

BAB II

KAJIAN PUSTAKA

2.1. Teori Dasar

Setelah masalah penelitian dirumuskan, maka langkah selanjutnya dalam proses penelitian (kuantitatif) adalah mencari teori, konsep, serta generalisasi hasil dari penelitian.

Dalam penelitian ini akan di jelaskan secara singkat tentang kecerdasan buatan (*Artificial intelligence*) atau yang biasa disingkat dengan AI serta beberapa subdisiplin ilmunya seperti; sistem pakar (*expert system*), logika *fuzzy* (*fuzzy logic*), dan jaringan syaraf tiruan (*artificial neural network*).

2.1.1. Kecerdasan Buatan (*artificial intelligence*)

Menurut (Budiharto & Sutarsono 2014) Kecerdasaan buatan atau *artificial intelligence* merupakan bidang ilmu komputer yang mempunyai peran penting di era saat ini dan masa yang akan datang. Lebih dari ribuan tahun, cara manusia berpikir terus diteliti. Prosesnya mencakup cara manusia mengetahui, memprediksi, memahami, dan melakuakn manipulasi terhadap hal-hal yang lebih rumit. Bidang keilmuan kecerdasan buatan saat ini terus berproses agar dapat melakukan pekerjaan tersebut. Bukan hanya untuk memecahkan sebuah masalah tetapi juga untuk membangun sebuah sistem atau alat yang memiliki kecerdasan.

Kecerdasan Buatan berasal dari bahasa inggris "*Artificial Intelligence*", yaitu *intellegence* adalah kata sifat yang berarti cerdas, sedangkan *artificial*

artinya buatan, sehingga kecerdasan buatan yang dimaksud yaitu merujuk tentang mesin yang mampu berfikir, menimbang tindakan yang akan dilakukan, dan mampu mengambil keputusan seperti yang dilakukan oleh seorang pakar. (Sutojo, Mulyanto, & Suhartono, 2011, p. 1)

2.1.1.1. Sejarah Kecerdasan Buatan (*artificial intelligence*)

Menurut (Budiharto & Suhartono, 2014, p. 3) McCulloch dan Pitts pada tahun 1943 mengusulkan model matematis bernama *perceptron* dari neuron di dalam otak. Mereka juga menunjukkan bagaimana neuron menjadi aktif seperti sakelar *on-off*. Neuron tersebut mampu belajar dan memberikan aksi berbeda terhadap waktu dari *input* yang diberikan. Sumbangan terbesar dibidang AI diawali oleh tulisan dari Alan Turing tahun 1950 yang mencoba menjawab pertanyaan, “Dapatkah komputer berfikir?” dengan menciptakan *Turing Machine*. Paper Alan Turing pada tahun 1950 berjudul *Computing Machinery and Intelligence* mendiskusikan syarat sebuah mesin dianggap cerdas. Turing beranggapan bahwa jika mesin dapat dengan sukses berperilaku seperti manusia, maka mesin itu dapat dianggap cerdas. Pada akhir 1955, Newell dan Simon mengembangkan *The Logic Theorist*, program AI pertama. Program ini mempresentasikan masalah sebagai model pohon, lalu penyelesaiannya dengan memilih cabang yang akan menghasilkan kesimpulan terbenar. Program ini berdampak besar dan menjadi batu loncatan penting dalam mengembangkan bidang AI.

Pada tahun 1956, John McCarthy dari *Massachusetts Institute of Technology* (MIT) yang dianggap sebagai bapak AI, mengadakan konferensi bertajuk *The Dartmouth Summer Research Project on Artificial Intelligence*. Konferensi ini bertujuan untuk menarik bakat serta minat orang banyak untuk masuk dalam bidang kecerdasan buatan. Konferensi ini juga mempertemukan para pendiri dalam bidang AI yang merupakan dasar bagi masa depan pengembangan serta penelitian AI. Definisi AI pada awalnya diusulkan oleh John McCarthy pada tahun 1955, yaitu: *The goal of AI is to develop machines that behave as though they were intelligence*.

McCarthy mendefinisikan AI sebagai cabang dari ilmu komputer yang berfokus pada pengembangan komputer agar dapat memiliki kemampuan, berpikir dan berperilaku seperti manusia. Untuk menguji definisi tersebut, anda dapat membayangkan sekelompok robot yang berjalan dan bergerak dengan berbagai macam manuver, namun tidak menabrak satu sama lain. Pada tahun 1960 hingga 1970, muncul berbagai macam diskusi bagaimana komputer dapat meniru sedetail mungkin kemampuan otak manusia. Masa tersebut dikategorikan sebagai *Classical AI*. Pada tahun 1980, komputer semakin mudah diperoleh dengan harga yang lebih murah. Hal ini mengakibatkan, berbagai riset di bidang kecerdasan buatan berkembang sangat pesat (Budiharto & Suhartono, 2014, p. 5)

2.1.1.2. Logika Fuzzy

Logika *fuzzy* adalah metodologi sistem kontrol pemecah masalah, yang cocok untuk mengimplementasikan pada sistem, mulai dari sistem yang

seederhana, sistem kecil, *embedded system*, jaringan PC, multi-channel atau workstation berbasisi akuisisi data, dan sistem kontrol. Kemudian metodologi ini diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya(Sutojo et al., 2011, p. 211) Sistem inferensi *fuzzy* merupakan sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia yang melakukan penalaran dengan nalurinya.

2.1.1.3. Jaringan Saraf Tiruan (JST) / *Neural Network*

Menurut (Sutojo et al., 2011, pp. 283–284) Jaringan saraf tiruan mempunyai kemampuan yang cukup luar biasa untuk mendapatkan informasi dari data yang rumit, mampu menyelesaikan permasalahan yang tidak teratur dan sulit di definisikan, mampu mengakuisisi pengetahuan walaupun tidak terdapat kepastian, dapat belajar dari pengalaman, mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, dapat menciptakan suatu pola dari pengetahuan melalui pengaturan diri atau kemampuan belajar (*self organizing*), memiliki kemampuan mengolah data input tanpa harus mempunyai target (*self organizing*), serta mampu menemukan jawaban terbaik sehingga dapat meminimalisasi fungsi biaya (optimasi).

Beberapa kelebihan yang dimiliki oleh jaringan saraf tiruan antara lain :

- a. Belajar *adaftive*: kemampuan yang mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan sebagai pelatihan atau pengalaman awal.

- b. *Self-Organization*: sebuah jaringan saraf tiruan mampu membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
- c. *Real Time Operation*: perhitungan jaringan saraf tiruan dapat dilakukan secara paralel sehingga perangkat keras yang dirancang maupun di produksi secara khusus dapat mengambil keuntungan dari kemampuan tersebut.

2.1.1.4. Sistem Pakar

Menurut(Sutojo et al., 2011, p. 13), Sistem pakar merupakan suatu sistem yang dirancang agar dapat menirukan keahlian dari seorang pakar dalam menjawab pertanyaan dan menyelesaikan suatu masalah. Menurut (Hartati & Iswanti, 2008, p. 2)Salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia adalah sistem pakar. Pemecahan masalah-masalah yang kompleks biasanya hanya dapat dilakukan oleh sejumlah orang yang ahli dalam bidang tersebut. Dengan penerapan teknik kecerdasan buatan, sistem pakar menirukan apa yang dikerjakan oleh seorang pakar ketika mengatasi permasalahan yang rumit, berdasarkan pengetahuan yang dimilikinya.

Pengetahuan sistem pakar dibentuk dari kaidah atau pengalaman tentang perilaku elemen dari domain bidang pengetahuan tertentu. Pengetahuan pada sistem pakar diperoleh dari orang yang mempunyai pengetahuan pada suatu bidang (pakar bidang tertentu), buku-buku, jurnal ilmiah, majalah, maupun dokumentasi yang tercetak lainnya. Sumber pengetahuan tersebut biasa dikenal

dengan sumber keahlian. Pengetahuan-pengetahuan tersebut direpresentasi dalam format tertentu, dan dihimpun dalam suatu basis pengetahuan. Basis pengetahuan ini selanjutnya dipakai sistem pakar untuk menentukan penalaran atas *problema* yang dihadapinya.

Menurut (Sutojo et al., 2011, p. 162) ciri-ciri dari sistem pakar ialah sebagai berikut;

1. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
2. Mampu menjelaskan alasan dengan cara yang dapat dipahami.
3. Terbatas pada domain keahlian tertentu
4. Basis pengetahuan (*Knowledge Base*) dan mekanisme inferensi terpisah.
5. Mudah di modifikasi

Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

2.1.1.4.1. Kelebihan dan Karakteristik Sistem Pakar

Menurut (Budiharto & Suhartono, 2014, p. 134) saat ini sistem pakar banyak digunakan aplikasi-aplikasi terkini dan kompleks, karena;

1. Sistem pakar mampu bertindak sebagai instruktur, konsultan maupun sebagai pasangan atau rekan.
2. Meningkatkan *availability* atau kepaakaan terhadap semua perangkat-prangkat komputer
3. Mengurangi tingkat bahaya

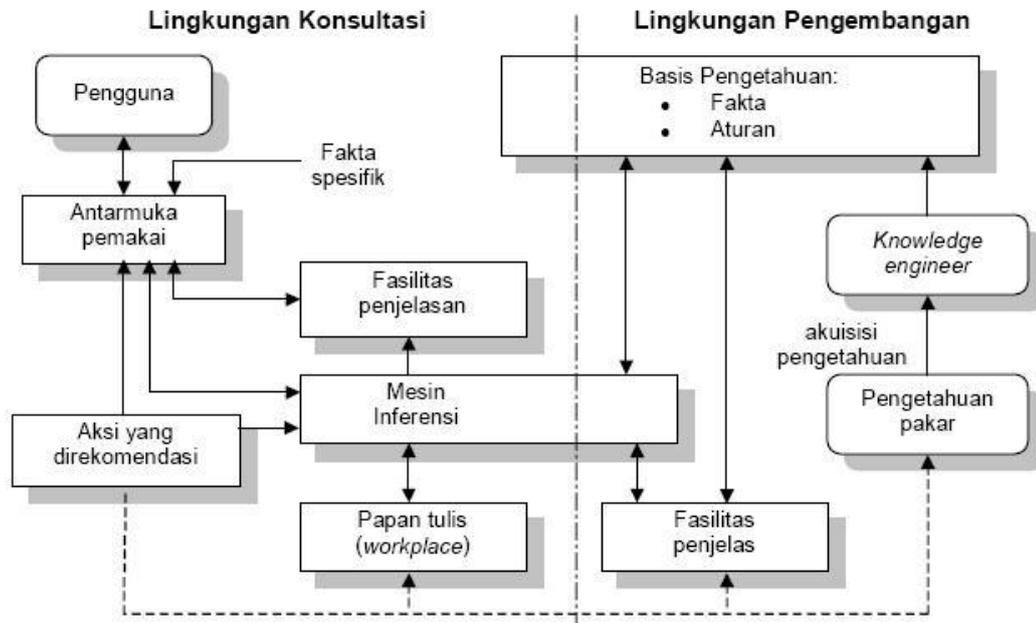
4. Tetap atau permanen
5. *Database* yang cerdas, contoh nya data mining.
6. Pengetahuan bisa saja tidak lengkap, tetapi keahliannya dapat diperluas sesuai dengan kebutuhan. Program konvensional harus “lengkap” sebelum dapat dipergunakan

Sistem pakar biasanya memiliki karakteristik sebagai berikut:

1. *High Performance*
2. *Adequate response time*
3. *Good reliability*
4. *Understandable*

2.1.1.4.2. Struktur Sistem Pakar

Menurut(Sutojo et al., 2011, p. 166) dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembang 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.



Gambar 2.1 Struktur Sistem Pakar
Sumber: (Sutojo et al., 2011, p. 166)

Berikut ini merupakan penjelasan dari subsistem yang ada pada struktur sistem pakar pada Gambar 2.1:

1. Akuisisi Pengetahuan

Akuisisi pengetahuan ini digunakan untuk memasukan pengetahuan yang dimiliki seorang pakar dengan cara merekayasa pengetahuan agar dapat di proses oleh komputer dan meletaknya kedalam basis pengetahuan dengan format dengan format tertentu.

2. Basis pengetahuan

Basis pengetahuan terdiri dari pengetahuan yang dibutuhkan untuk memahami, mengolah, serta menyelesaikan masalah. Dua elemen dasar yang terdapat pada basis pengetahuan yaitu fakta dan *rule*.

3. Mesin Inferensi

Mesin inferensi merupakan program yang digunakan untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang telah ada, merekayasa dan mengarahkan kaidah, fakta dan model yang disimpan dalam basis pengetahuan untuk mencapai solusi serta kesimpulan yang tepat.

4. Daerah kerja

Daerah kerja merupakan area pada memori yang berguna sebagai basis data. Tiga tipe keputusan yang dapat direkam pada daerah kerja yaitu agenda, rencana dan solusi.

5. Antar muka pengguna (*User interface*)

Difungsikan sebagai media komunikasi antara pengguna dengan sistem pakar.

6. Subsistem penjelasan

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

7. Sistem perbaikan pengetahuan

Kemampuan memperbaiki dari seorang pakar diperlukan untuk menganalisa pengetahuan. Belajar dari kesalahan masa lalu, selanjutnya memperbaiki pengetahuan tersebut, sehingga dapat digunakan di masa mendatang.

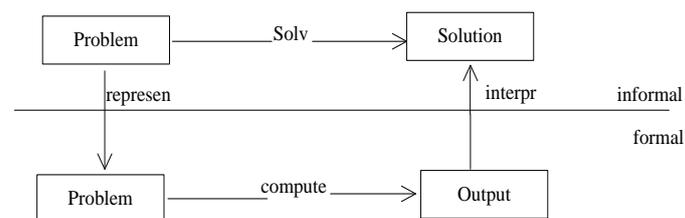
8. Pengguna (*user*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan saran, pelatihan (*training*) serta solusi dari berbagai

permasalahan yang ada. Pengetahuan yang dimasukkan ke dalam sistem pakar ditampilkan dalam bentuk yang mudah dimengerti dan diterima oleh sistem pakar, salah satunya adalah kaidah produksi.

2.1.1.4.3. Representasi Pengetahuan

Menurut (Budiharto & Suhartono, 2014, p. 72) *knowledge* (pengeahuan) adalah informasi mengenai domain yang dapat digunakan untuk menyelesaikan permasalahan dalam domain tersebut. Untuk menyelesaikan banyak masalah, dibutuhkan banyak pengetahuan. Pengetahuan tersebut harus direpresentasikan dalam komputer. Sebagai bagian perancangan program dalam menyelesaikan masalah, definisi pengetahuan harus dipresentasikan dengan jelas. *Representation scheme* adalah bentuk dari pengetahuan yang digunakan dalam *agent*. Representasi dari beberapa potongan pengetahuan adalah representasi internal dari pengetahuan. *Knowledge base* adalah representasi dari keseluruhan pengetahuan yang disamping oleh *agent*.



Gambar 2.2 Peran *Knowledge* Dalam Menyelesaikan Masalah

Sumber: *Artificial Intelligence* 2014

Secara mendasar, masalah dalam konteks umum dapat dibuat solusinya dengan intuisi atau kemampuan logika *user* seperti pada gambar 2.1 secara formal, ketika dari permasalahan diolah menjadi *knowledge* dalam bentuk

representasi pengetahuan, maka akan melibatkan komputasi dalam menentukan *output*. *Output* yang diperoleh dari kalkulasi representasi pengetahuan, akan diinterpretasikan ke dalam solusi permasalahannya. Sehingga dapat disimpulkan bahwa ketika kita dapat membuat representasi pengetahuan dengan baik, maka representasi tersebut dapat menjadi *knowledge base* yang baik dalam menentukan solusi atau memberikan penalaran yang tepat dalam mengambil kesimpulan.

2.1.1.4.5. Teknik Inferensi *Backward Chaining* dan *Forward Chaining*

Menurut (sutojo at al 2011, p. 171) pada system pakar berbasis rule, domain pengetahuan dipresentasikan dalam sebuah kumpulan rule berbentuk IF-THEN, sedangkan data dipresentasikan 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.

1. *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 *IF*, maka data tersebut dieksekusi. bila sebuah rule dirksekusi maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam *data base*. Setiap kali pencocokan dimulai rule teratas, setiap rule hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa di eksekusi

2. *backward chaining* adalah suatu metode penarikan kesimpulan yang dimulai dari *goal state* lalu mundur kebelakang menggunakan fakta atau premis yang dimiliki.

2.2. Variable Penelitian

Menurut (Sugiyono, 2014, p. 63) Variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya. Secara teoritis variabel dapat didefinisikan sebagai atribut seseorang, atau obyek, yang mempunyai “variasi” antara satu orang atau obyek dengan obyek lain (Hatch dan Farhady, 1981 dalam (Sugiyono, 2014, p. 63). Variabel juga dapat merupakan atribut dari bidang keilmuan atau kegiatan tertentu. Tinggi, berat badan, sikap, motivasi, kepemimpinan, disiplin kerja, merupakan atribut-atribut dari setiap orang. Berat, ukuran, bentuk, dan warna merupakan atribut-atribut dari obyek. Struktur organisasi, model pendelegasian, kepemimpinan, pengawasan, koordinasi, prosedur dan mekanisme kerja, deskripsi pekerjaan, kebijakan, adalah contoh variabel dalam kegiatan administrasi. Dinamakan variabel karena ada variasinya. Variabel yang tidak ada variasinya bukan dikatakan sebagai variabel.

2.2.1 Penyakit *Rabies*

Rabies adalah penyakit karena infeksi virus yang sangat serius yang menargetkan otak dan sistem saraf. *Rabies* merupakan penyakit *encephalomyelitis*

yang progresif, akut, terutama menyerang hewan karnivora, kelelawar dan juga mamalia, penyakit sudah fatal ketika gejala klinis terlihat. *Rabies* atau penyakit Anjing Gila adalah penyakit *zoonosis* (penyakit yang ditularkan dari hewan ke manusia), menyerang hewan peliharaan, hewan liar, dan menyebar ke orang melalui kontak dekat dengan material infeksius, biasanya air liur, melalui gigitan atau cakaran.

Rabies adalah penyakit yang diabaikan di masyarakat dan dalam populasi yang rentan yang kematian (karena *rabies*) jarang dilaporkan. Hal ini terjadi terutama di masyarakat pedesaan terpencil di mana langkah-langkah untuk mencegah penularan. Dengan pelaporan *rabies* juga mencegah mobilisasi sumber penyakit dari masyarakat internasional melalui penghapusan media *rabies* pada manusia. Penyebab penyakit ini memiliki daya tarik kuat untuk menginfeksi jaringan saraf yang menyebabkan terjadinya peradangan pada otak atau *ensefalitis*, sehingga berakibat fatal bagi hewan ataupun manusia yang tertular. (Akoso, 2007)

Penularan virus *rabies* mudah ditularkan di antara mamalia, apakah mereka spesies yang sama atau berbeda. Virus ini biasanya menyebar dalam air liur, ketika hewan yang terinfeksi menggigit hewan lain. Jarang hewan atau manusia terinfeksi melalui kontak dengan air liur atau jaringan saraf terinfeksi, melalui selaput lendir atau luka di kulit. Virus rabies tidak menular melalui kulit utuh. Ada juga kejadian langka penularan melalui rute lainnya. Beberapa kasus telah dilaporkan setelah transplantasi organ, terutama kornea tetapi juga pankreas, ginjal dan hati. Transmisi aerosol telah didokumentasikan dalam keadaan khusus,

seperti di laboratorium dan gua-gua kelelawar dengan kepadatan yang sangat tinggi dari *aerosol*, partikel virus yang viabel. Virus *rabies* telah ditularkan melalui ingesti pada hewan percobaan yang terinfeksi, dan ada bukti anekdot bahwa penularan melalui susu terhadap domba dan bayi manusia. (Banyak rute konvensional dari penyebaran tidak bisa dikesampingkan dalam kasus terakhir). Ada beberapa spekulasi bahwa ingesti bisa memainkan peran dalam penularan *rabies* pada hewan liar.

Satu epidemi di kalangan kudu mungkin telah menyebar di antara hewan ketika mereka makan di pohon-pohon berduri. Tidak ada catatan dari penyakit manusia yang diperoleh dengan rute ini. Namun demikian, dalam 2 insiden yang telah diselidiki oleh *U.S. Centers for Disease Control and Prevention (CDC)*, orang yang minum susu yang tidak dipasteurisasi dari sapi *rabies* diberi pencegahan pasca pajanan. Susu pasteurisasi dan daging yang dimasak diharapkan tidak menimbulkan risiko infeksi, seperti virus *rabies* yang dilemahkan oleh panas, Namun, sebagai tindakan pencegahan, *National Association of State Public Health Veterinarians* menganjurkan tidak mengkonsumsi daging dan susu dari hewan terkena *rabies*.

Risiko utama dari *rabies* berasal dari kontak dengan sumber virus, air liur, cairan tubuh, atau jaringan hewan yang terinfeksi. *rabies* terpelihara dalam dua siklus epidemiologi, yaitu siklus *rabies* urban atau perkotaan dan siklus *rabies sylvatic*.

Siklus *Sylvatic* (atau satwa liar) siklus adalah siklus utama. Namun ini juga hadir bersamaan dengan siklus perkotaan di beberapa bagian dunia. Epidemiologi

dari siklus ini adalah kompleks, faktor yang mempengaruhi itu termasuk strain virus, perilaku spesies inang, ekologi dan faktor lingkungan. Dalam ekosistem manapun, sering satu dan kadang-kadang sampai 3 spesies satwa liar bertanggung jawab akan terlanggengkanya strain tertentu *rabies*. Penyakit patternin satwa liar dapat menjadi relatif stabil, atau terjadi sebagai epidemi yang bergerak lambat.

2.2.2 Infeksi Dan Penyebaran

Virus *rabies* memasuki tubuh melalui luka terbuka atau lecet kulit yang tercemar air ludah binatang yang menderita *rabies*. Melalui saraf-saraf kecil, virus menyebar dengan cara difusi dan memperbanyak diri di sepanjang saraf tersebut, kemudian naik ke saraf perifer menuju ke spinal kord dan otak. Di sini virus mengadakan difusi kedalam sistem saraf pusat kemudian menyebar secara sentrifugal ke bawah menuju saraf-saraf pusat (ganglion) berbagai organ. Hasil sekresi organ-organ juga akan menjadi infeksi jika ia berhubungan dengan sel-sel ganglion yang terinfeksi.

Pemeriksaan jenazah pada manusia yang meninggal akibat *rabies* maupun pada hewan-hewan yang mati akibat penyakit ini pada umumnya tidak menunjukkan perubahan-perubahan yang mencolok. Selain dijumpai meningoensefalitis dengan udem dan hiperemia, sistem saraf pusat juga mengalami perdarahan-perdarahan kecil. *Negri bodies* suatu badan berbentuk bulat kecil berukuran antara 0,25-25 μ , banyak dijumpai pada sitoplasma sel-sel saraf dan sel otak. Penularan terjadi melalui beberapa cara:

1. Kontak dengan binatang

Inokulasi virus *rabies* kedalam luka atau membran mukosa akan menyebabkan infeksi. Kulit yang utuh merupakan *barrier* bagi masuknya virus. Manusia biasanya diinfeksi melalui ludah yang mengandung virus *rabies* dan diinokulasi sewaktu digigit kucing.

2. Dari manusia ke manusia

Terdapat laporan terjadinya transmisi *rabies* melalui transplantasi kornea dan pembuluh darah dari orang yang telah terinfeksi.

3. Inhalasi

Infeksi melalui inhalasi terjadi akibat inhalasi udara yang mengandung virus *rabies* yang didapatkan dari *airbone* binatang yaitu kelelawar yang terinfeksi (jarang).

4. Vaksinasi

Infeksi terjadi karena vaksin mengandung virus *rabies* yang belum mati. Meskipun air liur binatang yang menderita *rabies* adalah infeksi, tetapi tidak setiap gigitannya selalu menimbulkan *rabies*. Hanya sekitar 5-15% penderita gigitan kucing atau binatang penderita *rabies* akan menjadi sakit.

2.2.3 Gejala-gejala *Rabies*

Gejala klinis dapat dibagi menjadi 3 fase umum: *prodromal*, eksitasi akut, dan *paralisa*. Namun ini adalah nilai praktis terbatas karena variabilitas gejala dan lama fase yang tidak teratur. Selama periode *prodromal*, yang berlangsung 1-3

hari, hewan menunjukkan tanda-tanda samar samar yang tidak spesifik, yang kemudian dengan cepat meregang. Penyakit tersebut berkembang dengan cepat setelah timbul kelumpuhan, dan kematian dipastikan terjadi beberapa hari sesudahnya. Beberapa hewan mati dengan cepat tanpa tanda-tanda klinis yang menciri.

Adanya Istilah “*furious rabies*” mengacu pada hewan di mana agresi (fase saraf eksitasi akut) terlihat. Adapun bentuk *rabies* ganas (*furious rabies*) yaitu,

1. Kucing mengalami kegelisahan perubahan sifat dan tingkah laku
2. Kucing lebih suka bersembunyi ditempat yang gelap
3. lebih agresif dan suka menyerang atau menggigit segala obyek terutama obyek bergerak
4. Akan mengalami kelumpuhan
5. Nafsu makan menjadi berkurang
6. Menjadi lebih sensitive terhadap suara dan cahaya
7. Memakan benda-benda asing seperti batu, kayu dll
8. akan tampak kejang-kejang dan akhirnya mati dalam waktu dekat

Sedangkan “*Dumb rabies*” mengacu pada hewan di mana perubahan perilaku yang tenang atau diam. Adapun bentuk *dumb rabies* yaitu,

1. Terjadi kelumpuhan dan ini merupakan gejala yang paling menonjol pada *rabies* bentuk tenang
2. Tenggorokan dan otot masseter menjadi lumpuh sehingga hewan tidak dapat menelاندan terjadi hipersaliva (mengeluarkan saliva yang berlebihan)
3. Kelupuhan pada laring menyebabkan suara hewan menjadi berubah (parau)

4. Terjadi kelupuhan pada bagian wajah dan rahang bawah
5. Terjadi ataksia dan inkoordinasi
6. Terjadi kejang kejang yang berlangsung singkat kadang tidak terlihat

Furious *rabies* adalah bentuk klasik "sindrom anjing gila", meskipun dapat terlihat pada semua spesies. Selama tahap ini tidak ada kelumpuhan, binatang menjadi pemarah, dengan sedikit provokasi, seringkali menyerang dengan agresif dan kejam menggunakan gigi, kuku, tanduk, atau kuku. Postur dan ekspresi merupakan satu tanda dari kewaspadaan dan kecemasan, dengan pupil melebar. Kebisingan/kegaduhan dapat mengundang serangan. Hewan seperti kehilangan kendali /hati-hati, takut pada manusia dan hewan lainnya. Karnivora dengan bentuk *rabies* ini sering berkeliaran kemana mana, menyerang hewan lain, termasuk manusia, dan obyek/ benda bergerak.

Mereka biasanya menelan benda asing, misalnya kotoran, jerami, tongkat, dan batu. Kucing *rabies* dapat menggigit gigit kawat dan kerangka kandang mereka, mematahkan gigi mereka sendiri, dan akan mengikuti bergerak tangan orang di depan kandang mereka mencoba untuk digigit. Anak kucing biasanya mencari persahabatan dengan orang dan terlalu lucu, tapi menggigit ketika orang membelainya, biasanya menjadi kejam dalam beberapa jam. Sigung *rabies* dapat mencari dan menyerang anak kucing. Kucing jinak maupun *bobcats* (kucing liar amerika utara) *rabies* dapat menyerang tiba-tiba, menggigit dan mencakar dengan kejam. Kemudian penyakit berkembang, umumnya kemudian terjadi inkoordinasi otot dan kejang. Hasilnya adalah terjadi kematian dari kelumpuhan yang progresif.

Sedangkan *dumbrabies* gejalanya terlihat seperti ataksia dan kelumpuhan otot-otot tenggorokan dan masseter, sering disertai air liur yang berlimpah dan ketidakmampuan untuk menelan, juga rahang bawah menggantung (jatuh) adalah umum pada kucing. Apabila pemilik sering memeriksa mulut kucing dan mencari benda asing atau memasukkan obat dengan tangan mereka (tanpa sarung tangan karet), inilah yang membuka kemungkinan untuk terpapar *rabies*. Hewan-hewan ini mungkin tidak ganas dan jarang mencoba untuk menggigit. Kelumpuhan tersebut berkembang dengan cepat ke seluruh bagian tubuh, koma dan dalam beberapa jam kemudian terjadi kematian. Yang perlu diwaspadai adalah apabila kucing terus mempunyai kecenderungan menggigit beberapa orang atau hewan lain sambil menunjukkan perangai aneh berarti kucing tersebut adanya penyakit *rabies*. (Soeharsono, 2007, p. 70)

2.2.4 Penanganan Penyakit Rabies

Terdapat beberapa penanganan untuk mengatasi penyakit *rabies* pada kucing, diantaranya :

1. Laporkan ke instansi terkait

Pelaporan bisa ke Dinas Peternakan dan Pertanian yang mana dinas tersebut akan melakukan observasi dan memeriksa kesehatan kucing tersebut selama 10 sampai dengan 14 hari ke depan.

2. Laboratorium

Jika selama observasi kucing mengalami hal yang tidak diinginkan maka tubuhnya akan dikirmkan ke lab untuk memastikan penyebab kematiannya.

3. Vaksin

Ketika observasi dilakukan dan hasilnya negatif *rabies* maka kucing tersebut akan di vaksinasi anti *rabies*. Setelah di vaksinasi kucing tersebut akan dikembalikan kepada pemiliknya. Namun jika kucing liar maka kucing tersebut akan diberikan kepada orang yang mau memeliharanya.

4. Penangkapan

Jika kucing *rabies* terus berusaha menggigit dan sulit untuk ditangkap maka hewan tersebut harus dibunuh kemudian diambil kepalanya untuk diperiksa lebih lanjut. Langkah ini adalah upaya pencegahan agar tidak semakin banyak hewan yang terkena *rabies*.

5. Waspadaai gigitannya

Jika kucing yang terkena *rabies* sulit ditemukan maka orang yang terkena gigitan tersebut harus mendapatkan perakuan khusus.

2.3. *Software* Pendukung

Software merupakan perangkat yang menjembatani interaksi user dengan komputer yang menggunakan bahasa mesin. *Software* juga dapat dikatakan adalah penggerak dan pengontrol *hardware* (perangkat keras). *Software* dibuat dengan menggunakan bahasa pemrograman yang ditulis atau diciptakan oleh *programmer* yang selanjutnya dikompilasi dengan aplikasi kompilasi sehingga menjadi sebuah kode yang nantinya akan dikenali oleh mesin *hardware*. (Daulay, 2007, p. 22)

2.3.1. UML (*Unified Modeling Language*)

UML merupakan bahasa visual untuk pemodelan sebuah system dengan menggunakan diagram serta teks pendukung. Menurut (A.S & Shalahuddin, 2013a, p. 137) *UML (Unified Modeling Language)* ditetapkan menjadi standarisasi bahasa pemodelan dalam pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemograman berorientasi objek. UML tercipta karena adanya kebutuhan pemodelan visual untuk membangun, menspesifikasikan, menggambarkan, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

Fungsi dari UML adalah untuk melakukan pemodelan. Jadi penggunaannya tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML cukup banyak digunakan pada metodologi berorientasi objek. Perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Bukan berarti pandangan yang berbeda dalam penggunaan UML adalah suatu yang dianggap

salah, Jika digambarkan secara awam, anggaplah UML bahasa yang kita gunakan sehari-hari, tetapi belum tentu penyampaian bahasa dalam bentuk puisi merupakan hal yang salah. (A.S & Shalahuddin, 2013a, p. 138)



Gambar 2.3 Logo UML

Sumber: Data Penelitian 2018

2.3.1.1. Use Case Diagram

Menurut (A.S & Shalahuddin, 2013a, p. 155) *use case* diagram 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. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* diantaranya:

1. Aktor merupakan orang, proses, atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* di deskripsikan sebagai fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor.

Tabel 2.1 Simbol Pada *Use Case Diagram* penelitian

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i>
Aktor / <i>actor</i>  Nama Aktor	Proses, orang atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
Ekstensi / <i>extend</i> << <i>Extend</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan, sesuai dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan milik nama depan yang sama dengan <i>use case</i> yang ditambahkan.
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / <i>include</i> / <i>uses</i> << <i>include</i> >>  << <i>uses</i> >> 	Relasi <i>use case</i> tambah 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 dijalankan <i>use case</i> ini.

Sumber: Rekayasa Perangkat Lunak 2013

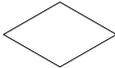
2.3.1.2. Activity Diagram

Menurut (A.S & Shalahuddin, 2013a, p. 161) diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem.

Diagram aktivitas juga digunakan untuk mendefinisikan hal-hal berikut:

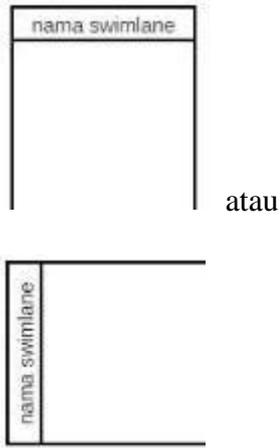
1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya.
4. Rancangan menu yang tampil pada perangkat lunak.

Tabel 2.2 Simbol-Simbol Pada *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

Sumber: Data Penelitian (2018)

Tabel 2.2 lanjutan

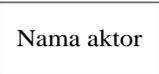
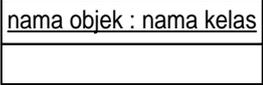
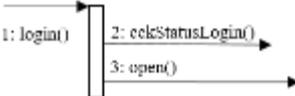
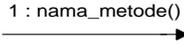
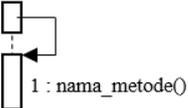
Simbol	Deskripsi
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Data Penelitian (2018)

2.3.1.3. *Sequence Diagram*

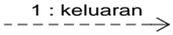
Menurut (A.S & Shalahuddin, 2013a, p. 165) diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang memiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut ini adalah simbol-simbol yang ada pada diagram *sequence*:

Tabel 2.3 Simbol-Simbol Pada *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama Aktor</p> <p>Atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya</p>  <p>maka <i>cekStatusLogin()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i> aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada ojek yang dibuat.</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>

Sumber: Data Penelitian (2018)

Tabel 2.3 Lanjutan

Simbol	Deskripsi
	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>creat</i> maka ada <i>destroy</i> .

Sumber: Data Penelitian (2018)

2.3.1.4. Class Diagram

Menurut (A.S & Shalahuddin, 2013a, p. 141) diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi, yaitu:

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau programmer dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

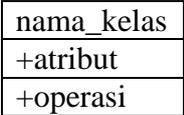
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut ini adalah simbol-simbol yang ada pada *class diagram*:

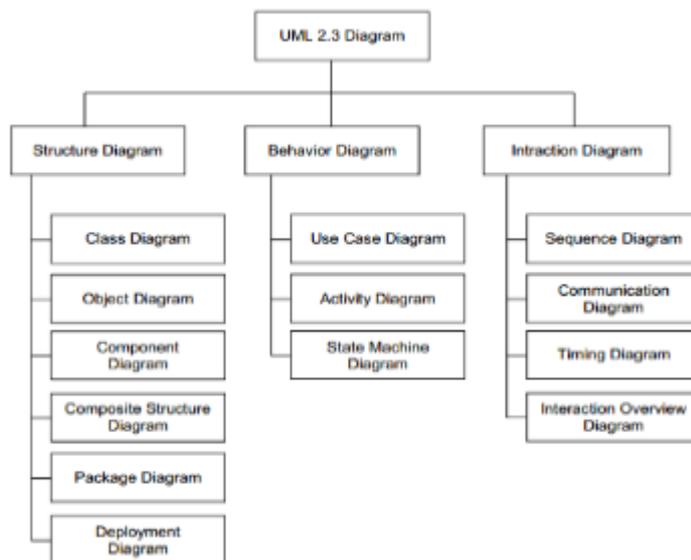
Tabel 2.4 Simbol-Simbol Pada *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur system
<p>Antarmuka / <i>interface</i></p>  <p>Nama_interface</p>	Sama dengan konsep interface dalam pemograman berorientasi objek
<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antar kelas
<p>Agregasi / <i>aggregation</i></p>	Relasi antarkelas dengan makna semua-bagian (<i>whole part</i>)

Sumber: Data Penelitian (2018)

2.3.1.5. Diagram UML

Menurut (A.S & Shalahuddin, 2013a, p. 140) diagram UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dari macam-macam diagram tersebut dapat dilihat pada gambar 2.15 dibawah ini.



Gambar 2.4 Diagram UML
Sumber: Rekayasa Perangkat Lunak 2013

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.2. StarUML

StartUML merupakan sebuah *software* pemodelan yang mendukung UML. Fitur-fitur yang ada pada *startUML*, contohnya seperti *usecase diagram*, *class diagram*, *sequence diagram*, *activity diagram*.



Gambar 2.5 Logo *StarUML*
Sumber: Data Penelitian, (2018)

2.3.3. PHP

PHP merupakan singkatan dari “*Hypertext Preprocessor*” yang digunakan sebagai bahasa *scrip server side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file* dan *folder*, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website*. Karena penggunaan PHP memungkinkan *web* dapat dibuat dinamis, *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP merupakan *software oper source* yang disebar dan dilisensikan secara gratis serta dapat di-*download* secara bebas dari situs resminya <http://www.php.net>. PHP ditulis menggunakan bahasa C. (KUNCIKOM, 2012, p. 5)



Gambar 2.6 Logo PHP
Sumber: Data Penelitian, (2018)

2.3.4. HTML

Menurut (Saputra, 2012, p. 1) HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengolah tampilan pada halaman *website*.

HTML versi 5 yang paling marak dibicarakan didunia maya. HTML 5 layaknya sebuah HTML yang sering digunakan dalam membangun aplikasi *web*, hanya saja HTML 5 ini memiliki keunggulan dibanding versi terdahulu menjadi lebih ringkas. Hal yang paling mencolok adalah tersediannya fitur baru seperti elemen multimedia, misalnya `<audio>` dan `<video>`, yang tak lain adalah fungsi untuk memutar audio dan juga video. Pada versi HTML sebelumnya, jika anda ingin memutar perangkat multimedia, haruslah menggunakan `<embed>`. Dengan adanya penambahan fitur dan tag khusus yang ada pada HTML 5, akan membuat semuanya menjadi sangat mudah. (Saputra, 2012)



Gambar 2.7 Logo HTML
Sumber: Data Penelitian (2018)

2.3.5. MySQL

Menurut (Saputra, 2012, p. 77) MySQL merupakan salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. MySQL bekerja dengan menggunakan bahasa SQL (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk memanipulasi *database*. Pada umumnya perintah yang paling sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau *query language* tersebut pada sub-sub selanjutnya.

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

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.

5. Memiliki dukungan (*group*) pengguna MySQL.
6. Lintas *Platform*, dapat digunakan pada berbagai sistem operasi berbeda.
7. *Multiuser*, dimana MySQL sapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.



Gambar 2.8 Logo MySQL
Sumber: Data Penelitian (2018)

2.3.6. XAMPP

Menurut (Dudung, 2016) XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.



Gambar 2.9 Logo XAMPP
Sumber: Data Penelitian (2018)

2.3.7 Bootstrap

Bootstrap merupakan *framework* css yang bisa digunakan untuk mempermudah membangun tampilan web. Pertama kali dikembangkan pada tahun 2010 oleh Mark Otto dan Jacob Thornton. Saat ini Bootstrap dikembangkan secara *open source* dengan lisenensi MIT



Gambar 2. 10 Logo Bootstrap
Sumber: Data penelitian (2018)

2.4. Penelitian Terdahulu

Penelitian terdahulu merupakan penelitian-penelitian yang dilakukan oleh para ahli terdahulu sebelum penelitian ini. Hasil penelitian-penelitian tersebut dijadikan referensi dalam penelitian ini, baik variabel-variabel terkait dan asumsi-asumsi yang relevan dari hasil penelitian tersebut. Sebagai bahan pertimbangan dalam penelitian ini, maka penulis mencantumkan beberapa penelitian yang diambil dari beberapa jurnal ilmiah, yaitu:

menurut (Teri Mangkarisnal dan Muhammad Zaki Rusti, (2016) **Sistem Pakar Diagnosa Penyakit Tanaman Hias Anthurium Menggunakan Metode Foward Chaining** (ISSN 2502-8758) peneliti mengatakan bahwa permintaan

tanaman hias semakin meningkat. Sehingga para petani berusaha untuk menghasilkan tanaman hias yang berkualitas dan dalam jumlah yang banyak. Tantangan yang dihadapi oleh para petani tanaman hias adalah penyakit yang menyerang pada ditanaman hias. Di antara beberapa tanaman hias yang ada, tanaman hias *anthurium* dijadikan sebagai objek penelitiannya. Dalam penelitian tersebut, peneliti berusaha membuat sistem pakar yang dapat mendiagnosis penyakit yang terdapat pada tanaman hias *anthurium*. Dengan adanya sistem pakar tanaman hias *anthurium*, para petani yang kurang pengetahuannya dengan tanaman hias tersebut bisa terbantu dengan adanya sistem pakar ini

Menurut (Tanzil, 2014) **Penyakit *rabies* dan penatalaksananya** (ISSN 2338-7793) Diperoleh fakta *Rabies* adalah infeksi virus akut yang menyerang sistem saraf pusat manusia dan mamalia dengan mortalitas 100%. Penyebabnya adalah virus *rabies* yang termasuk genus *Lyssavirus*, famili *Rhabdoviridae*. *Rabies* adalah penyakit zoonosis, penularan melalui jilatan atau gigitan hewan yang terjangkit *rabies* seperti anjing, kucing, kera, sigung, serigala, *raccoon* dan kelelawar. Walaupun telah tersedia vaksin *rabies* yang efektif dan aman bagi manusia dan hewan untuk pencegahan, sampai saat ini *rabies* masih menjadi masalah kesehatan diberbagai negara termasuk Indonesia. Tujuan penulisan peneliti ini untuk menjelaskan sifat-sifat virus *rabies*, patogenesis, gejala klinik, diagnosis, dan penatalaksanaannya. Dapat disimpulkan bahwa *rabies* adalah penyakit zoonosis yang menyerang sistem saraf pusat binatang menyusui dengan mortalitas 100%. Mortalitas *rabies* dapat dikurangi bila penyakit ini cepat diketahui dan disertai penatalaksanaan yang cepat dan tepat.

Menurut (Fahmi, I Wayan, I Made, 2017) **Perilaku dan Pemahaman Masyarakat Pemelihara Anjing terhadap Risiko Rabies di Kabupaten Karangasem, Bali** (ISSN 2477-6637) Rabies merupakan salah satu penyakit pada hewan yang bersifat zoonosis dan ditularkan melalui luka gigitan hewan terutama anjing yang terinfeksi rabies. Penyakit strategis nasional ini telah menular ke Kabupaten Karangasem, Bali. Penelitian ini bertujuan untuk mengetahui perilaku dan pemahaman masyarakat pemelihara anjing terhadap risiko rabies di Kabupaten Karangasem, Bali. Penelitian ini menggunakan 200 responden masyarakat pemelihara anjing di delapan desa Kabupaten Karangasem yang terbagi menjadi empat desa yang pernah dan empat desa yang belum dilaporkan terjadi kasus rabies. Kepada setiap responden ditanyakan sejumlah pertanyaan yang telah disiapkan berupa kuisioner. Hasil penelitian menunjukkan bahwa cara pemeliharaan anjing masyarakat desa yang belum dilaporkan terjadi kasus rabies 48% dilepas dengan persentase pengetahuan bahaya rabies dan ciri-ciri anjing rabies sebesar 78% , sementara masyarakat di desa yang pernah dilaporkan terjadi kasus rabies 71 % anjingnya dilepas dan 70% masyarakat pemelihara anjing mengetahui tentang bahaya rabies dan ciri-ciri anjing rabies. Berdasarkan hasil penelitian dapat disimpulkan bahwa pemeliharaan anjing masyarakat desa di Kabupaten Karangasem sebagian besar dilepas dan pengetahuan masyarakat tentang rabies baik, namun kesadaran terhadap risiko rabies masih kurang.

Menurut (Malang et al., 2017)**Sistem Pakar Untuk Mendiagnosa penyakit Gigi dan Mulut Dengan Metode *Forward Chaining* Berbasis Web** (ISSN 2303-1425) web diperoleh fakta bahwa Kesehatan memang menjadi barang

mahal bagi manusia, oleh karena itu butuh kepekaan pribadi untuk menjaganya. Salah satu organ tubuh yang sering lalai untuk dijaga adalah gigi dan mulut. Ini terbukti dari Data Dirjen Pelayanan Medik (2001) yang menunjukkan bahwa penyakit gigi dan mulut termasuk sepuluh ranking penyakit terbanyak di Indonesia (Saragih, 2009). Berdasarkan survey Yayasan Kesehatan Gigi Indonesia (2003) yang dilakukan pada anak-anak menunjukkan bahwa 70% anak-anak menderita karies gigi dan gingivitis (peradangan gusi), sedangkan pada orang dewasa ditemui sebanyak 73% yang menderita karies gigi. Dan, berdasarkan hasil survey kesehatan rumah tangga (SKRT) tahun 2004, sebanyak 90,05% penduduk Indonesia mempunyai masalah kesehatan gigi dan mulut (Saragih, 2009). Perancangan sistem pakar ini akan dibuat berbasis web melalui media php dengan basis data menggunakan mysql yang terdapat dalam paket zpanel. Perancangan sistem pakar ini dibangun dengan menggunakan penalaran – penalaran yang dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis yang disebut forward chaining dan perancangan sistem pakar ini pun dijelaskan melalui basis aturan yang merupakan rule yang terdiri dari dua bagian yaitu kondisi dan kesimpulan, basis pengetahuan yang merupakan inti program sistem pakar.

Menurut (Dadi Rosadi dan Asril Hamid 2014) **Sistem Pakar Diagnosa Penyakit Tanaman Padi Menggunakan Metode Forward Chaining** (ISSN2442-4943) peneliti mengungkapkan bahwa tanaman padi sering terjadi gagal panen yang disebabkan oleh terserangnya berbagai macam penyakit. Tidak hanya itu saja, peneliti menyebutkan bahwa terbatasnya pengetahuan para petani tentang penyakit padi dan kurangnya seorang ahli dibidang tersebut yang dapat

terjun langsung ke para petani. Dalam sistem pakar tersebut, metode inferensi yang digunakan adalah *forward chaining*. Dengan menggunakan sistem pakar diagnosis penyakit tanaman padi dengan menggunakan metode *forward chaining* dapat membantu para petani dalam menemukan penyakit yang terdapat pada tanaman padinya berdasarkan gejala-gejala yang dipilih. Sehingga para petani tidak perlu lagi menunggu seorang ahli tanaman padi untuk memberikan solusi terhadap penyakit yang ada pada tanamannya.

Menurut (M.S.Josephine, V.Jeyabalaraja, 2012) ***Expert System and Knowledge Management for Software Developer in Software Companies*** (ISSN2223-4985) *The Important work in the software companies are the development of the software and the software developer plays the important role in the developing process. Analyze the software developer to have the expert knowledge and develop the software. Developer must have the knowledge to define the problem, analysis, design and implementation. The every process must well defined by the developer. The developer entirely depends on the system and they have the expert solution of the system. The developer has communication every services using the SOA architecture. For the services and the expert knowledge to gain the developer, create the software developer optimizer. This optimizer will develop based on the knowledge base of the developer. This paper introduces the knowledge management process for an expert system that is the software developer optimizer (SDO). It is the design of expert system for the software developer process. Then the development stages are described. The SDO is an innovation since it is the first expert system*

developed for software developer It is a knowledge-based and SOA optimization system.

Menurut (Nana Yaw Asabere, Samuel Edusah Enguah, 2012) ***Integration of Expert Systems in Mobile Learning*** (ISSN2223-4985) *Sequel to advances and proliferations in Information and Communication Technologies (ICTs), Education Modes have changed from Traditional Face-to-Face (F2F) to different types such as Distance Learning, Electronic Learning (E-Learning) and Mobile Learning (M-Learning). This impact of ICT on education has resulted in technological usage in education. Desktop computers are used in ELearning while portable enabled-Wi-Fi devices such as Personal Digital Assistant (PDAs), Mobile Phones and Smartphones are used in M-Learning which allows learning/education to take place anywhere and anytime provided there is wireless network coverage in that particular area. Polytechnic Education in Ghana is practically-oriented and requires the influence of ICT due to large student numbers and reduced teacher to student ratio. The main objective of this paper is to analyze through a design-based research, the influence of M-Learning in Polytechnic Education in Ghana and how an Artificial Intelligence (AI) Expert System can be integrated in M-Learning to improve the productivity of a course entitled Computer Literacy 1 (CLT 101), which is a mandatory course for all admitted first year students of Ghanaian Polytechnics. After the research analysis, we proposed an expert system integrated in mobile learning to enhance teaching and learning of CLT 101 in Accra Polytechnic, Ghana. This study is in two folds. The first fold involves enhancing the mode of education and pedagogy in Accra Polytechnic through*

mobile learning and the second fold involves using the expert system to motivate and help students understand the course contents better and solve course problems with ease.

2.5. Kerangka Pemikiran

Model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting merupakan pengertian dari kerangka pemikiran (Sugiyono, 2014, p. 60). Agar penelitian ini lebih terarah dalam menyelesaikan masalah yang ada, maka dibutuhkan kerangka pemikiran.

Dari masalah yang telah diuraikan pada pembahasan sebelumnya, maka peneliti membuat kerangka pemikiran yang ditujukan untuk memberikan gambaran bagaimana proses kerja dari system pakar yang akan dibuat. Hal ini dianggap penting karena nantinya sistem pakar tidak hanya mendiagnosis penyakit *Rabies* pada kucing saja, akan tetapi juga akan menyediakan solusi untuk setiap permasalahan. Berikut ini adalah rancangan dari kerangka pemikiran dalam penelitian ini



Gambar 2.11 Kerangka Pemikiran
Sumber: Data penelitian (2018)