tabel, *insert*, hapus, serta *update* data menggunakan *GUI* tanpa harus melakukan pengetikan secara manual seperti perintah *SQL* merupakan kegunaan dari *PHPMysqlAdmin*. (Madcoms, 2016).



Gambar 2. 11 Logo *PHPMysqlAdmin*

MySQL adalah sistem pengaturan *database SQL (Structure Query Language)* yang memiliki sifat *open source* dan sangat populer sekarang ini. Sistem *database MySQL* didukung dengan kegunaan yang dimiliki seperti *multithreaded*, *multi-user*, dan *SQL database management system (DBMS)*. Penggunaan *database* ini ditujukan untuk kebutuhan sistem yang mudah digunakan, handal dan cepat. (Madcoms, 2016).



Gambar 2. 12 Logo *MySQL*

Versi awal *MySQL* pertamakalinya ditemukan oleh Ulf Micheal Widenius, yang berikutnya dikembangkan oleh beberapa perusahaan yakni *MySQL AB*. *MySQL AB* yang merupakan sebuah perusahaan komersial yang didirikan oleh pengembang *MySQL*. Tipe data numerik *MySQL* menggunakan standar ANSI (*American National Standar Institue*). Di bawah ini adalah tipe data yang digunakan dalam *MySQL* (Madcoms, 2016):

1. Di bawah ini adalah tipe data numerik yang sering digunakan dan penjabarannya:

Tabel 2. 5 Tipe Data Numerik *MySQL*

Tipe Data	Keterangan
INT	Nilai integer yang menyimpan bilangan bulat positif dan negatif. Dengan rentang yang diperbolehkan adalah -2147483648 sampai 2147483647
TINYINT	Nilai integer yang sangat kecil positif dan negatif. Rentang bilangan 128 – 127 untuk yang bertanda dan 0 – 255 untuk rentang bilangan yang tidak bertanda.
SMALLINT	Nilai integer yang sangat kecil dengan jangkauan 31768 sampai 32767 untuk yang bertanda sedangkan untuk yang tidak bertanda dari 0 – 65535
MEDIUMINT	Integer dengan ukuran sedang bilangan bulat positif dan negatif dengan jangkauan -8388608 hingga 8388607 atau 0 hingga 16777215
BIGINT	Integer dengan ukuran besar 8 <i>byte</i> dengan rentang -9223372036854775808 sampai dengan 9223372036854775807 dan 0 sampai dengan 18446744073709551615

Tabel 2. 5 Lanjutan

FLOAT(M,D)	Digunakan untuk menyimpan bilangan pecahan dengan panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimalnya sampai dengan 24 digit. Defaultnya Float (10,2). Bilangan float selalu bisa bertanda
DOUBLE(M,D)	Adalah bilangan pecahan yang memiliki presisi dua kali lipat. Panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimalnya hingga 53 digit. Defaultnya Double (16,4). Bilangan float selalu bertanda. Sinonim dari DOUBLE adalah REAL
DECIMAL(M,D)	Adalah bilangan pecahan yang harus didefinisikan M dan D-nya. Membutuhkan tempat 1 byte untuk setiap desimal. Sinonim dari DECIMAL adalah NUMERIC

2. Di bawah ini adalah tipe data tanggal dan waktu yang digunakan di dalam *MySQL*

Tabel 2. 6 Tipe Data Tanggal dan Waktu *MySQL*

Tipe Data	Keterangan
DATE	Adalah tipe data berupa tanggal yang memiliki format YYYY-MM-DD, dengan jangkauan 1000-01-01 dan 9999-12-31.
DATETIME	Tipe data kombinasi antara tanggal dan waktu dengan penulisan format YYYY-MM-DD HH:MM:SS dan jangkauan data antara 1000-01-01 00:00:00 sampai dengan 9999-12-31 23:59:59
TIMESTAMP	Penanda waktu pada 1 Januari 1970 saat tengah malam hingga tahun 2037. Formatnya mirip dengan DATETIME tetapi tanpa pembatas di antara angkanya
TIME	Tipe data yang menunjukkan waktu dan disimpan dalam format HH:MM:SS
YEAR(M)	Tipe data tahun dalam format 2 atau 4 digit. Jika M diisi dengan nilai 2, maka rentang tahunnya dari 1970-2069 sedangkan jika M diisi dengan nilai 4 maka YEAR bisa bernilai 1901 sampai dengan 2155. Default nilai M adalah 4.

3. Berikut ini adalah tipe data string yang paling umum di dalam *MySQL*

Tabel 2. 7 Tipe Data String *MySQL*

Tipe Data	Keterangan
CHAR (M)	Data string yang memiliki ukuran tetap. Rentangnya adalah 1 sampai 255 karakter. Ukuran ditentukan dengan nilai M. Contoh: CHAR(6)

Tabel 2. 7 Lanjutan

VARCHAR (M)	String yang memiliki ukuran bervariasi antara 1 sampai dengan 255 karakter. Contoh: VARCHAR(25)
TEXT	String dengan ukuran maksimum 65535 karakter. String yang tersimpan di dalam TEXT dianggap tidak case sensitive. Untuk kapasitas yang lebih kecil bisa menggunakan TINYTEXT dengan kapasitas maksimal 255 karakter, sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMTEXT (maksimal 16777215 karakter) dan LONGTEXT (maksimal 4294967295 karakter)
BLOB	<i>Binary Large Object</i> (BLOB) merupakan tipe data untuk menyimpan data dan binary dalam jumlah besar. Biasa digunakan untuk menyimpan citra. Untuk penyimpanan data yang lebih kecil bisa menggunakan TINYBLOB (maksimal 255 karakter) sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMBLOB (maksimal 16777215 karakter) dan LONGBLOB (maksimal 4294967295 karakter)
ENUM	Enumerasi atau sebuah list (daftar). Contoh: ada sebuah nilai yang dibatasi hanya boleh dengan nilai tertentu saja. Misalnya saja nilai itu hanya bisa terdiri dari A-E, maka bisa dibuat menjadi ENUM('A', 'B', 'C', 'D', 'E')

3.2.1 XAMPP (X Apache MySQL PHP Perl)

XAMPP (X Apache MySQL PHP Perl) adalah sebuah paket kumpulan *software* yang terdiri dari *Apache*, *MySQL*, *PHPMysqlAdmin*, *PHP*, *Perl*, *Filezilla*, dan lain-lain. *XAMPP* berfungsi untuk memudahkan instalasi lingkungan *PHP*, dimana biasanya lingkungan pengembangan web memerlukan *PHP*, *Apache*, *MySQL*, dan *PHPMysqlAdmin* serta software-software yang terkait dengan pengembangan *web*



Gambar 2. 13 Logo *XAMPP*

3.2.2 *Notepad++*

Notepad++ merupakan suatu perangkat lunak untuk mengedit *text* yang sangat memiliki nilai guna bagi setiap orang dan khususnya bagi para pengembang sistem dalam membuat program. Komponen yang digunakan oleh *Notepad++* yaitu *Scintilla*, *Scintilla* dapat memunculkan dan menyunting teks dan berkas kode sumber dari banyak jenis bahasa pemrograman yang ada dalam *Microsoft windows*(Madcoms, 2016).



Gambar 2. 14 Logo *Notepad++*

Notepad++ mendukung beberapa bahasa pemrograman antara lain adalah *C*, *C++*, *C#*, *Java*, *HTML*, *PHP*, *Javascript*, *XML*, dan masih banyak lagi. *Notepad++* ini juga memiliki banyak fungsi dalam membedakan sintak dengan fitur *highlighting*. Kegunaan dari fitur tersebut sebagai pemberitanda *source code* yang memiliki variabel. Tersedianya fitur *tab* yang dapat membantu mengelola beberapa kode dalam waktu bersamaan. Juga ada fitur penomoran baris turut membantu dalam mencari kode yang

error ketika dijalankan. *Function List* juga disediakan untuk mengelola fungsi-fungsi yang dikembangkan. *Notepad++* ini juga disertai fitur pencarian yang sangat mudah dan praktis. Suatu kata kunci yang dimasukkan, maka *Notepad++* bisa menandai semuanya dalam *source code*. Serta, bisa mengubah semuanya hanya dengan sekali perintah karena ada fitur “*Replace All*” (Madcoms, 2016).

3.2.3 Bootstrap

Bootstrap awalnya dibuat oleh seorang desainer dan pengembang Twitter. *Bootstrap* telah menjadi salah satu *front end framework* dan proyek *open source* terkenal di dunia. *Bootstrap* diciptakan pada Twitter di pertengahan 2010 oleh Mark Otto dan Jacob Thornton. Sebelum menjadi *open sourced framework*, *Bootstrap* dikenal sebagai Twitter Blue Print. Beberapa bulan dalam pengembangan, Twitter pertamakalinya menyelenggarakan *Hack Week* dan proyek ini meledak sebagai proyek pengembangan dari semua tingkat keahlian tanpa ada bimbingan eksternal. Dan menyediakan panduan gaya untuk alat pengembangan internal di perusahaan selama lebih dari satu tahun sebelum rilis publik dan terus melakukannya hingga hari ini. (Otto & Jacob Thornton, 2011)



Gambar 2. 15 Logo *Bootstrap*

Awal dirilisnya pada Jumat, 19 Agustus 2011, setelah kami merilis 20 lebih, termasuk dua penulisan utama dengan versi 2 dan versi 3. Dengan *Bootstrap 2*, kami menambahkan fungsi responsif ke seluruh *framework* sebagai *stylesheet* opsional. Membangun dengan *Bootstrap 3*, kita menuliskan ulang untuk membuat tampilan responsif secara default dengan ponsel. (Otto & Jacob Thornton, 2011)

2.4 Penelitian Terdahulu

Untuk mendukung teori yang memiliki kaitan dengan penelitian, berikut ini adalah penelitian terdahulu yang digunakan:

Yasidah Nur Istiqomah dan Abdul Fadlil (2013), Sistem Pakar untuk Mendiagnosa Penyakit Saluran Pencernaan Menggunakan Metode *Dempster Shafer*. Subjek dalam penelitian ini adalah sistem pakar untuk mendiagnosa penyakit saluran pencernaan dengan metode yang digunakan yakni *dempster shafer* . sistem dirancang dan kemudian diimplementasi menggunakan bahasa Visual Basic 6.0 dan database

menggunakan *Microsoft Access*. Pengujian sistem dilakukan dengan *black box testing* dan *alpha test*. Sebanyak 19 jenis penyakit menggunakan metode tersebut memberikan nilai kepastian berbentuk presentase yang tampil pada hasil diagnosa penyakitnya. (Istiqomah & Fadlil, 2013)

Anugerah Jaya Aziz Amrullah dan Ekojono (2013), Sistem Pakar Diagnosa Penyakit Paru-paru Dengan Metode *Forward Chaining*. Penyakit paru-paru merupakan penyakit yang tidak mudah untuk di sembuhkan, dapat menjadi buruk apabila tidak segera ditangani dengan serius. Keterbatasan jumlah pakar atau ahli paru-paru tidak dapat mengatasi permasalahan para penderita penyakit paru-paru, makadari itu diperlukan sebuah sistem yang mana sistem tersebut dapat membantu kerja seorang pakar. Pada sistem pakar diagnosa penyakit paru-paru ini menggunakan metode *forward chaining* untuk pencarian fakta dan metode *certainty factor* untuk perhitungan tingkat kepercayaannya. Sistem pakar ini diimplementasikan dalam bentuk *website*, yang bertujuan untuk memudahkan para pengguna mencari informasi atau mendiagnosa penyakit paru- parunya. Proses pengujian sistem pakar diagnosa penyakit paru-paru adalah dengan membandingkan perhitungan manual, perhitungan sistem, dan dari seorang pakar yang nantinya akan menghasilkan keakuratan sistem. Penelitian ini menghasilkan keakuratan diagnosa penyakit sebesar 86,66 % dan error sebesar 13,34 % dari 15x pengujian. (Amrullah & Ekojono, 2013)

Saefudin dan YG. Rosi Tri Rianti (2015), Sistem Pakar Untuk Mendiagnosa Gangguan Pencernaan Pada Anak Dengan Metode *Forward Chaining*. Sistem pakar

merupakan aplikasi yang pengetahuannya diadopsi dari manusia kemudian dituangkan ke komputer. Dengan menggunakan metode *forward chaining* yang diawali dengan pemilihan gejala awal dan dilanjutkan sampai menemukan hasil informasi yang telah sesuai dengan *rule*. Tujuan yang diperoleh dari penelitian ini adalah untuk membangun aplikasi sistem pakar dalam mendiagnosa gangguan pencernaan pada anak dengan menggunakan metode *forward chaining*. Perancangan aplikasi menggunakan beberapa *software* diantaranya adalah *UML*, bahasa *Borland Dhelpi 7*, serta *database* yang menggunakan *MySQL* dan sistem pakar tersebut berhasil diimplementasikan di Klinik Prima Husada (Saefudin & Rianti, 2015)

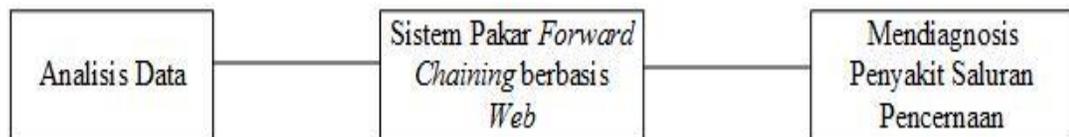
Oktozia, Cheng-Hong Yang dan Li-Yeh Chuang (2016), *An Application of Expert System for Diagnosing Fever Caused by Viral Infection*. Demam juga dikenal sebagai suhu tinggi, bukan disebabkan oleh penyakit. Itu biasanya merupakan gejala kondisi yang mendasarinya, paling sering adalah infeksi. Pada anak-anak dengan demam, gejala yang menyertainya seperti lesu, rewel, nafsu makan yang buruk, sakit tenggorokan, sakit telinga, muntah dan diare. Beberapa orang tua berpikir mereka tidak boleh pergi ke rumah sakit ketika anak-anak demam, karena mereka akan diejek. Sebagai solusi pertolongan pertama, Aplikasi sistem pakar (*AEX*) dapat membantu orang tua untuk mengidentifikasi demam yang disebabkan oleh infeksi virus. Aplikasi ini mengimplementasikan *forward chaining* sebagai teknik pencarian melalui sistem *rule-base*. Beberapa kode pemrograman juga ditulis dalam *PHP* untuk membuat pengurangan fakta baru dari aturan di basis pengetahuan. Sistem pakar berbasis *web*

memungkinkan pengguna untuk mendiagnosis penyakit anak-anak kapan saja dan di mana saja, hanya dengan mengakses internet. (Oktoria, Yang, & Chuang, 2016)

Ahmad Mrouf, Islam Albatish, Msbah Mosa, Samy S. Abu Naser (2017), *Knowladge Based System for Long-term Abdomial Pain (Stomach Pain) Diagosis and Treatmet*). Perut adalah bagian tubuh antara *thorax* (dada) dan panggul, pada manusia. Perut berisi sebagian besar tabung seperti organ saluran pencernaan, serta beberapa organ padat. Organ-organ perut berongga terdiri dari perut, usus kecil, dan usus besar dengan usus buntu yang menempel. Organ-organ seperti hati, kandung empedu yang melekat, dan pankreas berhubungan erat dengan saluran pencernaan berkomunikasi dengannya melalui saluran. Tujuan utama dari sistem pakar ini adalah untuk mendapatkan diagnosis penyakit pencernaan yang tepat dan pengobatan yang tepat. Sistem pakar yang diusulkan yang dibuat untuk membantu dokter dalam mendiagnosis banyak penyakit pencernaan seperti: hiatal hernia, gastritis, maag atau nyeri ulu hati; sistem pakar yang diusulkan menyajikan gambaran tentang penyakit perut yang diberikan, penyebab penyakit diuraikan dan pengobatan penyakit bila memungkinkan diberikan. Sistem pakar yang diusulkan sangat berguna untuk dokter internis, pasien dengan masalah perut dan dokter yang baru lulus. (Mrouf, Albatish, Mosa, & Naser, 2017)

2.5 Kerangka Pemikiran

Alur yang dipahami dan digunakan untuk memecahkan masalah dalam penelitian secara logis dan sistematis dimuat dalam kerangka pemikiran. Variabel penelitian yang ditetapkan untuk menjadi permasalahan harus dijelaskan secara teoritis pada kerangka berfikir (Sugiyono, 2014). Kerangka pemikiran dalam mendasari penelitian ini adalah berikut ini:



Gambar 2. 16 Kerangka Pemikiran

Seluruh data yang dibutuhkan tentang penyakit saluran pencernaan dianalisa terlebih dahulu agar dalam melakukan pengolahan data menjadi lebih mudah. Dari data yang telah diperoleh dan dianalisis kemudian diolah dengan metode inferensi *forward chaining* yang akan dituangkan menjadi suatu sistem berbasis *web*. Sehingga menghasilkan suatu sistem pakar untuk mendiagnosa penyakit saluran pencernaan pada manusia.