

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. Berikut adalah beberapa definisi kecerdasan buatan yang telah didefinisikan oleh beberapa ahli. Menurut pendapat Turing, ahli matematika berkebangsaan Inggris yang dijuluki bapak komputer modern dan pembongkar sandi Nazi dalam era Perang Dunia 1950, menetapkan definisi *Artificial intelligence* yaitu jika komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, dan mempunyai kecerdasan (Sutojo, dkk., 2011:1).

Menurut pendapat ahli McCarthy dari Stanford mendefinisikan kecerdasan sebagai kemampuan untuk mencapai sukses dalam menyelesaikan suatu permasalahan. Berbeda dengan Simon mengartikan kecerdasan buatan merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman

komputer untuk melakukan suatu hal yang dalam pandangan manusia adalah cerdas (Sutojo, dkk., 2011: 2).

Para ahli mendefinisikan *Artificial intelligence* secara berbeda-beda tergantung pada sudut pandang mereka masing-masing. Ada yang fokus pada logika berfikir manusia saja, tetapi ada juga yang mendefinisikan *Artificial intelligence* secara lebih luas pada tingkah laku manusia. Russel dan Norvig mengelompokkan definisi *Artificial intelligence*, yang diperoleh dari beberapa *textbook* berbeda, kedalam empat kategori, yaitu:

1. ***Thinking humanly: the cognitive modeling approach***

Pendekatan ini dilakukan dengan dua cara sebagai berikut:

- a. Melalui introspeksi: mencoba menangkap pemikiran-pemikiran kita sendiri pada saat kita berfikir. Tetapi, seorang psikolog Barat mengatakan: “*how do you know that you understand?*” Bagaimana anda tahu bahwa anda mengerti? Karena pada saat anda menyadari pemikiran anda, ternyata pemikiran tersebut sudah lewat dan digantikan kesadaran anda. Sehingga, definisi ini terkesan mengada-ada dan tidak mungkin dilakukan.
- b. Melalui eksperimen-eksperimen psikolog (Suyanto, 2014: 3).

2. ***Acting humanly: the Turing test approach***

Pada tahun 1950, Alan Turing merancang suatu ujian bagi komputer berinteligensi untuk menguji apakah komputer tersebut mampu mengelabui seorang manusia yang menginterogasinya melalui *teletype* (komunikasi berbasis teks jarak jauh). Jika *interrogator* tidak dapat

membedakan yang diinterogasi adalah manusia atau komputer, maka komputer berinteligensi tersebut lolos dari *Turing test*. Komputer tersebut perlu memiliki kemampuan: *Natural Language Processing, Knowledge Representation, Automated Reasoning, Machine Learning, Computer Vision, Robotics*. *Turing test* sengaja menghindari interaksi fisik antara *interrogator* dan komputer karena simulasi fisik manusia tidak memerlukan inteligensi.

3. ***Thinking rationally: the laws of thought approach***

Terdapat dua masalah dalam pendekatan ini, yaitu:

- a. Tidak mudah untuk membuat pengetahuan informal dan menyatakan pernyataan tersebut kedalam *formal term* yang diperlukan oleh notasi logika, khususnya ketika pengetahuan tersebut memiliki kepastian kurang dari 100%.
- b. Terdapat perbedaan besar antara dapat memecahkan masalah “dalam prinsip” dan memecahkannya “dalam dunia nyata”.

4. ***Acting rationally: the rational agent approach***

Membuat inferensi yang logis merupakan bagian dari suatu *rational agent*.

Hal ini disebabkan satu-satunya cara untuk melakukan aksi secara rasional adalah dengan menalar secara logis. Dengan menalar secara logis, maka bisa didapatkan kesimpulan bahwa aksi yang diberikan akan mencapai tujuan atau tidak. Jika mencapai tujuan, maka *agent* dapat melakukan aksi berdasarkan kesimpulan tersebut.

Thinking humanly dan *acting humanly* adalah dua definisi dalam arti yang sangat luas. Sampai saat ini, pemikiran manusia yang diluar rasio, yakni refleks dan intuitif (berhubungan dengan perasaan), belum dapat ditirukan sepenuhnya oleh komputer. Dengan demikian, kedua definisi ini dirasa kurang tepat untuk saat ini. Jika kita menggunakan definisi ini, maka banyak produk komputasi cerdas saat ini yang tidak layak disebut sebagai produk *Artificial intelligence*. Definisi *thinking rationally* terasa lebih sempit daripada *acting rationally*. Oleh karena itu, definisi *Artificial intelligence* yang paling tepat untuk saat ini adalah *acting rationally* dengan pendekatan *rational agent*. Hal ini berdasarkan pemikiran bahwa komputer bisa melakukan penalaran secara logis dan juga bisa melakukan aksi secara rasional bedasarkan hasil penalaran tersebut (Suyanto, 2014: 4).

Menurut Winston dan Prendergast (1984), tujuan dari kecerdasan buatan adalah:

1. Membuat mesin menjadi lebih pintar (tujuan utama)
2. Memahami apa itu kecerdasan (tujuan ilmiah)
3. Membuat mesin lebih bermanfaat (tujuan *entrepreneurial*)

Dari definisi ini, maka kecerdasan buatan menawarkan media maupun uji teori tentang kecerdasan. Teori-teori ini nantinya dapat dinyatakan dalam bahasa pemograman dan eksekusinya dapat dibuktikan pada komputer nyata. Dari sini, dapat dikatakan bahwa cerdas adalah memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Jadi, agar mesin bisa cerdas (bertindak seperti manusia), maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar (Sutojo, dkk., 2011: 3).

Selama bertahun-tahun para filsuf berusaha mempelajari kecerdasan yang dimiliki manusia. Dari pemikiran tersebut lahirlah *Artificial intelligence* sebagai cabang ilmu yang mempelajari dan meniru kecerdasan manusia. Sejak saat itu para peneliti mulai memikirkan perkembangan *Artificial intelligence* sehingga teori-teori dan prinsip-prinsipnya berkembang terus hingga sekarang. Mulai pada abad ke-17 sampai abad ke-19 merupakan titik awal perkembangan kecerdasan buatan. Hal ini ditandai oleh penemuan-penemuan seperti Rene Descartes yang mengemukakan bahwa semua tidak ada yang pasti, kecuali kenyataan bahwa seseorang bisa berpikir, Blaise Pascal berhasil menciptakan mesin penghitung digital mekanisme pertama pada 1642, Charles Babbage dan Ada Lovelace berhasil membuat mesin penghitung mekanisme yang dapat di program, Bertrand Russell dan Alfred North Whitehead yang berhasil menerbitkan buku *Principia Mathematica*, yang merombak logika formal, Walter Pitts menerbitkan Kalkulus logis pada 1943 yang merupakan fondasi untuk jaringan saraf tiruan. Pada 1950-1970 merupakan tahun pembuka bagi kecerdasan buatan, di mana para ilmuwan dan peneliti mulai memikirkan cara agar mesin dapat melakukan pekerjaannya seperti yang dilakukan oleh manusia. Hal ini dibuktikan oleh beberapa penemuan seperti pada tahun 1951, *University of Manchester* telah berhasil mengembangkan komputer elektronik pertama di dunia yang diberi nama "Ferranti Mark I", pada tahun 1951 sebuah program permainan catur berhasil dibuat oleh Dietrich Prinz, Alan Turing seorang matematikawan Inggris yang pertama kali mengusulkan adanya tes untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. John McCarthy membuat istilah kecerdasan buatan pada konferensi pertama yang

disediakan untuk pokok persoalan ini, pada 1956. Dia juga menemukan bahasa pemrograman Lisp, ELIZA diprogram oleh Joseph Weizenbaum (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan, Alain Colmerauer mengembangkan bahasa komputer Prolog, selama 1960-an dan 1970-an Joel Moses mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah didalam program Maccy, program berbasis pengetahuan yang sukses pertama kali dalam bidang matematika, Marvin Minsky dan Seymour Papert menerbitkan Perceptrons, yang mendemonstrasikan batas jaringan saraf tiruan sederhana, Ted Shortliffe mendemonstrasikan kekuatan sistem berbasis aturan untuk representasi pengetahuan dan inferensi dalam diagnosis dan terapi medis yang kadang kala disebut sebagai sistem pakar pertama. Pada 1980-an, jaringan saraf tiruan digunakan secara meluas dengan algoritma perambatan balik. Perolehan besar dalam berbagai bidang *Artificial intelligence* dan demonstrasi berbagai macam aplikasi berlangsung pada 1990-an. Lebih khusus *Deep Blue*, sebuah komputer permainan catur, mengalahkan Garry Kasparov dalam sebuah pertandingan 6 *game* yang terkenal pada tahun 1997. DARPA menyatakan bahwa biaya yang disimpan melalui penerapan metode *Artificial intelligence* untuk unit penjadwalan dalam Perang Teluk pertama telah mengganti seluruh investasi dalam penelitian *Artificial intelligence* sejak tahun 1950 pada pemerintah Amerika Serikat. Tantangan hebat DARPA, yang dimulai pada 2004 dan berlanjut hingga hari ini, adalah sebuah pacuan untuk hadiah \$2 juta di mana kendaraan dikemudikan sendiri tanpa komunikasi dengan manusia, menggunakan GPS, komputer dan

susunan sensor yang canggih, melintasi beberapa ratus mil daerah gurun yang menantang (Sutojo, dkk., 2011: 3-8).

Dibawah ini akan dijelaskan beberapa macam bidang yang menggunakan kecerdasan buatan.

2.1.1.1 Logika Fuzzy (*Fuzzy Logic*)

Konsep tentang logika *fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akuisi data, dan sistem control. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak ataupun kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 atau 1. Dalam logika *fuzzy* memungkinkan nilai keanggotaan berada diantara 0 dan 1. Artinya bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”. “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan di berbagai bidang, seperti pada sistem diagnosa penyakit (dalam bidang kedokteran), pemodelan sistem pemasaran, riset operasi (dalam bidang ekonomi), kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola dalam bidang teknik (Sutojo, dkk., 2011: 211).

Bila dibandingkan dengan logika konvensional, kelebihan logika *fuzzy* adalah kemampuannya dalam proses penalaran secara bahasa sehingga dalam perancangannya tidak memerlukan persamaan matematik yang rumit. Beberapa alasan yang dapat diutarakan mengapa kita menggunakan logika *fuzzy* diantaranya adalah mudah dimengerti, memiliki toleransi terhadap data-data yang tidak tepat, mampu memodelkan fungsi-fungsi *nonlinear* yang sangat kompleks, dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan, dapat bekerja sama dengan teknik-teknik kendali secara konvensional, dan didasarkan pada bahasa alami (Sutojo, dkk., 2011: 211).

Memahami logika *fuzzy*, sebelum memahami logika *fuzzy* terlebih dahulu perhatikan konsep himpunan *fuzzy*. Himpunan *fuzzy* memiliki 2 atribut, yaitu (Sutojo, dkk., 2011: 212):

1. *Linguistik*, yaitu nama suatu kelompok mewakili suatu keadaan tertentu menggunakan bahasa alami, misalnya DINGIN, SEJUK, PANAS, mewakili variabel temperatur. Contoh lain misalnya MUDA, PAROBAYA, TUA, mewakili variabel umur.
2. *Numeris*, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 10, 35, 40, dan sebagainya.

Disamping itu, ada beberapa hal yang harus dipahami dalam memahami logika *fuzzy*, yaitu (Sutojo, dkk., 2011: 211-212):

1. Variabel *fuzzy*, yaitu variabel yang akan dibahas dalam suatu sistem *fuzzy*.
Contoh: penghasilan, temperatur, permintaan, umur, dan sebagainya.

2. Himpunan *fuzzy*, yaitu suatu kelompok yang mewakili suatu keadaan tertentu dalam suatu variabel *fuzzy*.
3. Semesta pembicara, yaitu seluruh nilai yang diijinkan untuk dioperasikan dalam suatu variabel *fuzzy*.
4. *Domain* himpunan *fuzzy*, yaitu seluruh nilai yang diijinkan dalam semesta pembicara dan boleh dioperasikan dalam suatu himpunan *fuzzy*.

Struktur sistem inferensi *fuzzy*, terbentuk dari empat elemen antara lain

(Sutojo, dkk., 2011: 211-232):

1. Basis pengetahuan *fuzzy*, yaitu kumpulan aturan *fuzzy* dalam bentuk pernyataan *IF...THEN*.
2. Fuzzifikasi, yaitu proses untuk mengubah input sistem yang mempunyai nilai tegas menjadi variabel *linguistik* menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.
3. Mesin inferensi, yaitu proses untuk mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan yang telah ditetapkan pada basis pengetahuan *fuzzy*.
4. Defuzzifikasi, yaitu mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzifikasi.

Terdapat beberapa metode yang digunakan dalam sistem inferensi *fuzzy*, yaitu (Sutojo, dkk., 2011: 233-237):

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan berikut:

- a. Fuzifikasi
- b. Pembentukan basis data pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan nilai *MIN* (minimum)
- d. Defuzzyfikasi menggunakan nilai *Average* (rata-rata)

2. Metode Mamdani

Metode Mamdani paling sering digunakan dalam aplikasi-aplikasi karena strukturnya sederhana, yaitu menggunakan operasi *MIN-MAX* atau *MAX-PRODUCT*. Untuk mendapatkan *output*, diperlukan 4 tahapan berikut:

- a. Fuzzyfikasi
- b. Pembentukan basis pengetahuan *fuzzy* (dalam bentuk *IF...THEN*)
- c. Aplikasi fungsi implikasi menggunakan fungsi *MIN* dan komposisi antara-*rule* menggunakan fungsi *MAX* (menghasilkan himpunan *fuzzy* baru)
- d. Defuzzyfikasi menggunakan metode *Centroid*

3. Metode Sugeno

Dalam metode Sugeno, *output* sistem berupa konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada 1985. Dalam inferensinya metode Sugeno menggunakan tahapan:

- a. Fuzzyfikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan fungsi implikasi *MIN*
- d. Defuzzyfikasi menggunakan *Average* (rata-rata)

2.1.1.2 Jaringan Saraf Tiruan (JST)

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja jaringan saraf tiruan seperti cara kerja manusia, yaitu belajar melalui contoh. Sebuah jaringan saraf tiruan dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, melalui proses pembelajaran. Belajar dalam sistem biologis melibatkan penyesuaian terhadap koneksi *synaptic* yang ada antar *neuron*, hal ini berlaku juga untuk jaringan syaraf tiruan (Sutojo, dkk., 2011: 283).

Menurut definisi dari Alexander dan Morton jaringan syaraf tiruan adalah prosesor tersebar paralel (*parallel distributed processor*) yang sangat besar yang memiliki kecenderungan untuk menyimpan pengetahuan yang bersifat pengalaman dan membuatnya siap untuk digunakan. Jaringan syaraf tiruan menyerupai otak manusia dalam dua hal, yaitu: pengetahuan diperoleh jaringan melalui proses belajar; kekuatan hubungan antar sel saraf (*neuron*) yang dikenal sebagai bobot-bobot *sinaptik* digunakan untuk menyimpan pengetahuan.

Jaringan syaraf tiruan mempunyai struktur tersebar paralel yang sangat besar dan mempunyai kemampuan belajar, sehingga bisa melakukan *generalization* atau diterjemahkan sebagai generalisasi, yaitu bisa menghasilkan

output yang benar untuk *input* yang belum pernah dilatihkan. Dengan kedua kemampuan pemrosesan informasi ini, jaringan saraf tiruan mampu menyelesaikan masalah-masalah yang sangat kompleks. Pada prakteknya, jaringan saraf tiruan tidak bisa memberikan solusi dengan bekerja sendiri, tetapi perlu diintegrasikan kedalam pendekatan rekayasa sistem yang konsisten. Setiap masalah yang kompleks didekomposisi kedalam sejumlah tugas sederhana, dan jaringan saraf tiruan ditugaskan sebagai bagian dari tugas-tugas tersebut (seperti pengenalan pola, *assosiative memory*, kontrol, dan sebagainya) yang sesuai dengan kemampuannya (Suyanto, 2014: 170).

Jaringan saraf tiruan memiliki kelebihan-kelebihan antara lain (Sutojo, dkk, 2011: 284):

1. Belajar *Adaptive*, ialah kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-Organisation*, ialah kemampuan untuk membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real Time Operation*, ialah perhitungan jaringan saraf tiruan yang dapat dilakukan secara parallel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Kelemahan-kelemahan yang dimiliki oleh jaringan saraf tiruan, yaitu (Sutojo, dkk, 2011: 284-285):

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi *numerik* dengan persis tinggi.

2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatik, operasi logika, dan simbolis.
3. Untuk beroperasi jaringan saraf tiruan butuh pelatihan sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

Suatu model jaringan saraf tiruan dapat disebut baik atau ditentukan oleh hubungan antara *neuron* atau yang biasa disebut arsitektur jaringan. *Neuron-neuron* tersebut terkumpul dalam lapisan-lapisan yang disebut *layers*. Terdapat 3 lapisan penyusun jaringan saraf tiruan, antara lain (Sutojo, dkk., 2011: 292):

1. Lapisan *input* (*input layer*)

Unit-unit dalam lapisan *input* disebut unit-unit yang bertugas menerima pola masukan dari luar yang menggambarkan suatu permasalahan.

2. Lapisan tersembunyi (*hidden layer*)

Unit-unit dalam lapisan tersembunyi disebut unit-unit tersembunyi, yang mana nilai *output*-nya tidak dapat diamati secara langsung.

3. Lapisan *output* (*output layer*)

Unit-unit dalam lapisan *output* disebut unit-unit *output*, yang merupakan solusi jaringan saraf tiruan terhadap suatu permasalahan.

2.1.1.3 Sistem Pakar

Sistem pakar merupakan cabang dari *artificial intelligence* (AI), sistem pakar berasal dari istilah *knowledge based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan kedalam komputer. Seorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk untuk *knowledge assistant* (Sutojo, dkk., 2011: 160).

2.1.2 Sistem Pakar

Sistem pakar merupakan cabang dari *artificial intelligence* (AI) yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver* (GPS) yang dikembangkan oleh Newel dan Simon. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti MYCIN untuk diagnosa penyakit, DENDRAL untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON & XSEL untuk membantu konfigurasi sistem komputer besar, SHOPIE untuk analisis sirkuit elektronik, Prospector digunakan di bidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberikan keputusan bagi seorang *manager* dalam stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya (Sutojo, dkk., 2011: 159).

Menurut Sutojo dkk, sistem pakar adalah suatu sistem yang dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari dalam menjawab pertanyaan dan memecahkan suatu masalah. Dalam bukunya Kecerdasan Buatan (2011: 160), ada beberapa defenisi sistem pakar menurut para ahli:

1. Turban (2001, p402)

“Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia”.

2. Jackson (1999, p3)

“Sistem pakar adalah program komputer yang mempresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memberikan saran”.

3. Luger dan stubblefield (1993, p308)

“Sistem pakar adalah program yang berbasiskan pengetahuan yang menyediakan solusi, kualitas pakar kepada masalah-masalah dalam bidang (domain) yang spesifik.

2.1.2.1 Manfaat Sistem Pakar

Menurut Sutojo, dkk. (2011: 160) Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya:

1. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.

2. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, sistem pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespon dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawabannya.
10. Bisa digunakan sebagai media pelengkap dan pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

2.1.2.2 Kekurangan Sistem Pakar

Sutojo, dkk. (2011: 161) menerangkan bahwa selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya:

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.

2.1.2.3 Ciri-Ciri Sistem Pakar

Ciri ciri dari sistem pakar (Sutojo dkk, 2011: 162) adalah sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/*rule* tertentu.
5. Mudah di modifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

2.1.2.4 Area Permasalahan Aplikasi Sistem Pakar

Menurut Sutojo, dkk. (2011: 162) biasanya aplikasi sistem pakar menyentuh beberapa area permasalahan berikut:

1. *Interpretasi*: menghasilkan deskripsi situasi berdasarkan data-data masukan.
2. *Prediksi*: memperkirakan akibat yang mungkin terjadi dari situasi yang ada.
3. *Diagnosis*: menyimpulkan suatu keadaan berdasarkan gejala-gejala yang diberikan (*symptoms*).
4. *Desain*: melakukan perancangan berdasarkan kendala-kendala yang diberikan.
5. *Panning*: merencanakan tindakan-tindakan yang akan dilakukan.
6. *Monitoring*: membandingkan hasil pengamatan dengan proses perencanaan.
7. *Debugging*: menentukan penyelesaian dari suatu kesalahan sistem.
8. *Reparasi*: melaksanakan rencana perbaikan.
9. *Instruction*: melakukan instruksi untuk diagnosis, *debugging*, dan perbaikan kinerja.
10. *Kontrol*: melakukan kontrol terhadap hasil interpretasi, diagnosis, *debugging*, monitoring, dan perbaikan tingkah laku sistem.

2.1.2.5 Konsep Dasar Sistem Pakar

1. Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran dapat memungkinkan para ahli untuk mengambil keputusan dengan cepat dan lebih baik daripada seorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang (Sutojo, dkk., 2011: 163):

1. Fakta-fakta tentang bidang permasalahan tertentu.
2. Teori-teori tentang bidang permasalahan tertentu.
3. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya.
4. Aturan *heuristic* yang harus dikerjakan dalam suatu situasi tertentu.
5. Strategi global untuk memecahkan permasalahan tertentu.
6. *Meta knowledge* atau pengetahuan tentang pengetahuan (Sutojo, dkk., 2011: 163).

2. Pakar (*Expert*)

Pakar adalah seorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu

menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya (Sutojo, dkk., 2011: 163).

3. Pemindehan Kepakaran (*Transferring Expertise*)

Menurut Sutojo, dkk. (2011: 164) tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam computer, kemudian di transfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan yaitu:

1. Akuisi pengetahuan (dari pakar atau sumber lain).
2. Representasi pengetahuan (pada komputer)
3. Inferensi pengetahuan.
4. Pemindehan pengetahuan ke pengguna.

4. Inferensi

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin *inferensi* yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin *inferensi* adalah untuk mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya (Sutojo, dkk., 2011: 164).

5. Aturan-aturan (*rule*)

Kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based systems*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecah masalah (Sutojo, dkk., 2011: 165).

6. Kemampuan Menjelaskan (*Explanation Capability*)

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya. Karakteristik dan kemampuan yang dimiliki oleh sistem pakar berbeda dengan sistem konvensional. Perbedaan ini ditunjukkan pada tabel di bawah ini (Sutojo, dkk., 2011: 165):

Tabel 2.1 Perbandingan antara Sistem Konvensional dengan Sistem Pakar

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesannya biasanya digabungkan dalam satu program.	Basis pengetahuan dipisahkan secara jelas dengan mekanisme inferensi.
Program tidak membuat kesalahan (yang membuat kesalahan: pemrograman atau pengguna).	Program dapat berbuat kesalahan

Tabel 2.1 Lanjutan

Biasanya tidak menjelaskan mengapa data masukan diperlukan atau bagaimana <i>output</i> dihasilkan.	Penjelasan merupakan bagian terpenting dari sistem pakar.
Perubahan program sangat menyulitkan.	Perubahan dalam aturan-aturan mudah di lakukan.
Sistem hanya bisa beroperasi setelah lengkap atau sesuai.	Sistem dapat beroperasi dengan aturan-aturan yang sedikit (sebagai <i>prototipe</i> awal).
Eksekusi dilakukan langkah demi langkah (<i>algoritmik</i>).	Eksekusi dilakukan dengan menggunakan <i>heuristic</i> dan logika pada seluruh basis pengetahuan.
Perlu informasi lengkap agar bias beroperasi.	Dapat beroperasi dengan informasi tidak lengkap atau mengandung ketidakpastian.
Manipulasi efektif dari basis data yang besar.	Manipulasi efektif dari basis pengetahuan yang besar.
Menggunakan data	Menggunakan pengetahuan.
Tujuan utama: efisiensi.	Tujuan utama: efektivitas.
Mudah berurusan dengan data kuantitatif	Mudah berurusan dengan data kualitatif.
Menangkap, menambah dan mendistribusikan akses ke data <i>numerik</i> atau informasi.	Menangkap, menambah dan mendistribusikan akses ke pertimbangan dan pengetahuan.

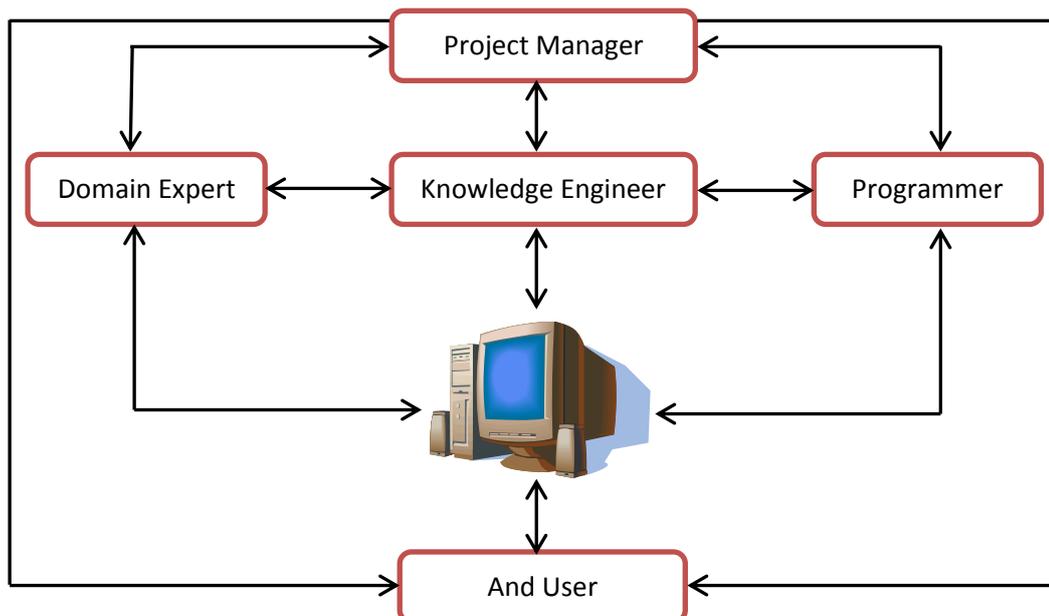
Sumber: Sutojo, dkk., (2011: 165)

7. Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan kedalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar (Sutojo, dkk., 2011: 166).

8. Tim Pengembangan Sistem Pakar

Berikut adalah gambaran tim pengembangan sistem pakar yang akan dijabarkan berupa gambar dan penjelasannya.



Gambar 2.1 Tim Pengembangan Sistem Pakar

Domain expert adalah pengetahuan dan kemampuan seorang pakar untuk menyelesaikan masalah terbatas pada keahliannya saja. Misalnya seorang pakar penyakit jantung, ia hanya mampu menangani masalah-masalah yang berkaitan dengan masalah penyakit jantung saja. Ia tidak bisa menyelesaikan masalah-masalah ekonomi, politik, hukum dan lain-lain. Keahlian inilah yang dimasukkan kedalam sistem pakar. *Knowledge Engineer* (perekayasa pengetahuan) adalah orang yang mampu mendesain, membangun, dan menguji sebuah sistem pakar. *Programmer* adalah orang yang membuat program sistem pakar, mengode domain pengetahuan agar dapat dimengerti oleh komputer. *Project manager* adalah pemimpin dalam tim pengembangan sistem pakar. *End-User* (biasanya disebut user saja) adalah orang yang menggunakan sistem pakar (Sutojo, dkk., 2011: 170).

2.1.2.6 Teknik Inferensi

Pada sistem pakar berbasis *rule*, domain pengetahuan di presentasikan dalam sebuah kumpulan rule berbentuk *IF-THEN*, sedangkan data di presentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat ini. Mesin inferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam *database*. Jika bagian *IF* (kondisi) dari *rule database* sebagai fakta baru yang ditambahkan (Sutojo, dkk., 2011: 171).

2.1.2.6.1 *Forward Chaining*

Forward chaining adalah suatu strategi pengambilan keputusan yang dimulai dari premis (fakta) menuju konklusi (kesimpulan akhir). Salah satu aspek penting dari perolehan fakta adalah dengan menanyakan pertanyaan yang benar. Pertanyaan benar yang diajukan menghasilkan efisiensi dalam menentukan jawaban yang benar. Salah satu syarat yang nyata untuk hal ini adalah sistem pakar hanya akan menanyakan pertanyaan yang berhubungan dengan hipotesis yang dicoba dibuktikan (Yunus dan Setyowibowo, 2010: 102).

Forward chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam *database*. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. Metode pencarian yang digunakan adalah *depth-first search* (DFS), *breadth-first search* (BFS) atau *best first search* (Sutojo, dkk., 2011: 171).

2.1.2.6.2 *Backward Chaining*

Backward chaining adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada di bagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-

fakta yang ada cocok dengan premis-premis di bagian *IF*. Jika cocok, *rule* dieksekusi, kemudian hipotesis di bagian *THEN* di tempatkan di bagian basis data sebagai fakta baru. Jika tidak cocok, simpan premis dibagian *IF* ke dalam *stack* sebagai *subGoal*. Proses berakhir jika *Goal* ditemukan atau tidak ada *rule* yang bisa membuktikan kebenaran dari *subGoal* atau *Goal* (Sutojo, dkk., 2011:178).

Sedangkan menurut Suraya (2012: B-341) *Backward chaining* adalah pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (*THEN* dulu). Dengan kata lain, penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. Kedua metode *inferensi* tersebut dipengaruhi oleh tiga macam penelusuran, yaitu *Depth-first search*, *Breadth-first search* dan *Best-first search*.

1. *Breadth-first search*, Pencarian dimulai dari simpul akar terus ke *level* 1 dari kiri ke kanan dalam 1 *level* sebelum berpindah ke *level* berikutnya.
2. *Depth-first search*, Pencarian dimulai dari simpul akar ke *level* yang lebih tinggi. Proses ini dilakukan terus hingga solusinya ditemukan atau jika menemui jalan buntu.
3. *Best-first search*, bekerja berdasarkan kombinasi kedua metode sebelumnya.

2.1.3 World Wide Web

World wide web (www), lebih dikenal dengan *web*, merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke *internet*. *Web*

pada awalnya adalah ruang informasi dalam *internet*, dengan menggunakan teknologi *hyperteks*, pemakai dituntun untuk menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen *web* yang ditampilkan dalam *browser web*. Kini *internet* identik dengan *web*, karena kepopuleran *web* sebagai standar *interface* pada layanan-layanan yang ada di *internet*, dari awalnya sebagai penyedia informasi, kini digunakan juga untuk komunikasi dari *email* sampai dengan *chatting*, sampai dengan melakukan transaksi bisnis (*commerce*). *Web* memudahkan pengguna komputer untuk berinteraksi dengan pelaku *internet* lainnya dan menelusuri (informasi) di *internet*. Selain itu, *web* telah diadopsi oleh perusahaan sebagai sebagian dari strategi teknologi informasinya, karena beberapa alasan (Sidik dan Pohan, 2012: 1):

1. Akses informasi mudah,
2. Setup server lebih mudah,
3. Informasi mudah distribusikan, dan
4. Bebas platform; informasi dapat disajikan oleh browser web pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan (Sidik dan Pohan, 2012: 1).

2.1.3.1 Langkah Kerja Web

Langkah kerja *web* yang pertama yaitu, informasi *web* disimpan dalam dokumen yang disebut dengan halaman-halaman *web* (*web pages*), *web page* adalah *file-file* yang disimpan dalam komputer yang disebut dengan *server-server web* (*web servers*). Selanjutnya komputer membaca *web page* disebut sebagai *web*

client, *web client* menampilkan *page* dengan menggunakan program yang disebut dengan *browser web* (*web browser*), *browser web* yang populer adalah *Internet Explorer* dan *Netscape Navigator* (Sidik dan Pohan, 2012: 4).

2.1.3.2 Browser Web

Browser web adalah *software* yang digunakan untuk menampilkan informasi dari *server web*. *Software* ini kini telah dikembangkan dengan menggunakan *user interface grafis*, sehingga pemakai dapat dengan melakukan ‘*point dan click*’ untuk pindah antar dokumen. Dapat dikatakan saat ini hanya dua *browser web* GUI yang populer: *Internet Explorer* dan *Netscape Navigator*. Kedua *browser* ini bersaing untuk merebut pemakainya, dengan berusaha untuk mendekati standar spesifikasi dokumen HTML yang direkomendasikan oleh W₃C. Suatu *browser* mengambil sebuah *web page* dari *server* dengan sebuah *request*. Sebuah *request* adalah sebuah *request* HTTP standar yang berisi sebuah *page address*. Sebuah *page address* terlihat seperti berikut: *http// www.kita.web.id/page htm*. Seluruh *web page* berisi instruksi-instruksi bagaimana untuk ditampilkan. *Browser* menampilkan *page* dengan membaca instruksi-instruksi ini. Instruksi yang paling umum untuk menampilkan disebut dengan *tag* HTML (Sidik dan Pohan, 2012: 5).

2.1.3.3 *Server Web*

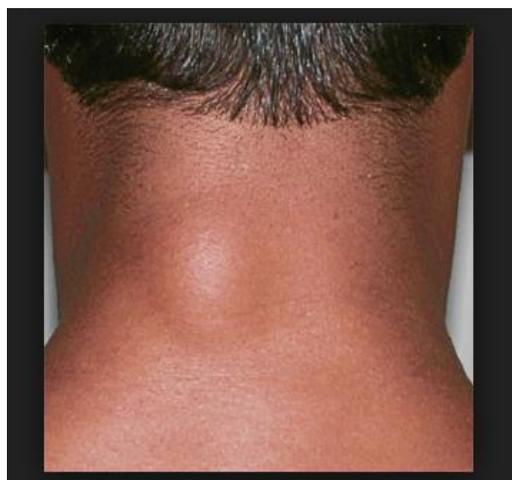
Server web adalah komputer yang digunakan untuk menyimpan dokumen-dokumen *web*, komputer ini akan melayani permintaan dokumen *web* dari kliennya. *Browser web* seperti Explorer atau Navigator berkomunikasi melalui jaringan (termasuk jaringan internet) dengan *server web*, menggunakan HTTP. *Browser* akan mengirimkan *request* kepada *server* untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh *server*. *Server* memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protokol HTTP (Sidik dan Pohan, 2012: 5).

2.1.3.4 **Istilah-Istilah yang Harus Diketahui**

Menurut Sidik dan Pohan (2012: 6) penjelasan tentang istilah WWW, *Web*, HTML, URL, *Surfing* dan *Server*. WWW, awalnya merupakan suatu layanan penyajian informasi di *internet* dengan menggunakan HTML. *Web* identik dengan *internet*, karena kepopulerannya saat ini, *web* sudah menjadi *interface* aplikasi untuk melakukan transaksi dan sajian informasi yang lengkap dari seluruh dunia. Protokol secara definisi adalah standar pemrosesan informasi yang diterima secara umum. Protokol TCP/IP merupakan protokol yang digunakan dalam komunikasi dalam *internet*. Sedangkan URL (*Universal Resource Locator*) adalah konsep nama *file* standar yang diperluas dengan jaringan.

2.1.4 Lipoma

Lipoma adalah benjolan lemak yang tumbuh secara lambat di antara kulit dan lapisan otot. Lipoma bisa bergerak atau bergeser jika ditekan dengan jari secara perlahan-lahan dan terasa lunak. Ketika ditekan, lipoma biasanya tidak menyebabkan rasa sakit. Lipoma tidak memerlukan perawatan karena biasanya tidak berbahaya dan tidak bersifat kanker, namun operasi pengangkatan lipoma bisa dilakukan jika lipoma yang diderita tumbuh besar dan mulai menimbulkan rasa sakit. Sebagian pasien memiliki lebih dari satu lipoma dan umumnya orang-orang paruh baya yang lebih sering terkena lipoma. www.alodokter.com/lipoma. (Diakses 08 November 2016).



Gambar 2.2 Lipoma pada Pundak

Dalam penelitian yang dilakukan oleh Ratih dan Anna (2012: 44) lipoma adalah tumor jinak yang tumbuh di bawah kulit dan merupakan endapan lemak. Lipoma adalah tumor jinak dan jarang berubah menjadi ganas. Beberapa orang hanya memiliki satu lipoma, sedangkan lainnya memiliki beberapa lipoma. Lipoma lebih sering tumbuh di lengan, batang tubuh dan leher bagian belakang

juga lebih sering ditemukan pada perempuan dibandingkan laki-laki. Hal ini sesuai dengan temuan bahwa mayoritas kasus lipoma berdasarkan jenis kelamin laki-laki berusia lebih banyak pada perempuan dari pada laki-laki. Berdasarkan data-data penelitian prevalensi lipoma sekitar 1 % pada populasi umum dan bisa terjadi pada semua umur, sedangkan lipoma cutan jarang terjadi pada anak-anak. Kejadian Lipoma lebih banyak ditemukan pada perempuan karena pengaruh *hormone estrogen*. Tindakan yang diperlukan adalah pengangkatan tumor. Hasil penelitian memperlihatkan bahwa kasus lipoma menyerang penduduk dari umur 10 sampai dengan 70 tahun. Dan mayoritas menyerang kelompok umur 40-49 tahun pada pria dan pada perempuan pada kelompok umur 30-39 tahun dan 50-59 tahun. Hal ini sesuai dengan teori yang mengatakan bahwa lipoma biasanya ditemukan pada orang dewasa dari umur 40 sampai dengan 60 tahun.

Pengertian lipoma tidak jauh berbeda dengan sumber lain, lipoma adalah suatu tumor (benjolan) jinak yang berada di bawah kulit yang terdiri dari lemak. Biasanya lipoma dijumpai pada usia lanjut (40-60 tahun), namun juga dapat dijumpai pada anak-anak. Lipoma kebanyakan berukuran kecil, namun dapat tumbuh hingga mencapai lebih dari diameter 6 cm (Zainuddin, dkk., 2014: 259).

Menurut Siregar dan Bachsinar (1993: 102) dijelaskan bahwa seorang pria yang berumur 58 tahun terkena lipoma pada tengkuk dengan benjolan kira-kira sebesar telur bebek pada tengkuk sejak 4 tahun terakhir. Dengan konsistensi lembek, mobilitas +, tidak nyeri. Penderita hanya mengeluhkan nyeri kepala terutama pada daerah sekitar tengkuk. Sehingga didiagnosis sebagai penderita penyakit lipoma pada tengkuk, dan tindakan pengobatannya adalah dengan

ekstirpasi tumor, yaitu tindakan pengangkatan seluruh massa tumor beserta kapsulnya.

Seorang pria yang berumur 46 tahun terkena lipoma pada lengan atas dengan keluhan benjolan pada lengan atas sejak 7 tahun lalu. Benjolan berdiameter 2,5 x 1,5 cm, dengan konsistensi lunak. Mobilitas ⁺, bila dipijit nyeri. Keluhan lain, kadang-kadang benjolan terasa nyeri. *Mutiple dimpling* ⁺, didiagnosis orang tersebut sebagai penderita penyakit lipoma pada lengan atas. Tindakan yang dianjurkan dengan *ekstirpasi* tumor, yaitu tindakan pengangkatan seluruh massa tumor beserta kapsulnya (Siregar dan Bachsinar, 1993: 107).

2.1.4.1 Gejala Lipoma

Lipoma bisa muncul di bagian tubuh manapun. Berikut ini adalah beberapa gejala atau ciri-ciri lipoma. www.alodokter.com/lipoma. (Diakses 08 November 2016):

1. Ukuran lipoma biasanya memiliki diameter kurang dari 5 cm, namun ukurannya bisa bertambah besar karena lipoma bisa tumbuh.
2. Jika ditekan menggunakan jari, lipoma akan mudah bergerak, serta terasa lembek.
3. Lipoma terletak di bawah kulit dan biasa muncul di area punggung, paha, leher, lengan, perut, atau bahu.
4. Jika lipoma tumbuh makin besar dan mengandung banyak pembuluh darah atau menekan saraf di sekitarnya, lipoma akan terasa sakit www.alodokter.com/lipoma. (Diakses 08 November 2016).

Lipoma bisa tumbuh lebih besar dan lebih dalam, namun hal ini jarang terjadi. Jika tumbuh benjolan di area tubuh manapun, segera temui dokter untuk pemeriksaan dan penanganan lebih lanjut. Selain lipoma ada juga penyakit lain yang menyebabkan tumbuhnya benjolan berisi cairan, biasanya nanah, di bawah kulit yaitu disebut kista. Berikut ini adalah beberapa perbedaan antara lipoma dengan kista. www.alodokter.com/lipoma. (Diakses 08 November 2016):

1. Jika disentuh, lipoma akan terasa lunak, namun kista akan terasa keras.
2. Lipoma terletak lebih dalam di bawah kulit, sedangkan kista dekat dengan permukaan kulit.
3. Lipoma tidak menyebabkan peradangan pada kulit, tapi kista bisa menyebabkan kulit membengkak dan berwarna kemerahan.

2.1.4.2 Penyebab Lipoma

Penyebab pasti lipoma belum diketahui, tapi ada beberapa faktor risiko yang bisa meningkatkan resiko seseorang terkena lipoma. www.alodokter.com/lipoma. (Diakses 08 November 2016):

1. Faktor genetika kemungkinan berperan dalam munculnya lipoma.
2. Faktor keturunan.
3. Luka pada bagian kulit.
4. Walau lipoma bisa menimpa orang pada segala usia, namun lebih sering terjadi pada orang-orang yang berusia antara 40-60 tahun, dan jarang menimpa anak-anak.

5. Orang-orang yang menderita penyakit tertentu, seperti penyakit madelung, *sindrom Cowden*, *sindrom Gardner*, dan *adiposis dolorosa*.

2.1.4.3 Jenis-jenis Lipoma

Berikut beberapa jenis lipoma www.halosehat.com/penyakit/lipoma/bahaya-lipoma (Diakses 08 November 2016):

1. Jenis lipomatosis simetris, adalah lipoma yang sering tumbuh pada bagian leher, bahu, kulit kepala dan *proximal extremitas* atas. Benjolan lunak berisi lemak tersebut dapat tumbuh dengan subur pada orang-orang yang memiliki penyakit diabetes mellitus kronis dan para pecandu alkohol yang telah mengkonsumsi minuman keras sejak usia remaja.
2. Jenis lipoma soliter, adalah lipoma yang berukuran kecil dan akan seiring waktu akan bertambah besar ketika seseorang mengalami kegemukan atau menderita obesitas. Benjolan lunak akan berangsur-angsur hilang dengan sendirinya ketika seseorang melakukan diet untuk mengembalikan berat idealnya kembali.
3. Jenis *dercum* lipoma-*Dercum* lipoma menimbulkan rasa nyeri, dapat muncul dan tumbuh dengan subur pada wanita yang telah mengalami menopause, para pecandu alkohol dan orang-orang yang menderita depresi berat.
4. Jenis Lipoma *diffuse* kongenital-Lipoma *diffuse* kongenital adalah lipoma yang sering tumbuh pada bagian punggung dan pinggang bagian belakang yang disebabkan oleh genetika. Lipoma tersebut dapat tumbuh

dalam jumlah yang cukup banyak namun mengalami pembesaran yang sangat lambat. Tumor ini terdiri dari jaringan lemak yang immatur. www.halosehat.com/penyakit/lipoma/bahaya-lipoma (Diakses pada 08 November 2016).

2.1.4.4 Diagnosis Lipoma

Ada beberapa cara untuk mendiagnosis lipoma, seperti pemeriksaan fisik dan pengambilan sampel jaringan untuk pemeriksaan laboratorium yang disebut dengan *biopsi*. Untuk membedakan benjolan lipoma dari benjolan kista, bisa dilakukan pemeriksaan dengan *ultrasound*. Jika lipoma yang timbul memiliki bentuk yang tidak biasa, berukuran besar, dan terlihat lebih dalam dari jaringan lemak, diagnosis dapat dilakukan dengan tes pencitraan, seperti CT scan atau MRI. Dan jika lipoma yang muncul tumbuh dengan cepat, terasa sakit dan tidak bergerak di bawah kulit, ada kemungkinan itu adalah *liposarcoma* atau tumor ganas yang tumbuh di jaringan lemak. www.alodokter.com/lipoma. (Diakses 08 November 2016).

2.1.4.5 Perawatan Lipoma

Ada beberapa perawatan yang dapat dilakukan jika lipoma tumbuh menimbulkan rasa yang tidak nyaman, sakit atau mengganggu, dan terus berkembang. Anda bisa melakukan sedot lemak menggunakan jarum atau selang besar untuk menyingkirkan gumpalan lemak. Alternatif lain adalah dengan

melakukan suntik *steroid* untuk menyusutkan lipoma, namun biasanya tidak bisa untuk menghilangkan lipoma seluruhnya. Cara terakhir paling banyak dilakukan adalah melalui operasi pengangkatan lipoma. Biasanya lipoma tidak akan tumbuh kembali setelah diangkat, namun bisa menimbulkan efek samping seperti memar dan bekas luka. www.alodokter.com/lipoma. (Diakses 08 November 2016).

2.2 Variabel

Variabel penelitian adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk di pelajari sehingga diperoleh informasi dan kesimpulannya (Sudaryono, 2015: 16).

Menurut Sugiyono (2014: 39) variabel bebas adalah merupakan variabel yang mempengaruhi atau yang menjadi sebab perubahannya atau timbulnya variabel dependen (terikat). Variabel bebas pada penelitian ini yaitu penyakit lipoma pada manusia. Variabel terikat atau dependen merupakan variabel yang dipengaruhi atau yang menjadi akibat, karena adanya variabel bebas. Variabel terikat pada penelitian ini yaitu, timbulnya benjolan di permukaan kulit dengan ukuran 2 sampai 5 senti meter dan akan terus membesar hingga berdiameter 10 senti meter, benjolan tidak menimbulkan rasa sakit, lipoma terasa seperti gumpalan yang berada dibawah kulit dan dapat tumbuh pada hampir semua bagian tubuh, benjolan tidak menimbulkan rasa nyeri ketika disentuh atau ditekan oleh jari, terasa lunak dan dapat bergeser jika disentuh tangan dengan perlahan, benjolan lipoma tidak berwarna, lipoma dapat tumbuh lebih dari satu buah,

benjolan lipoma akan memiliki ukuran yang sama walaupun sudah bertahun-tahun karena pertumbuhan lipoma sangat lambat.

2.3 Software Pendukung

Software pendukung merupakan beberapa perangkat lunak yang digunakan untuk mendukung pembuatan sistem pakar dalam penelitian ini. Perangkat lunak tersebut antara lain: HTML, PHP, *phpMyAdmin*, CSS, *Dreamweaver CS5.5*, *Notepad++*, XAMPP, dan MySQL.

2.3.1 Hyper Text Markup Language (HTML)

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*. Menurut sumber yang penulis kutip dari Wikipedia, *hyper text markup language* digunakan untuk menampilkan berbagai informasi didalam sebuah penjelajah *web Internet* dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII (*American Standard Code for Information Interchange*) agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata, disimpan ke dalam format ASCII normal sehingga menjadi *homepage* dengan perintah-perintah HTML. *Hyper text markup language* menggunakan dua macam *ekstensi file* yaitu **.htm** dan **.html**. Format

ekstensi berformat **.htm** awalnya hanyalah untuk mengakomodasi penggunaan html dalam operasi DOS (Saputra, 2012: 1).



Gambar 2.3 Logo HTML

1. Struktur Dasar HTML (*Hyper Text Markup Language*)

Saputra (2012: 5) dokumen *hyper text markup language* memiliki sebuah struktur yang harus kita ikuti aturan pembuatannya. Beberapa elemen wajib yang ada pada *file hyper text markup language* apabila kita ingin membangun suatu pondasi kerangka *website*. Elemen tersebut diantaranya:

1. Elemen *hyper text markup language* (html)

Elemen html merupakan *tag* dasar apabila kita ingin memulai suatu dokumen html. Secara logika, jika kita menemukan *tag* ini, berarti secara jelas dapat didefinisikan bahwa dokumen ini merupakan suatu dokumen html. *Tag* ini merupakan perintah wajib bagi pemrogram *web* untuk menuliskan *tag* pertama dalam dokumen html. Contoh *tag*-nya adalah: **<html>** dan diakhiri dengan **</html>**

2. Elemen *Head*

Head merupakan *tag* berikutnya setelah elemen html (**<html>**), yang berfungsi untuk menuliskan keterangan tentang dokumen *web* yang akan

ditampilkan. Elemen ini nantinya akan diakhiri dengan tanda penutup

`</head>`. Jadi jika secara runtun dapat dituliskan format berikut:

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

3. Elemen *Title*

Elemen *title* merupakan suatu elemen yang harus dituliskan didalam *elemen head* yang digunakan untuk memberikan judul/informasi pada *caption browser web* tentang topik/tema atau judul dari suatu dokumen *web* yang ditampilkan pada *browser*. Berikut struktur penggunaannya:

```
<html>
<head>
  <title Tuliskan Judul disini </title>
</head>
</head>
```

4. Elemen *Body*

Elemen *body* merupakan bagian utama dalam dokumen *web*. Jika kita ingin menampilkan suatu teks atau informasi atau yang dikenal dengan sebutan konten, maka kita harus meletakkan teks tersebut pada elemen *body*. Struktur elemennya sebagai berikut:

```
<html>
<head>
  <title> Tuliskan Judul disini </title>
</body>
</html>
```

2.3.2 PHP (*Hypertext Preprocessor*)

PHP (*hypertext preprocessor*) adalah bahasa scrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP (*hypertext preprocessor*) banyak dipakai untuk pemrograman situs *web* dinamis. PHP (*hypertext preprocessor*) dapat digunakan untuk membangun sebuah CMS (*content management system*). Untuk menggunakan *hypertext preprocessor* ada beberapa *software* yang harus kita install, diantaranya Apache server, Php, php My admin, dan Mysql server (Aditya, 2011: 1).



Gambar 2.4 Logo PHP

Saputra, (2012: 91) untuk menggunakan bahasa pemrograman *hypertext preprocessor*, kita harus memulainya dengan tanda khusus seperti:

```
<?php
// Sintaks PHP
?>
```

Ada empat macam format yang bisa digunakan untuk memulai pemrograman *hypertext preprocessor* didalam kode Anda, yaitu:

```
<?php ..... ?>
<? ..... ?>
<script language="PHP"> ..... </script>
<% ..... %>
```

Dari ke-empat macam format tersebut, sintaks `<?php ... ?>` dan `<? ... ?>` merupakan format yang paling banyak digunakan oleh programmer. Ada dua macam perintah untuk menampilkan suatu kalimat atau *string*, yaitu

menggunakan perintah *echo* dan *print*, terserah Anda ingin menggunakan yang mana.

1. Variabel dan Tipe Data

Variabel adalah tempat penyimpanan sementara didalam memory komputer. Penulisan variabel didalam pemrograman *hypertext preprocessor*, ada aturan tertentu yang harus kita ketahui, diantaranya (Saputra, 2012: 92):

1. Penulisan variabel harus diawali dengan simbol dollar (\$).
2. Karakter pertama setelah simbol dollar, tidak boleh menggunakan angka (harus huruf).

Contoh penggunaan yang salah: **\$123**

Contoh penggunaan yang benar: **\$hore**

3. Setelah simbol dollar dan huruf, maka karakter selanjutnya boleh menggunakan angka.

Contoh: **\$hore123**

2. Parameter (*POST* dan *GET*)

PHP (*hypertext preprocessor*) memiliki dua macam method yang bisa digunakan untuk mengirim data dari *client (browser)* ke *server*, yaitu *POST* dan *GET*. Metode *POST*, digunakan untuk mengirim data dari *client* ke *server* dimana data tersebut akan dikirimkan melalui *http header*. Metode *GET*, yang bekerja dengan perintah fungsi **\$_GET** akan mengirimkan data atau berkomunikasi melalui parameter yang akan terlihat pada URL (Saputra, 2012: 93).

3. Fungsi Waktu

PHP (*hypertext preprocessor*) memiliki banyak sekali fungsi untuk menampilkan waktu, diantaranya (Saputra, 2012: 97):

1. *Checkdate()*, digunakan untuk memeriksa keabsahan tanggal dan waktu.
Format dasar: *Checkdate* (bulan, hari, tahun).
2. *Date()*, digunakan untuk menampilkan waktu saat ini. Format dasar: *Date* ('fungsi tanggal').
3. *Date_Default_Timezone_Set*, digunakan untuk menentukan/menerapkan zona waktu. Zona waktu berdasarkan daerah dimana Anda tinggal.
Format: *Date_default_timezone_set* ('zona daerah').
4. *Getdate()*, digunakan untuk memperoleh informasi mengenai tanggal dan waktu saat ini dalam bentuk array. Format: *Getdate* (*timestamp*)
5. *Idate()*, digunakan untuk men-set waktu dan tanggal lokal ke dalam bentuk *integer* (bilangan bulat) sehingga tanggal lokal yang mengandung format *string*, tidak akan ditampilkan. Format: *idate* (*format, timestamp*)
6. *Strtotime()*, digunakan untuk merubah tipe *string* menjadi format penanggalan / waktu. Format: *Strtotime* (*time, now*).

Array merupakan variabel tunggal yang dapat menampung banyak nilai. Tiap nilai ditampung dalam *index array* yang berbeda pula. *Array* biasanya digunakan untuk menampung data yang dapat disusun berdasarkan nomor tertentu (Saputra, 2012: 107).

4. Kalimat Pengulangan

Sering kali kita mendapatkan kondisi dimana terdapat suatu proses yang mengharuskan kita menggunakan kalimat pengulangan, contohnya: menampilkan nomor bilangan 1 s/d 10. Kalo kita menuliskan secara manual, maka sangat tidak efisien apalagi jika kita ingin menampilkan bilangan 1 s/d 1000. Ada beberapa fungsi yang dapat kita gunakan, diantaranya *For*, *For Loop*, *While*, *Do... While*, dan *Foreach* (Saputra, 2012: 109).

1. Pengulangan *For*, digunakan untuk melakukan pengulangan yang terhitung. Sintaks dasar:
For (awal; kondisi; penambahan) {Kode untuk di jalankan}
2. Berbeda dengan perintah *for* yang hanya bisa melakukan perulangan yang terhitung, perintah *while* bisa melakukan perulangan baik yang terhitung maupun tidak terhitung. Fungsi *while*, akan kita temukan pada saat kita hendak menampilkan seluruh data dari tabel ke halaman *web*.
3. *Do ... While*, merupakan suatu perulangan yang akan terus dilakukan jika nilai yang dihasilkan terpenuhi dan akan terhenti jika nilai tidak terpenuhi. Kode yang ada didalam *Do While* akan dieksekusi minimal sekali.
4. *Foreach*, merupakan sebuah perulangan pada variabel bertipe *array* (Saputra, 2012: 109).

5. Pernyataan Kontrol

Pada pemrograman PHP, kita dapat melakukan pengkodean dengan menggunakan perintah bersyarat atau perintah *If Else* yaitu perintah yang akan dieksekusi apabila persyaratan telah terpenuhi. PHP memiliki empat macam model / kerangka fungsi pernyataan kontrol, diantaranya *if*, *if - else*, *if - elseif - else*, dan *switch* (Saputra, 2012: 113).

1. Pernyataan *If*, digunakan apabila kita hanya memiliki satu kondisi dan kondisi tersebut akan dijalankan jika pernyataan dianggap benar. Format dasarnya:

```
If (kondisi) {
    // Pernyataan / kondisi
}
```

2. Pernyataan *If-Else*, digunakan apabila kita memiliki dua kondisi perbandingan. Jika kondisi/pernyataan bernilai *TRUE*, maka pernyataan 1 akan dijalankan, namun jika kondisi bernilai sebaliknya, maka pernyataan 2 akan dijalankan. Formatnya:

```
If (kondisi) {
    // Pernyataan / kondisi 1
}
else{
    // Pernyataan / kondisi 2
}
```

3. Pernyataan *If - Elseif - Else*, pernyataan ini digunakan apabila kita memiliki banyak kondisi (minimal tiga kondisi). Formatnya:

```
if (kondisi) {
    // Pernyataan 1
}
elseif (kondisi2) {
    // Pernyataan 2
}
```

```

else{
    // Pernyataan 3
}

```

4. *Switch*, merupakan alternatif pengganti *If*. Perintah *switch* akan menyeleksi kondisi yang diberikan dan kemudian membandingkan hasilnya dengan konstanta-konstanta yang ada didalam *case*. Perbandingan akan dimulai dari konstanta pertama sampai terakhir. Jika kondisi ditemukan maka program akan membaca kode didalam konstanta tersebut, dan ketika bertemu perintah *break*.

```

Switch (ekspresi) {
    Case "Kondisi 1";
    Pernyataan 1;
    Break;
    Case "kondisi 2";
    Pernyataan 2
    Break;
}

```

6. Operasi CRUD

CRUD merupakan singkatan dari *Craet*, *Read*, *Update*, dan *Delete*, yang merupakan suatu fungsi yang digunakan untuk mengolah suatu data secara dinamis. Untuk menggunakan operasi tersebut, kita harus terhubung dengan *database*. Berikut penjelasan dari CRUD (Saputra, 2012: 117):

1. *Creat* (Menambahkan Data)

Creat, istilah lain dari proses "Menambah Data", digunakan apabila kita ingin menambah suatu data kedalam tabel pada database.

2. *Read* (Membaca/Menampilkan Data)

Setelah data berhasil disimpan, maka langkah selanjutnya timbul pertanyaan, bagaimana untuk menampilkan data yang telah disimpan tersebut? Proses inilah yang dinamakan dengan proses “*Read*”. Untuk menampilkan data, kita akan menggunakan fungsi *mysql_fetch_array*.

3. *Update* (Merubah Data)

Data yang telah kita simpan terkadang mengandung kesalahan *input* atau perlu diperbaharui karena itu, kita perlu proses mengupdate data.

4. *Delete* (Menghapus Data)

Ketika data sudah tidak dibutuhkan lagi, Anda bisa menghapusnya dari *database* agar hilang dari muka bumi ini (Saputra, 2012: 117).

2.3.3 *phpMyAdmin*

phpMyAdmin adalah perangkat lunak yang ditulis dalam bahasa pemrograman *PHP*, dimaksudkan untuk menangani administrasi *MySQL* melalui *Web*, berikut logo dari *phpMyAdmin*. www.phpmyadmin.net/ (Diakses pada 10 November 2016).



Gambar 2.5 Logo *phpMyAdmin*

phpMyAdmin adalah perangkat lunak yang ditulis dalam bahasa pemrograman *PHP*, dimaksudkan untuk menangani administrasi *MySQL* melalui *Web*. *phpMyAdmin* mendukung berbagai operasi pada *MySQL* dan *MariaDB*. Operasi

(mengelola database, tabel, hubungan, *indeks*, *users*, *permissions*, dan lain-lain) dapat dilakukan melalui antarmuka pengguna masih memiliki kemampuan langsung mengeksekusi pernyataan *SQL*. www.phpmyadmin.net/ (Diakses pada 10 November 2016).

2.3.4 *Cascading Style Sheet (CSS)*

CSS merupakan singkatan dari *cascading style sheet*, yaitu bahasa pemrograman *web* yang didesain khusus untuk mengendalikan dan membangun komponen dalam *web* sehingga tampilan *web* lebih rapih, terstruktur dan seragam.



Gambar 2.6 Logo CSS

Cascading style sheet merupakan salah satu pemrograman wajib disamping html yang harus dikuasai oleh setiap pemrogram *web*, terlebih lagi itu adalah *web designer*. Tujuan utama dari *cascading style sheet* adalah memisahkan konten utama dengan tampilan dokumen lainnya (html dan sejenisnya). Dengan adanya pemisahan ini, akses konten pada *web* meningkat. *Web* yang menggunakan *cascading style sheet* akan lebih ringan dan mudah untuk dibuka dibandingkan dengan *web* yang tidak menggunakan *cascading style sheet*. Perbedaan ini akan semakin terasa ketika *web* yang anda buka mempunyai data yang banyak. Tujuan lainnya adalah untuk mempercepat pembuatan halaman *web*. Anda hanya perlu

membuat satu properti dan properti tersebut dapat digunakan pada elemen lainnya, artinya anda tidak perlu menulis ulang kode program yang digunakan berulang kali. *Cascading style sheet* saat ini dikembangkan oleh *World Wide Web Consortium* atau yang biasa lebih dikenal dengan istilah W3C. Sehingga *Cascading Style Sheet* menjadi bahasa standard dalam pembuatan *web*. *Cascading Style Sheet* bukan menggantikan kode *hyper text markup language*, tetapi hanya difungsikan sebagai penopang atau pendukung (pelengkap) dari *file hyper text markup language* yang berperan dalam penataan kerangka dan *layout* (Saputra, 2012: 27).

1. Mengetahui Berbagai Macam Versi *Cascading Style Sheet*

Cascading style sheet (CSS) saat ini sudah mencapai versi 3 dimana pada setiap versi pasti ada peningkatan yang dilakukan. Versi yang pertama adalah CSS-1, masih kuno, CSS hanya dikembangkan dan digunakan untuk *formatting* dokumen *hyper text markup language*, sedangkan versi kedua yaitu CSS-2, disini sudah mulai terasa mantapnya. *Cascading style sheet* sudah mulai menggunakan *font*, *table-layout*, dan berbagai media untuk *printer*. Maksudnya, *cascading style sheet* dikembangkan untuk memenuhi kebutuhan terhadap format dokumen supaya bisa ditampilkan di *printer*. Versi selanjutnya adalah CSS-3 merupakan *cascading style sheet* pengembangan dari versi sebelumnya. Peningkatan yang paling mencolok pada versi ketiga ini adalah peningkatan fitur yang mengarah pada efek animasi, seperti *crop*, *slide images*, dan lain-lain. Namun, satu yang

harus diingat, saat ini belum semua *web browser* mendukung *Cascading style sheet* 3 (Saputra, 2012: 27).

2. Mengapa *Cascading Style Sheet* (CSS)

Saputra (2012: 28) banyak yang selalu bertanya, mengapa CSS? Bukankah memakai tabel kode html saja sudah cukup? Memang, tanpa CSS pun *web* dapat dibangun, namun tahukah Anda, salah satu kunci kesuksesan *web* agar banyak pengunjung adalah tampilan yang menarik, interaktif, cepat diakses dan lain-lain. Jika *web* Anda sangat lambat hanya saat *loading*, tentu pengunjung juga akan merasa bosan dan menutup *browser web* mereka sebelum seluruh informasi dalam *web* Anda berhasil ditampilkan. Dengan menggunakan *cascading style sheet*, akan banyak yang dapat kita peroleh, diantaranya:

1. Memisahkan pembuatan dokumen (CSS dan HTML).
2. Mempermudah dan mempersingkat pembuatan dan pemeliharaan dokumen *web*.
3. Akses *web* lebih cepat saat di-*loading* (mempercepat pembacaan HTML).
4. Fleksibel, interaktif, tampilan lebih menarik dan nyaman dipandang.
5. Lebih kecil ukuran *file* sehingga *bandwith* yang digunakan juga otomatis menjadi lebih kecil.
6. Dapat digunakan pada semua *web browser*.

2.3.5 *Dreamweaver CS5.5*

Dreamweaver merupakan *software* aplikasi yang digunakan sebagai *editor HTML* untuk mendesain *web* secara *visual*. Aplikasi ini juga yang bisa dikenal dengan istilah *WYSIWYG* (*What You See Is What You Get*), yang intinya adalah bahwa pengguna tidak harus berurusan dengan *tag-tag HTML* yang cukup rumit untuk membuat sebuah halaman *web*. Dengan kemampuan fasilitas yang optimal dalam jendela desain membuat program ini memberikan kemudahan untuk mendesain *web* meskipun untuk para *web desainer* pemula sekalipun. Sedangkan kemampuan *dreamweaver* untuk berinteraksi dengan beberapa bahasa pemrograman seperti *PHP*, *ASP*, *JavaScript*, dan yang lainnya. *Dreamweaver CS5.5* adalah *software* terkemuka untuk membangun dan mengedit *web* dengan menyediakan kemampuan *visual* dan tingkat kode, yang dapat digunakan untuk membuat *website* berbasis standar dan desain untuk *Desktop*, *Mobile*, *Smartphone*, *Tablet*, dan perangkat lainnya (Madcoms, 2016: 14).

2.3.6 *Notepad++*

Menurut Gilmore (2010: 36) *Notepad++* merupakan *editor teks open source* yang matang dan diakui sebagai pengganti *Notepad*, berikut logo dari *Notepad++*.



Gambar 2.7 Logo *Notepad++*

Notepad++ tersedia untuk *platform Windows* yang dapat digunakan untuk menulis kode dengan beberapa pilihan bahasa (pemrograman). *Notepad++* menawarkan beragam kenyamanan fitur yang diharapkan dari setiap kemampuan IDE (*Integrated Development Environment*), termasuk kemampuan untuk menunjukkan baris tertentu dari suatu dokumen sebagai referensi yang mudah; sintaks, tanda kurung, *indentation highlighting*, fasilitas pencarian yang tangguh, *macro recording* untuk tugas-tugas seperti memasukkan *template* komentar, dan sebagainya. Salah satu kelebihan *Notepad++* adalah dukungan dasar untuk *auto-completion* dari nama fungsi yang ditawarkan sehingga akan mengurangi beberapa proses pengetikan kode (Gilmore, 2010: 36).

2.3.7 XAMPP (*X Apache MySQL PHP Perl*)

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program, berikut logo dari XAMPP.



Gambar 2.8 Logo XAMPP

Fungsinya adalah sebagai server yang berdiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl.

Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis (Aditya, 2011: 16).

2.3.8 MySQL

MySQL merupakan salah satu database kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. MySQL bekerja menggunakan bahasa SQL (*Structure Query Language*) yang merupakan bahasa standard yang digunakan untuk manipulasi *database*. Pada umumnya, perintah yang sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, *structure query language* juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data (Saputra, 2012: 77).



Gambar 2.9 Logo MySQL

1. Mengapa MySQL

Ada beberapa alasan yang menjadikan database MySQL sangat diminati oleh para *programmer*, diantaranya (Saputra, 2012: 78):

1. Bersifat *open source*.
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dalam pengolahan data.
3. *Performance* dan *reliable*, pemrosesan *database*-nya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*ease of use*).
5. Memiliki dukungan (*group*) pengguna MySQL.
6. Lintas *Platform*, dapat digunakan pada berbagai sistem operasi berbeda.
7. *Multiuser*, dimana MySQL dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.
8. Dan masih banyak lagi (Saputra, 2012: 78).

2.4 StarUML

Salah satu pemodelan yang saat ini paling banyak digunakan adalah UML (*Unified Modeling Language*), berikut logo UML.



Gambar 2.10 Logo *StarUML*

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S. dan Shalahuddin, 2013: 133).

Menurut A.S. dan Shalahuddin (2013: 137-138) UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML tidak terbatas pada metodologi pemrograman tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

StarUML merupakan salah satu *CASE (Computer-Aided Software Engineering) tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak (A.S. dan Shalahuddin, 2013: 122-123).

Terdapat 13 macam diagram dalam *UML* yang dibagi menjadi 3 kategori yaitu (A.S. dan Shalahuddin, 2013: 140-141):

1. *Structure diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Diagram UML yang termasuk dalam kategori ini antara lain *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.

2. *Behaviour diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada

sebuah sistem. Diagram UML yang termasuk dalam kategori ini antara lain *use case diagram*, *activity diagram*, dan *state machine diagram*.

3. *Interaction diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Diagram UML yang termasuk dalam kategori ini antara lain *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

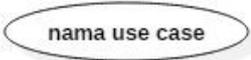
Menurut A.S. dan Shalahudin (2013: 18) *use case* dan *sequence diagram* merupakan bagian dari desain sistem. Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:

1. *Use case diagram*

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal utama yang terdapat pada *use case* yaitu aktor dan *use case*.

Berikut ini adalah simbol-simbol yang digunakan dalam *use case diagram* (A.S. dan Shalahuddin, 2013: 155).

Tabel 2.2 Simbol *Use Case Diagram*

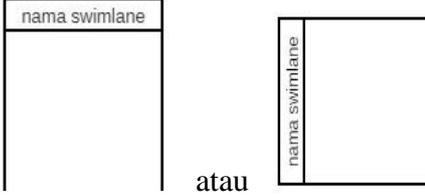
Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi/<i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p>generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan/<i>include/uses</i></p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini. Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>

Sumber: A.S. dan Shalahuddin (2013: 162)

2. Activity diagram

Activity diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Jadi dapat dikatakan bahwa *activity diagram* menggambarkan aktifitas sistem, bukan apa yang dilakukan oleh aktor. Simbol-simbol yang digunakan dalam *activity diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin: 2013: 161-162).

Tabel 2.3 Simbol *Activity Diagram*

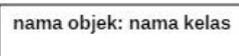
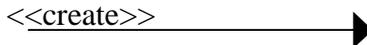
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktifitas memiliki sebuah status awal
Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi

Sumber: A.S. dan Shalahuddin (2013: 162-163)

3. *Sequence diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup (*life cycle*) objek dan *message* (pesan) yang dikirimkan dan diterima antar objek. Jumlah *sequence diagram* yang harus digambar minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. Semakin banyak *use case* yang didefinisikan semakin banyak pula *sequence diagram* yang harus dibuat. Simbol-simbol yang digunakan pada *sequence diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin, 2013: 165-166).

Tabel 2.4 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor/ <i>actor</i>  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Garis hidup/ <i>lifeline</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat

Sumber: A.S. dan Shalahuddin (2013: 166- 167)

2.5 Penelitian Terdahulu

Pada sub judul ini dijabarkan jurnal dari beberapa penelitian yang berhubungan dengan judul yang diangkat pada penelitian ini yang telah dilakukan oleh peneliti lain sebelumnya, yang dapat mendukung sebagai dasar pembahasan penelitian serta digunakan untuk memperkuat dan menambah referensi penelitian.

Paryati (2011), dalam penelitiannya tentang **“Sistem Pakar Berbasis Web Untuk Mendiagnosa Penyakit Kulit”** diperoleh kesimpulan:

1. Memberikan informasi kepada *user* mengenai penyakit kulit yang dideritanya (diagnosa awal) berdasarkan gejala-gejala yang diberikan.
2. Memahami dan memperoleh model pengetahuan untuk gejala-gejala penyakit kulit.
3. Membantu dalam melakukan identifikasi penyakit kulit secara dini, melalui pengolahan computer, sehingga penanganan lebih lanjut terhadap penyakit tersebut dapat dengan cepat dilakukan.
4. Memberikan informasi pengobatan yang bisa dilakukan dan cara mengatasinya.
5. Data yang terdapat pada sistem dapat di *update* atau ditambah sesuai dengan kebutuhan.

Ratih Oemiati dan Anna Maria Sirait (2011), dalam penelitian mereka tentang **“Kejadian Tumor di Desa Grujugan Kabupaten Banyumas Jawa Tengah”** diperoleh kesimpulan:

1. Kasus tumor sebanyak 46 kasus tersebar di 8 RW dan mayoritas penderita tinggal di RW 6.
2. Dari 46 kasus, 6 diantaranya meninggal dunia tanpa dilakukan otopsi sehingga penyebab kematian tidak diketahui.
3. Sebanyak 14 kasus telah didiagnosis di bagian patologi anatomi (PA) RSUD dr Margono, 12 kasus sedang dalam proses pemeriksaan PA, sementara itu 20 kasus lainnya hanya didiagnosis secara klinis saja.
4. Hasil patologi anatomi (PA) menunjukkan bahwa 12 orang menderita Lipoma (tumor jinak pada jaringan lunak).
5. Sebanyak 17 kasus dinyatakan sembuh oleh penderita/ keluarga.
6. Hasil pemeriksaan logam pada sumur penduduk di tiga rumah menunjukkan hasil yang masih dalam batas-batas yang normal.
7. Berdasarkan kriteria kejadian luar biasa (KLB) kluster kanker, maka kasus kanker di desa Grujungan belum dapat dimasukkan sebagai kejadian luar biasa (KLB).

M. Rosidi Zamroni, Choirul Anggun Cahyanti, dan Ahmad Jalaluddin (2013), dalam penelitian mereka tentang “**Sistem Pakar Perkembangan Anak Usia 0-12 Bulan Berbasis Web dengan Metode *Forward Chaining***” diperoleh kesimpulan:

1. Sistem pakar disusun dengan menggunakan metode *forward chaining*, yaitu sistem dengan penulusuran alur maju. Sistem *forward chaining* yaitu sistem menyediakan gejala-gejala dari penyakit dalam, *user*

memilih gejala-gejala tersebut kemudian akan ditemukan penyakit dan pengobatannya.

2. Sistem pakar yang dibuat dapat membantu masyarakat, yaitu orang awam dalam mendiagnosa awal perkembangan pada anak tersebut sehingga mempermudah dalam penanganannya.
3. Sistem pakar ini dirancang beserta keluarannya berupa solusi tentang perkembangan anak yang terjadi pada usia 0–12 bulan serta memberikan hasil perkembangan dan saran yang harus dilakukan terhadap perkembangan yang sedang dialami.
4. Seorang *admin*, pakar dan pengguna dapat mengaksesnya melalui jaringan internet sehingga mudah untuk mengelola dan mendapatkan sebuah informasi.

Roni Pambudi dan Sumarno (2015), dalam penelitiannya tentang “**Aplikasi Sistem Pakar Diagnosa Penyakit Kanker Menggunakan Metode *Certainty Factor***” diperoleh kesimpulan:

1. Semakin banyak gejala yang mendekati penyakit sehingga memperbesar kemungkinan seseorang terdeteksi terkena salah satu penyakit kanker.
2. Besarnya nilai Densitas total ditentukan oleh banyaknya kecocokan antara ID gejala dan ID penyakit, serta besarnya nilai densitas (bobot) tiap aturan pada kaidah diagnosa.
3. Nilai densitas berada pada kisaran 0 sampai dengan 1, jika densitas (bobot) mendekati satu, maka kepastiannya mendekati benar.

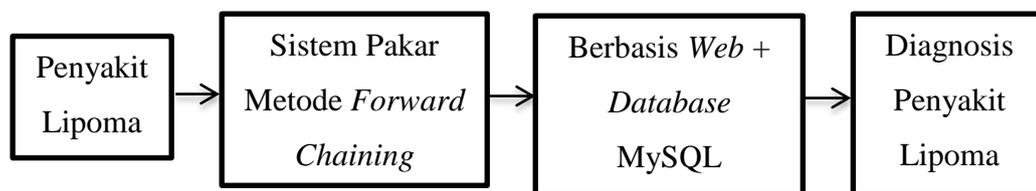
Filipe Jaeger, *et al.*, (2015), dalam penelitian mereka tentang “*Oral Spindle Cell Lipoma in a Rare Location: A Differential Diagnosis*” diperoleh kesimpulan:

1. *In summary, oral spindle cell lipoma is a rare benign lipomatous tumor. Although it presents a predilection for the tongue and cheek, occasional SCL can occur in the palate. The histologic picture shows a range of variations and the observation of morphological features is important in distinguishing SCL from other fusiform tumors. Immunohistochemistry should be helpful in this differentiation.*

2.6 Kerangka Pemikiran

Secara teoritis, kerangka berfikir yang baik akan menjelaskan peraturan antar variabel yang akan di teliti. Kerangka berfikir dalam penelitian perlu dikemukakan apabila penelitian berkaitan dengan dua variabel atau lebih. Kerangka pemikiran adalah penjelasan sementara terhadap gejala-gejala yang menjadi objek permasalahan. Jadi, kerangka berfikir adalah sintesis tentang hubungan antarvariabel yang disusun dari berbagai teori yang telah dideskripsikan. Sintesis itulah yang selanjutnya digunakan untuk merumuskan hipotesis (Sudaryono, 2015: 21).

Dari penjelasan diatas, dengan ini dapat dibuat kerangka pemikiran sebagai berikut:



Gambar 2.11 Kerangka Pemikiran

Langkah pertama akan mengidentifikasi apa yang menjadi masalah, dengan melihat apa yang menjadi penyebab sehingga mengakibatkan manusia terkena penyakit lipoma. Setelah mengidentifikasi masalah tersebut, maka akan melakukan dan menyelesaikannya dengan sistem pakar menggunakan metode *forward chaining*, dari sistem pakar maka akan menemukan apa yang menjadi penyebabnya dan menghasilkan apa saja yang menjadi penyebab penyakit lipoma, kemudian akan menyelesaikannya dengan mesin inferensi dengan metode *forward chaining*, jika sesuai dengan aturan *rule* maka akan dilanjutkan dengan mengaplikasikannya dalam sebuah bahasa pemrograman berbasis *web* dengan menggunakan *database* MySQL, jika tidak sesuai maka akan kembali mengidentifikasi masalah yang terjadi. Agar lebih mudah di pahami, maka penulis akan memasukkan data dari gejala penyakit ke dalam bahasa pemrograman dengan menginputkan gejala-gejala yang dialami oleh pasien (melakukan diagnosa penyakit lipoma). Setelah melakukan pengimputan maka akan menghasilkan (*output*) status penyakit, keterangannya, jenis penyakit lipoma apa yang di derita oleh pasien tersebut, dan solusi yang bisa menyembuhkan penyakit lipoma tersebut.