

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Teori Dasar**

##### **2.1.1 Kecerdasan Buatan atau *Artificial Intelligence***

Kecerdasan buatan berasal dari bahasa Inggris, yaitu *Artificial Intelligence* atau disingkat dengan AI. *Intelligence* jika diterjemahkan ke dalam bahasa Indonesia memiliki arti cerdas, sedangkan *artificial* adalah buatan atau tiruan. Jadi pada dasarnya kecerdasan buatan yang dimaksud adalah bukan manusia, melainkan mesin yang dapat berpikir, mengambil tindakan serta keputusan selayaknya manusia lakukan. (Hartati & Iswanti, 2008) berpendapat bahwa kecerdasan buatan merupakan cabang bidang ilmu komputer yang menggunakan mesin/komputer untuk berperilaku cerdas selayaknya manusia. Ilmu komputer yang dimaksud adalah perangkat keras dan perangkat lunak yang dikembangkan untuk penalaran, pemecahan masalah selayaknya manusia lakukan.

Cerdas berarti memiliki pengetahuan, penalaran, serta pengalaman dalam mengambil atau membuat keputusan maupun tindakan. Untuk membuat sebuah mesin yang cerdas, maka harus dibekali pengetahuan serta diberi juga kemampuan untuk menalar. Kecerdasan buatan termasuk bidang ilmu yang terbilang cukup baru, menurut (Jones, 2008) pada tahun 1950-an dipercaya sebagai awal mula kecerdasan buatan atau AI, yaitu saat awal-awal sistem komputer mulai dibangun dan ide-ide membangun mesin cerdas mulai terbentuk. Pada tahun 1950, Alan

Turing mulai memperkenalkan konsep mesin yang diciptakan yang beri nama mesin turing yang bisa menyelesaikan apa saja masalah matematika. Dalam membangun ide ini Turing masih menyimpan pertanyaan dalam pikirannya “apakah sebuah mesin memiliki kemampuan untuk berpikir”. Hasil percobaannya ini disebut dengan *Turing Test*. Dalam *Turing Test*, jika sebuah mesin dapat mengelabui seseorang yang berpikir mesin itu adalah manusia, maka mesin itu dianggap telah lulus dari tes kecerdasan (*intelligence test*).

Ide mengenai kecerdasan buatan itu sendiri dimunculkan oleh seorang professor dari *Massachusetts Insitute of Tecnology* yang bernama John McCarthy pada tahun 1956 pada *Dartmounth Conference* yang juga dihadiri oleh para peneliti seperti Marvin Minsky, Nathaniel Rochester, dan Claude Shannon. Pada konferensi itu juga difenisikan tujuan utama dari kecerdasan buatan, yaitu untuk memodelkan proses-proses pikir manusia serta memecahkan masalah dan mendesainnya kedalam bentuk mesin yang dapat meniru selayaknya manusia (Jones, 2008). Allen Newell dan Herbert Simon beserta rekan yang lain membuat program *AI* pertama pada tahun 1956 yang diberi nama *The logic theorist*. Berselang setahun setelah program *AI* pertamanya, Allen dan Simon membuat program *GPS (General Problem Solver)*, *GPS* menggunakan analisis rata-rata dalam memecahkan masalah namun masih terbatas pada masalah dalam permainan.

Pada awal-awal penciptaan *AI* hanya terfokus pada permainan dan pemecahan masalah secara umum, semenjak konferensi *AI* yang pertama diselenggarakan mulailah banyak konferensi lainnya mengenai *AI* dilaksanakan

hingga memunculkan berbagai macam model disiplin ilmu dari cabang AI itu sendiri. Pada awal tahun 80-an peneliti AI sukses secara komersial dari software jenis *Expert System*. Dan pada tahun 90-an hingga awal abad 21, AI meraih kesuksesannya setelah banyak digunakan secara luas pada industri teknologi. Kombinasi AI dengan bidang ilmu lainnya menciptakan beberapa macam model disiplin AI, beberapa diantaranya logika *fuzzy* (*fuzzy logic*), jaringan syaraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*) (Sutojo T & Suhartono, 2011).

#### **2.1.1.1 Fuzzy Logic**

Istilah *fuzzy* dalam kamus *oxford* memiliki arti *blurred* (kabur), *indistinct* (tidak jelas), *confused* (membingungkan) (Putri & Effendi, 2017). Logika *fuzzy* merupakan suatu metode pengontrolan sistem untuk pemecahan masalah dan diimplementasikan pada sistem, yang digunakan oleh sistem yang sederhana, *embedded system*, jaringan komputer, *workstation* berbasis akuisisi data, hingga sistem kontrol. Dalam logika *fuzzy* dinyatakan bahwa sesuatu yang bersifat biner 0 dan 1 memiliki dua kemungkinan, “Ya dan Tidak”, “Benar atau Salah”. Logika *fuzzy* dapat diterapkan di beberapa bidang seperti dalam sistem mendiagnosis penyakit (bidang kedokteran); pemodelan sistem pemasaran (bidang ekonomi); prediksi adanya gempa bumi dan klasifikasi pencocokan pola (dalam bidang teknik) (Sutojo T & Suhartono, 2011).

Keuntungan yang didapat menggunakan *fuzzy logic* untuk memecahkan masalah menurut (Sutojo T & Suhartono, 2011) :

1. Tidak memerlukan persamaan matematik yang rumit karena perancangan yang mudah
2. Mudah dipahami
3. Toleransi pada data yang tidak tepat
4. Kompleks dalam memodelkan fungsi-fungsi nonlinear
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa melalui proses pelatihan
6. Bekerja sama dengan teknik-teknik kendali secara konvensional
7. Berasal dari bahasa alami

#### **2.1.1.2 Jaringan Syaraf Tiruan (*Artificial Neural Network*)**

Jaringan syaraf tiruan adalah model dalam pengelolaan informasi yang terinspirasi dari sistem saraf secara biologis, seperti pada saat proses penyampaian informasi pada otak manusia. Komponen utamanya adalah struktur dari sistem pengolahan informasi yang memiliki beberapa elemen pemrosesan yg saling terhubung(*neuron*), yang bertugas untuk menyelesaikan masalah. Cara kerja dari jaringan syaraf tiruan ini sama seperti cara kerjanya pada manusia, yaitu belajar melalui contoh. Sejumlah model pengimplementasian jaringan syaraf tiruan seperti implementasi pada bidang kedokteran, yaitu diagnosis sistem

kardiovaskular, hidung elektronik, dan dokter instan; pada bsektor bisnis, jaringan syaraf tiruan diintegrasikan pada merek dagang *The Airline Marketing Tactician (AMT)* menggunakan *backpropagation* yang dapat mengontrol penjualan pendistribusian kursi penerbangan (Sutojo T & Suhartono, 2011).

Sejumlah kelebihan yang ada pada jaringan syaraf tiruan antara lain (Sutojo T & Suhartono, 2011):

1. Belajar adaptif, yaitu kecakapan untuk mendalami bagaimana melaksanakan suatu pekerjaan bersumber pada data yang diperoleh untuk pelatihan atau pengalaman awal.
2. *Self-Organization*, yaitu kemahiran dalam membentuk organisasi sendiri atau gambaran dari informasi yang diterima.
3. *Real Time Operation*, yaitu perhitungan yang dibuat bentuk paralel hingga *hardware* yang dipersiapkan dan dibuat secara khusus dapat mengambil manfaat dari kemampuan ini.

Selain memiliki sejumlah kelebihan, jaringan syaraf tiruan juga mempunyai kelemahan, yaitu (Sutojo T & Suhartono, 2011):

1. Tidak efisien dalam mengerjakan operasi-operasi numerik dengan akurasi tinggi.
2. Tidak efektif untuk mengerjakan operasi algoritma aritmatika, operasi logika, dan simbolis.
3. Memerlukan instruksi supaya dapat beroperasi sehingga apabila memiliki jumlah data yang besar, waktu yang diperlukan untuk proses pelatihan sangat lama.

### **2.1.1.3 Sistem Pakar (*Expert System*)**

Sistem pakar mulai diperluas pada pertengahan tahun 1960 yang di indikasikan dengan terciptanya sistem pakar pertama yang dikembangkan oleh Newel dan Simon bernama *General-purpose Solver* (GPS). Kemudian mulailah bermunculan sistem pakar lainnya diberbagai bidang seperti MYCIN untuk diagnosis penyakit, DENDRAL untuk identifikasi struktur molekul campuran yang tidak diketahui, XCON & EXSEL untuk membantu konfigurasi sistem komputer besar, SOPHIE untuk menemukan deposit, FOLIO untuk membantu memberikan keputusan bagi seorang manajer dalam masalah stok dan investasi, dan DELTA dipakai untuk pemeliharaan lokomotif listrik diesel (Sutojo T & Suhartono, 2011).

Menurut (Kusumadewi, 2003) sistem pakar adalah usaha mengadopsi pengetahuan manusia ke komputer agar dapat menyelesaikan masalah dan mengambil keputusan seperti dilakukan oleh para ahli. Sistem pakar dirancang untuk dapat menyelesaikan masalah tertentu dengan meniru kerja si pakar. Dengan sistem pakar orang awam dapat menyelesaikan masalah dan bagi para ahli, sistem pakar juga membantu aktivitasnya sebagai asisten yang berpengalaman. Sedangkan menurut suatu sistem yang dirancang untuk meniru keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah (Sutojo T & Suhartono, 2011).

( Mulawarman, & Honggowibowo, 2009) berpendapat bahwa sistem pakar ditampilkan dalam dua lingkungan, yaitu: pengembangan dan konsultasi. Lingkungan pengembangan digunakan sistem pakar dalam membangun komponen serta memasukkan pengetahuan ke dalam basis pengetahuan, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan ahli untuk memperoleh informasi, pengetahuan dan juga berkonsultasi.

Pakar adalah seseorang yang mempunyai pengetahuan khusus, pemahaman wawasan, serta teknik dalam menyelesaikan suatu masalah pada bidang tertentu. Seorang pakar perlu mempunyai kemampuan kepakaran seperti mengidentifikasi dan mendeskripsikan suatu masalah dan juga memecahkan masalah dengan akurat, mendeskripsikan solusi dari masalah, restrukturisasi pengetahuan, mengetahui batas kemampuan, serta keterampilan dalam mengaplikasikan pengetahuannya dan membantu memberi solusi dan pemecahan masalah pada bidang tertentu (Hartati & Iswanti, 2008)

Sebuah sistem dapat disebut sebagai sistem pakar jika memiliki ciri-ciri sebagai berikut (Sutojo T & Suhartono, 2011):

1. Terpaku pada ranah keahlian tertentu
2. Mengajukan penalaran untuk data-data yang tidak cukup atau samar
3. Menggambarkan kausa-kausa dengan model yang dapat dipahami
4. Bertindak bersumber pada aturan tertentu
5. Mudah ditransformasikan
6. Basis pengetahuan dan teknik inferensi diposisikan terpisah
7. Keluarannya (*output*) bersifat anjuran

8. Sistem dapat mengoperasikan aturan secara terpisah secara searah, serasi dengan dialog dan pengguna

Representasi pengetahuan adalah model untuk mengodekan pengetahuan pada suatu sistem pakar yang berbasis pengetahuan. Hal ini bertujuan untuk mengikat ciri-ciri penting dari suatu masalah agar informasi tersebut bisa diakses oleh mekanisme pemecahan masalah. Bahasa representasi dibuat supaya fakta-fakta serta pengetahuan lain yang ada didalamnya bisa dipakai sebagai penalaran (Kusrini, 2006).

Menurut (Hartati & Iswanti, 2008) representasi pengetahuan bertujuan untuk mengintegrasikan pengetahuan dalam bentuk struktur tertentu sehingga dapat dipahami oleh komputer. Pemilihan representasi pengetahuan akan menciptakan sebuah sistem pakar yang efisien. Salah satu model representasi pengetahuan yaitu kaidah produksi (*production rule*).

Sistem pakar pada penelitian ini menerapkan bentuk representasi pengetahuan berdasarkan pada kaidah produksi. Menurut Firebaugh (1998) dalam (Hartati & Iswanti, 2008) struktur sistem pakar berbasis kaidah produksi terdiri dari 4 komponen yaitu:

1. Antar Muka Pengguna ( *User Interface* )

Menurut (Kusrini, 2006) antarmuka pengguna adalah penyambung antara pengguna dengan sistem pakar. Bagian tersebut berguna seperti alat komunikasi antara sistem dan *user*. Bagian ini juga dibuat serupa itu agar efektif dan mudah

pemakaiannya terlebih untuk pengguna yang tidak mahir pada bagian yang diterapkan pada sistem pakar (Hartati & Iswanti, 2008).

## 2. Basis Pengetahuan

Basis pengetahuan merupakan bagian yang terdiri dari sejumlah aturan yang berasal dari pengetahuan pada ranah tertentu dan umumnya diperlihatkan dalam bentuk aturan produksi (*IF...THEN...*). Pengetahuan dari si pakar selanjutnya dipersembahkan pada bentuk tertentu yang terdapat dalam sejumlah pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti buku-buku, jurnal ilmiah, majalah, maupun dokumentasi tercetak lainnya. Basis pengetahuan diposisikan terpisah dari mesin inferensi supaya peningkatan pengetahuan sistem pakar bisa dibuat tanpa harus mengganggu mesin inferensi (Hartati & Iswanti, 2008).

Ada dua bentuk pendekatan basis pengetahuan yang umum digunakan, yaitu (Kusumadewi, 2003):

### a. Penalaran berbasis aturan (*Rule-Based Reasoning*)

Penalaran berbasis aturan yang pengetahuannya direpresentasikan dalam bentuk aturan *IF-THEN* dan digunakan ketika memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan pakar dapat menyelesaikan masalah tersebut secara berurutan. Bentuk ini juga digunakan apabila membutuhkan penjelasan langkah untuk mencapai solusi.

b. Penalaran berbasis kasus (*Case-Based Reasoning*)

Penalaran berbasis kasus pada dasarnya berisi solusi-solusi yang sudah dicapai sebelumnya, dan selanjutnya akan diturunkan suatu bentuk solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Biasanya bentuk ini digunakan saat pengguna menginginkan untuk tahu lebih banyak suatu kasus yang hampir mirip. Dan bentuk ini digunakan juga pada saat memiliki keadaan, atau kasus dalam basis pengetahuan.

3. Struktur kontrol (Mesin Inferensi)

Struktur kontrol adalah *interpreter* aturan atau mesin inferensi mengaplikasikan pengetahuan-pengetahuan yang tersimpan dalam basis pengetahuan yang bertujuan untuk mengatasi permasalahan yang ada. Inferensi adalah prosedur untuk memproduksi suatu informasi bermodelkan konklusi logis yang didapatkan berasal dari informasi yang tersedia atau fakta yang diketahui (Kusrini, 2006).

Prosedur inferensi sistem pakar membutuhkan proses pengujian aturan-aturan dalam barisan tertentu dan bertujuan menelusuri sebuah situasi yang tepat dengan situasi awal dan untuk menegaskan situasi yang sedang berlangsung dan disinkronisasika kedalam *database*. Prosedur ini disebut dengan perunutan atau penalaranm, yaitu proses pencocokkan bukti atau situasi tertentu yang tersimpan dalam basis pengetahuan dan memori kerja dan dijabarkan dalam premis atau bagian kondisi pada suatu aturan.

Ada sejumlah model penalaran yang bisa dipakai oleh mesin inferensi yaitu:

a. Penalaran Maju (*Forward Chaining*)

Rancangan ini disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses penalaran dimulai dari premis-premis atau informasi masukan (*IF*) dahulu lalu menuju konklusi atau *derived information* (*THEN*), *Forward chaining* juga disebut pencocokkan fakta atau pernyataan dimulai dari sebelah kiri dahulu. Rancangan ini dapat dibentuk sebagai berikut:

*IF* ( informasi masukan)

*THEN* ( konklusi)

Informasi anjuran berbentuk suatu pengamatan sedangkan konklusi berbentuk diagnosa jadi dapat dijabarkan jalannya penalaran runut maju dimulai dari pengamatan menuju diagnosa. Pada metode ini, sistem tidak melakukan praduga terhadap konklusi, namun sistem menerima semua gejala yang diberikan dan selanjutnya memeriksa gejala-gejala tersebut serta mencocokkan dengan konklusi yang tepat (Hartati & Iswanti, 2008).

b. Penalaran Mundur (*Backward Chaining*)

Secara umum, rancangan ini diimplementasikan saat tujuan sudah ditentukan sebagai kondisi atau keadaan awal. Disebut juga konsep *goal-driven search*. Arah penalaran pada model ini berlawanan dengan *forward chaining* karena pencocokkan fakta pada *backward chaining* dimulai dari sebelah kanan atau *THEN*. Konsep ini dapat dimodelkan sebagai berikut:

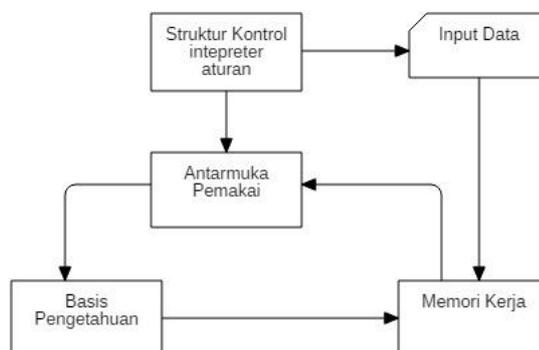
Tujuan

*IF* (kondisi)

Proses penalaran *backward chaining* dimulai dari tujuan lalu memeriksa balik ke jalur yang mengarah ke tujuan tersebut. Untuk memperlihatkan bagian situasi pada aturan sudah terpenuhi. Proses *internal* akan mengecek konklusi (tujuan) terlebih dulu selaku praduga awal, selanjutnya mengecek dan memastikan gejala-gejala (kondisi) sudah terpenuhi dan menghasilkan konklusi sebagai keluaran. Apabila sistem mendapati bagian kondisi tidak terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada kaidah selanjutnya (Hartati & Iswanti, 2008).

#### 4. *Working memory* (memori kerja)

Berguna untuk menuliskan situasi permasalahan yang terjadi serta *history* solusi. Dan membentuk bagian yang berisi fakta-fakta masalah yang akan dijumpai pada suatu sesi saat prosedur konsultasi berlangsung (Kusrini, 2006).



**Gambar 2.1** Struktur Sistem Pakar Aturan Produksi

**Sumber:** (Kusrini, 2006)

Dari gambar dapat dijelaskan bahwa kaidah atau aturan menjabarkan cara formal dan dituliskan dalam model jika-maka (*IF-THEN*) untuk mencerminkan arahan, rekomendasi, serta strategi. Aturan *IF-THEN* merangkaikan antarsenden (*antecedent*) dengan konsekuensi yang diakibatnya. Contoh struktur kaidah *IF-THEN* yang menghubungkan obyek menurut Adedeji, 1992 dalam (Hartati & Iswanti, 2008)

1. *IF* premis *THEN* konklusi
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* gejala *THEN* diagnosa

Premis mengarah kepada bukti yang benar sebelum konklusi tertentu diperoleh. Masukan mengarah pada data yang ada sebelum keluaran diperoleh. Situasi membentuk kepada bentuk yang benar sebelum langkah diambil. Antesenden mengacu ke kondisi yang terjadi sebelum konsekuensi diamati. Data harus mengacu pada informasi yang tersedia sehingga hasil diperoleh.

Tindakan mengarah ke aktivitas yang harus dilakukan sebelum hasil dapat diharapkan Aksi mengara kepada aktivitas yang menyebabkan efek dari tindakan tersebut. Serta gejala mengarah pada kondisi yang mengakibatkan kerusakan atau

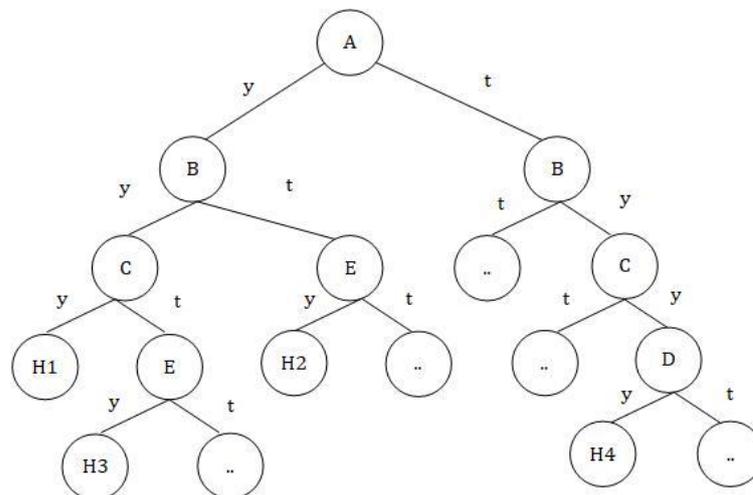
kondisi tertentu yang menggerakkan pemeriksaan (diagnosa) (Hartati & Iswanti, 2008).

Sebelum mencapai pada model aturan produksi, pengetahuan yang diperoleh dari ranah tertentu selanjutnya ditampilkan dalam model tabel keputusan dan dibuat pohon keputusannya. Contoh penyajian bentuk tabel keputusan dan pohon keputusan.

**Tabel 2.1** Keputusan

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

**Sumber:** Hartati dan Iswanti (2008)



**Gambar 2.2** Pohon Keputusan  
**Sumber:** (Hartati & Iswanti, 2008)

Keterangan:

A = *evidence A*, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, \*\* = tidak menghasilkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Gambar 2.2 menjelaskan bahwa hipotesis H1 akan terpenuhi jika *evidence* A, B, dan C terpenuhi. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan E. Hipotesa H3 akan terpenuhi apabila memiliki *evidence* A, B, dan E. Dan hipotesa H4 dihasilkan apabila memenuhi *evidence* B, C, dan D. Notasi “y” memiliki arti memenuhi *node (evidence)*, sedangkan notasi “t” artinya tidak memenuhi.

Tahapan konsultasi pada sistem pakar, *node-node* yang menggantikan *evidence* umumnya berbentuk pertanyaan yang diberikan oleh sistem. Terlihat dari pohon keputusan pada gambar 2.2 permasalahan biasanya terjadi pada awal sesi konsultasi, yaitu sistem pakar akan menanyakan “apakah memiliki *evidence* A? Intinya adalah segala jawaban dari pengguna baik “ya” maupun “tidak” maka sistem akan melanjutkan pertanyaan pada *evidence* B. Ini berarti jawaban pengguna tidak mempengaruhi sistem

Menurut (Kusumadewi, 2003) ada banyak manfaat yang dapat dirasakan dengan adanya sistem pakar, antara lain:

1. Memungkinkan orang awan mengerjakan pekerjaan para pakar
2. Dapat melakukan proses berulang-ulang dan otomatis
3. Dapat menyimpan pengetahuan dari para pakar
4. Dapat beroperasi dalam lingkungan yang berbahaya
5. Memiliki reliabilitas

6. Menghemat waktu dalam pengambilan keputusan
7. Kapabilitas dalam penyelesaian masalah

Disamping memiliki beberapa keuntungan, Sistem pakar juga memiliki beberapa kelemahan antara lain (Kusumadewi, 2003):

1. Biaya pemeliharaan yang mahal
2. Sulit dikembangkan, Karena berkaitan erat dengan ketersediaan pakar pada bidangnya
3. Sistem pakar tidak 100% benar

## **2.2 Variabel**

(Sugiyono, 2014) Variabel penelitian adalah sesuatu yang berupa apapun, obyek ataupun kegiatan dan memiliki bentuk tertentu yang sudah ditentukan oleh peneliti untuk seterusnya dipahami sehingga memperoleh informasi atau data mengenai hal tersebut, dan kemudian diambil kesimpulannya. Dapat ditarik kesimpulannya bahwa variabel dari penelitian ini adalah jenis tingkatan bipolar disorder.

### **2.2.1 Tingkatan Bipolar Disorder**

Menurut (Sylvia & Gitayanti, 2010) terdapat 3 jenis bipolar yaitu:

1. Episode Manik ( Bipolar Disorder I)

a. Gangguan Afektif Bipolar Manik Dengan Gejala Psikotik

Merupakan bentuk mania yang berat dari keadaan yang digambarkan, peningkatan percaya diri yang membumbung dan gagasan kebesaran yang dapat berkembang menjadi waham dan iritabilitas, serta kecurigaan yang waham (gangguan pada isi pikiran). Percepatan cara berbicara yang menyebabkan seseorang tidak dapat memahami lagi. Aktivitas serta eksitasi fisik yang berlebihan secara terus menerus menyebabkan agresi dan kekerasan; mengabaikan aktivitas makan dan minum yang berakibat dehidrasi dan kelalaian diri yang berbahaya. Adapun gejala bipolar manik dengan gejala psikotik seperti peningkatan mood yang ekspansif, iritabel, dan abnormal yang berlangsung selama 1 minggu, suka berbicara, kebutuhan tidur yang berkurang, meningkatnya kepercayaan diri atau grandiositas, distraktilitas atau ketidakmampuan individu untuk berkonsentrasi dan mudah beralih, meningkatnya aktivitas seksual, serta mengalami halusinasi atau waham (Sylvia & Gitayanti, 2010).

b. Gangguan Afektif Bipolar Manik Tanpa Gejala Psikotik

Keadaan suasana perasaan meningkat tinggi tidak sepadan dengan keadaan individu, serta dapat bervariasi antara kesenangan (seakan-akan bebas dari masalah) hingga keadaan eksitasi yang tidak terkendali. Eliasi (*mood* yang meningkat tajam) itu disertai dengan energi yang meningkat hingga mengakibatkan aktivitas berlebihan, dan seringkali perhatian saat mudah dialihkan. Adapun gejala bipolar manik tanpa gejala psikotik seperti episode berlangsung sedikitnya 1 minggu, cukup berat hingga mengacaukan hampir seluruh pekerjaan

dan aktivitas sosial, peningkatan energi atau overaktivitas, suka berbicara, kebutuhan tidur yang berkurang, distraktilitas, dan gemar berfoya-foya (Sylvia & Gitayanti, 2010).

## 2. Episode Depresif ( Bipolar Disorder II)

### a. Episode Depresi Ringan atau Sedang

Dalam mendiagnosis depresi digunakan kriteria DSM-IV-TR. Selain itu, di Indonesia terdapat juga pedoman diagnosis menurut PPDGJ III. Dalam episode depresif ringan atau sedang harus ada sekurangnya satu episode hipomanik, manik, ataupun campuran dimasa lampau. Adapun gejala yang terjadi pada episode depresif ringan seperti kehilangan minat dan kesenangan, mudah lelah, mengalami *mood* yang depresif, nafsu makan menurun, dan mengalami kesulitan dalam pekerjaan dan sosial, terjadi sekurangnya dua minggu (Maslim, 2003).

### b. Episode Depresif Berat Tanpa Gejala Psikotik

Pada episode ini sekurangnya ada 3 gejala utama, dan 4 dari gejala lainnya harus berintensitas berat kecuali agitasi atau retardasi yang mencolok dan berlangsungnya kurang lebih dua minggu atau bias lebih pendek. Adapun gejala yang dialami pada episode ini seperti kehilangan minat, mengalami perasaan diri tidak berguna, mengalami sindrom somatik, pernah mengalami episode manik, hipomanik atau campuran dimasa lampau, memiliki niatan bunuh diri pada kasus berat, serta mengalami semua gejala depresi (Maslim, 2003).

### c. Episode Depresif Berat Dengan Gejala Psikotik

Episode ini harus memenuhi kriteria F32.2 dengan disertai waham dan halusinasi. Biasanya waham yang dialami berupa tentang dosa, malapetaka yang mengancam individu dan penderita merasa bertanggung jawab pada hal tersebut. Gejala yang dialami pada episode ini seperti mengalami semua gejala depresif, mengalami waham dan halusinasi, mengalami halusinasi auditorik (halusinasi pendengaran), olfaktorik (halusinasi penciuman), terjadi retardasi motorik yang mengakibatkan stupor (penurunan kesadaran), mengalami afek serasi dan tidak serasi (Sylvia & Gitayanti, 2010).

### 3. Bipolar Campuran

Pada episode ini sekurangnya mengalami satu episode afektif manik, hipomanik dimasa lampau. Dan dimasa kini menunjukkan gejala-gejala manik, hipomanik, serta depresi secara berulang atau bergantian dengan cepat. Gejala yang terjadi seperti dalam beberapa hari atau minggu menjalani aktivitas berlebihan, suka berbicara cepat, menurunnya aktivitas seksual, mengalami gejala depresif dan hipomania secara bergantian dari hari ke hari, tiap episode yang terjadi sama-sama mencolol selama kurang lebih dua minggu (Maslim, 2003).

---

**Tabel 2.2** Gejala

<b>Variabel</b>	<b>Indikator</b>
Jenis Bipolar Disorder	1. Bipolar Tipe I 2. Bipolar Tipe II 3. Bipolar Campuran

**Sumber:** Data penelitian, 2019

---

## **2.3 Software Pendukung**

*Software* pendukung adalah sejumlah perangkat lunak yang akan dipakai untuk membantu pembuatan sistem pakar dalam penelitian ini. Ada sejumlah perangkat lunak yang dipakai antara lain: *XAMPP*, *phpMyAdmin*, *PHP*, *HTML*, *CSS*, *MySQL*, *Notepad++*, dan *StarUML*.

### **2.3.1 XAMPP**

(Sidik & Husni, 2014) *Server web* adalah komputer yang berguna sebagai penyimpan dokumen-dokumen berbentuk *web* serta mengoperasikan permintaan dokumen *web* dari kliennya. *XAMPP* adalah perangkat lunak berbasis *web server apache* yang ada pada *database server MySQL* yang juga membantu pemrograman *PHP*. *XAMPP* juga adalah *software* yang mudah dipakai, gratis (*free*), dan juga *support* pada sistem operasi *Linux* dan *Windows*, serta tersedia *Apache Web Server*, *MySQL Database Server*, *PHP support (PHP 4 dan PHP 5)* dan modul lainnya.

### **2.3.2 PhpMyAdmin**

Dikutip dari situs resminya *phpMyAdmin* adalah perangkat lunak gratis yang ditulis dalam *PHP*, yang bertujuan untuk menangani administrasi *MySQL* melalui *web*. *phpMyAdmin* mendukung berbagai operasi pada *MySQL* dan

*MariaDB*. Operasi yang sering digunakan (mengelola basis data, tabel, kolom, relasi, indeks, pengguna, perizinan, dll) dapat dilakukan melalui *interface* pengguna. *PhpMyAdmin* adalah proyek yang matang dengan basis kode yang stabil dan fleksibel serta merupakan bagian dari *Software Freedom Conservancy* (*SFC*). *SFC* adalah organisasi nirlaba yang membantu mempromosikan, meningkatkan, mengembangkan, dan membela proyek-proyek *Free, Libre, dan Open Source Software* (*FLOSS*).

### **2.3.3 Hypertext Preprocessor (PHP)**

Menurut (Mundzir, 2018) *hypertext preprocessor* atau dikenal *PHP* merupakan bahasa pemrograman *universal* untuk penanganan serta pengembangan situs *web* yang dapat digunakan secara bersamaan dengan *HTML*. Kode *PHP* diinterpretasikan pada *web server* dan menciptakan *HTML* atau keluaran lainnya yang bisa diamati oleh pengunjung. Bahasa pemrograman *PHP* banyak digunakan untuk membuat situs *web* dinamis. Contoh aplikasi program *PHP* adalah forum (*phpBB*) dan mediaWiki (*software* dibelakang *Wikipedia*). *PHP* juga menjadi pilihan untuk membuat atau menjalankan program seperti *ASP.NET/C#/VB.NET, Microsoft, dan CGI/PERL*.

*PHP* juga merupakan sekumpulan skrip yang memiliki fungsi utama untuk mengumpulkan dan mengevaluasi hasil dalam bentuk apapun ke *server database* serta mengumpulkan dan mengelompokkan informasi yang kemudian ditampilkan

pada saat tamu *website* memerlukannya (menampilkan informasi yang ingin diakses sesuai permintaan *user*)

#### **2.3.4 HTML (Hyper Text Markup Language)**

*Hyper text markup language* atau *HTML* merupakan bahasa paling dasar dan penting yang dipakai untuk mengelola serta menampilkan halaman *website*. *HTML* juga berfungsi untuk menyajikan bermacam informasi didalam sebuah penjelajah *web* internet dan formatting *hypertext* sederhana yang ditulis dalam format ASCII untuk menciptakan bentuk yang menyatu dan *HTML* menggunakan dua jenis ekstensi file yaitu *.htm* dan *.html*. (Saputra, 2012).

Elemen adalah istilah untuk bentuk-bentuk dasar dalam menciptakan dokumen *HTML*. *Tag HTML* berfungsi sebagai tanda berbagai komponen dalam suatu dokumen *HTML*. Elemen dasar *HTML* yang dibutuhkan dalam pembuatan suatu dokumen *HTML* dimulai dengan *tag<html>*, *<head>*, dan *<body>* berikut *tag-tag* pasangannya yaitu *</html>*, *</head>*, *</body>*. Setiap dokumen *HTML* mempunyai pola dasar sebagai berikut (Sidik & Husni, 2014):

1. Setiap dokumen *HTML* dimulai dengan mencantumkan *tag<html>* dan *tag</html>* diakhir dokumen. *Tag* berguna sebagai penanda komponen *html* yang bermakna dokumen tersebut ialah dokumen *HTML*.
2. Komponen *head* diawali menggunakan *tag<head>* dan *tag</head>*. Komponen ini menyimpan informasi mengenai dokumen *HTML*. Yang

dicatatkan pada komponen ini adalah judul dokumen yang ditandai dengan *tag<title>* dan diakhiri *tag</title>*.

3. Elemen *body* diawali dengan *tag<body>* dan diakhiri *tag</body>*. Ini merupakan komponen paling besar didalam dokumen *HTML* karena berisi inti dokumen yang akan diperlihatkan di *browser* yang melibatkan *paragraph*, grafik, *link*, tabel, dan lainnya.

### 2.3.5 CSS (*Cascading Style Sheet*)

*Cascading Style Sheet* atau dikenal *CSS* adalah bahasa pemrograman web yang dibuat eksklusif untuk membangun dan mengendalikan bermacam-macam elemen pada web agar bentuk web terlihat lebih rapi, terstruktur, dan seragam. Tujuan *CSS* ini adalah untuk mengurai konten utama dengan tampilan dokumen lain dan untuk mempercepat pembuatan halaman web (Saputra, 2012).

Beberapa kelebihan memakai *CSS* adalah sebagai berikut (Saputra, 2012):

1. Mengurai pembuatan dokumen (*CSS* dan *HTML*)
2. Memudahkan serta mempercepat dalam pembuatan dan pemeliharaan dokumen *web*
3. Lebih cepat dalam mengakses *web*
4. Fleksibel, interaktif, dan tampilan lebih menarik
5. Ukuran file lebih kecil sehingga *bandwidth* yang digunakan otomatis juga lebih kecil
6. *Useable* pada semua *browser*

### 2.3.6 MySQL

*MySQL* merupakan sistem manajemen *database SQL* yang bersifat *open source* (terbuka) dan paling banyak digunakan. Sistem *database MySQL* mendukung fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system (DBMS)*. *MySQL* juga bisa dikatakan sebagai implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis (Mundzir, 2018).

Instruksi yang sering dipakai pada *MySQL* ialah *Select* (mengambil), *Insert* (menambah), *Update* (mengubah), dan *Delete* (menghapus). Selain itu *SQL* juga menyiapkan instruksi untuk membuat *database*, *field*, dan *index* untuk menambah ataupun menghapus data ( (Saputra, 2012).

kelebihan *MySQL* menurut (Mundzir, 2018):

1. *Portabilitas*: *MySQL* dapat berjalan dengan stabil pada banyak sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac Os X Server*, *Solaris*, *Amiga*, dan banyak lagi.
2. *Open Source*: didistribusikan secara gratis atau *free*.
3. *Multi-User*: bisa digunakan oleh beberapa pengguna dalam waktu bersamaan tanpa mengalami masalah.
4. *Performance Tunning*: memiliki kecepatan akses dan mampu memproses lebih banyak *SQL*.

5. Ragam Tipe Data: *MySQL* memiliki beragam tipe data seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Keamanan: lapisan keamanan *MySQL* dengan menggunakan *password* yang terenkripsi.
7. Lokalisasi: dapat mendeteksi pesan kesalahan klien dengan menggunakan lebih dari dua puluh bahasa.
8. Koneksi: dapat melakukan koneksi, seperti protocol *TCP/IP, Unix socket (UNIX)*, atau *named pipes (NT)*.
9. Struktur Tabel: memiliki struktur tabel yang fleksibel dalam menangani *ALTER TABLE*, dibandingkan *PostgreSQL* dan *Oracle*.
10. Klien dan Peralatan: dilengkapi dengan *tool* untuk administrasi basis data dan disertakan petunjuk *online*.

Adapun 4 perintah dasar atau query commands yang ada pada *MySQL* antara lain (Saputra, 2012):

1. Menambahkan Data (*Insert*)

Instruksi ini dipakai jika ingin menambahkan data. biasanya ditandai dengan adanya query “Insert Into”. Format dasarnya sebagai berikut:

```
INSERT INTO (Field1,Field2,Field3)
```

```
VALUES ('Var1', 'Var2', 'Var3')
```

## 2. Menampilkan Data (*Select*)

Instruksi ini dipakai jika ingin menampilkan data dari suatu tabel. Format dasarnya sebagai berikut:

```
SELECT * FROM NamaTabel
```

## 3. Mengubah Data (*Update*)

Perintah ini digunakan jika ingin mengubah suatu data. Format dasarnya sebagai berikut:

```
UPDATE NamaTabel SET Field1= 'Var1' , Field2= 'Var2' , Field3= 'Var3'  
WHERE FieldX= 'VarX'
```

## 4. Menghapus Data (*Delete*)

Perintah ini digunakan jika ingin menghapus suatu data. Format dasarnya sebagai berikut:

```
DELETE FROM NamaTabel WHERE FieldX= 'VarX'
```

### **2.3.7 StarUML**

UML (*Unified Modeling Language*) merupakan standar bahasa yang banyak dipakai didunia industri untuk menjelaskan sebuah *requirement*, menganalisis dan mendesain, serta menjabarkan desain dalam pemrograman berorientasi objek (Rosa & Shalahuddin, 2013).

Terdapat 13 macam diagram UML 2.4.0 yang terbagi menjadi 3 kategori yaitu (Rosa & Shalahuddin, 2013)

1. *Structure Diagrams*

Sekumpulan skema yang berfungsi untuk menjelaskan struktur statis dari sebuah sistem yang dimodelkan. Diagram *UML* yang terdapat pada jenis ini antara lain *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.

2. *Behaviour Diagrams*

Sekumpulan skema berfungsi untuk menjelaskan perilaku sebuah sistem atau deretan perubahan yang terjadi pada sistem tersebut. Diagram *UML* yang terdapat dalam jenis ini seperti *use case diagram*, *activity diagram*, dan *state machine diagram*.

3. *Interaction Diagrams*

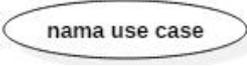
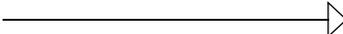
Sekumpulan skema yang berfungsi menjelaskan hubungan suatu sistem dengan sistem lain ataupun hubungan antar subsistem pada sebuah sistem. Menurut (Rosa & Shalahuddin, 2013) *use case* dan *sequence diagram* adalah anggota dari desain sistem, Untuk penelitian ini skema diagram yang akan dipakai antara lain:

1. *Use Case*

*Use case diagram* adalah jenis pemodelan yang menjelaskan perilaku (*behaviour*) suatu sistem informasi yang akan dirancang. *Use case* menjelaskan

sebuah hubungan antara satu atau lebih aktor dengan sistem informasi yang akan dirancang.

**Tabel 2.3** Atribut *Use Case Diagram*

Atribut	Penjelasan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai bagian yang saling berkirim pesan antar bagian atau aktor; biasanya dipakai menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor</p> 	<p>Orang, proses, yang berhubung dengan sistem informasi yang akan diciptakan diluar dari sistem informasi yang akan diciptakan sendiri, biasanya dipakai menggunakan kata benda di awal frase nama aktor</p>
<p>Asosiasi</p> 	<p>Hubungan antar aktor dan <i>use case</i> yang ikut serta dan memiliki keterikatan dengan aktor</p>
<p>Ekstensi &lt;&lt;extend&gt;&gt;</p> 	<p>Hubungan <i>use case</i> ke suatu <i>use case</i> dimana <i>use case</i> yang diciptakan mampu berdiri sendiri meskipun tanpa <i>use case</i> baru. Arah panah menuju ke <i>use case</i> yang ditambahkan</p>
<p>Generalisasi</p> 	<p>Interaksi generalisasi dan spesialisasi antar dua <i>use case</i> yang satu berperan secara <i>universal</i> dari fungsi yang lainnya . arah panah cenderung menuju pada <i>use case</i> yang tergeneralisasi</p>
<p><i>Include / uses</i> &lt;&lt;incude&gt;&gt;</p>  <<uses>> 	<p>Hubungan <i>use case</i> yang dibuat ke suatu <i>use case</i> dimana <i>use case</i> baru yang diciptakan membutuhkan <i>use case</i> ini untuk memproses kegunaannya. Arah panah cenderung menuju ke <i>use case</i> yang dibuat baru</p>

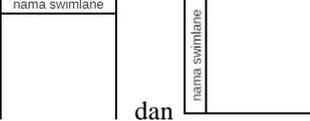
**Sumber:** (Rosa & Shalahuddin, 2013)

## 2. Activity Diagram

*Activity diagram* adalah skema yang mendeskripsikan *workflow* (aliran kerja) atau kegiatan dari sistem atau menu yang terdapat pada perangkat lunak. Dapat disimpulkan bahwa *activity diagram* menjelaskan kegiatan yang dilakukan

sistem, dan tidak terpengaruh dari apapun yang actor lakukan. Atribut-atribut yang dipakai pada *activity diagram* data dilihat dpada tabel dibawah ini (Rosa & Shalahuddin, 2013):

**Tabel 2.4** Atribut *Activity Diagram*

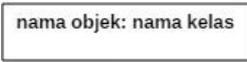
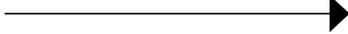
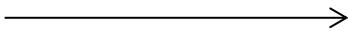
Atribut	Penjelasan
Kondisi awal 	Kondisi awal pada kegiatan sistem, sebuah skema kegiatan yang mempunyai kondisi awal
Aktivitas 	Kegiatan yang dibuat sistem serta dimulai dengan kata kerja
Percabangan 	Hubungan percabangan yang apabila harus memilih lebih dari satu kegiatan
Penggabungan 	Hubungan penggabungan yang apabila lebih dari satu kegiatan dijadikan satu
Kondisi akhir 	Kondisi akhir yang dibuat sistem, suatu diagram aktivitas yang mempunyai kondisi akhir
<i>Swimlane</i> 	Menguraikan sistem bisnis yang berkewajiban terhadap keegiatan yang sedang terjadi

**Sumber:** (Rosa & Shalahuddin, 2013)

### 3. *Sequence Diagram*

*Sequence diagram* mendeskripsikan perilaku suatu objek pada *use case* dan menggambarkan alur hidup (*life cycle*) objek dan *message* (pesan) yang akan dikirim dan diterima antar objek. Banyak *sequence diagram* dibuat minimum sebanyak penjabaran pada *use case* yang mempunyai alur kerja sendiri. Lebih banyak *use case* yang gambar lebih banyak pula *sequence diagram* yang harus dirancang. Adapun atribut-atribut yang dipakai pada *sequence diagram* dilihat pada tabel berikut (Rosa & Shalahuddin, 2013):

**Tabel 2.5** Atribut *Sequence Diagram*

Atribut	Penjelasan
<p>Aktor</p> 	<p>Orang, proses, yang berhubung dengan sistem informasi yang akan diciptakan diluar dari sistem infomasi yang akan diciptakan sendiri, biasanya dipakai menggunakan kata benda di awal frase nama aktor</p>
<p>Garis hidup</p> 	<p>Hubungan antara aktor dengan <i>use case</i> yang berkontribusi, atau <i>use case</i> mempunyai interaksi dengan aktor</p>
<p>Objeka</p> 	<p>Menjelaskan objek yang berhubungan dengan pesan</p>
<p>Waktu aktif</p> 	<p>Menjelaskan objek pada kondisi aktif dan berkomunikasi antar objek. Seluruh yang berinteraksi dengan kondisi aktif adalah suatu tahapan yang dibuat didalamnya.</p>
<p>Pesan jenis <i>create</i>            &lt;&lt;create&gt;&gt;  </p>	<p>Menjelaskan sebuah objek menciptakan objek lain. Arah panah cenderung ke objek yang diciptakan</p>
<p>Pesan jenis <i>call</i>            1.:Nama_model()  </p>	<p>Menjelaskan suatu objek memanggil unit yang terdapat pada objek lain. Arah panah cenderung mengarah ke objek yang memiliki unit/model</p>
<p>Pesan jenis kembali (<i>return</i>)            1.: Keluaran  </p>	<p>Menjelaskan sebuah objek yang sudah melakukan sebuah proses atau cara mendapatkan sebuah kembalian ke objek. Arah panah cenderung ke objek yang menerima</p>
<p>Pesan jenis mengirim            1: Masukan  </p>	<p>Menjelaskan bahwa sebuah objek yang mengirimkan data atau pesan kepada objek lain.</p>
<p>Pesan jenis <i>destroy</i>            &lt;&lt;destroy&gt;&gt;            2:  </p>	<p>Menjelaskan sebuah objek menghentikan hidup objek lain, arah panah cenderung mengarah ke objek yang diakhiri</p>

**Sumber:** Data Penelitian, 2019

### 2.3.8 *Java Script* dan *JQuery*

*JavaScript* adalah alih bentuk dari bahasa *C++* dengan model penulisannya lebih minimalis. Sejumlah faktor penting pada *JavaScript* adalah:

1. Memakai blok awal “{“ dan blok akhir “}”
2. *Automatic conversion* untuk operasi tipe data yang tidak sama
3. *Sensitive case*, yaitu menyeleksi antara huruf kecil dan huruf kapital hingga harus berhati-hati untuk memakai nama variabel, fungsi, dan lain-lain.
4. *Extension* biasanya menggunakan “\*.js”
5. Setiap *statement* ditutup tanda baca *semi colon* (;) dapat juga tidak
6. Jika tidak didukung oleh *browser* versi lama, *script* dapat disembunyikan diantara *tag* “<!--“ dan “-->”

*jQuery* merupakan *framework* berbasis *JavaScript*, yang didalamnya terdapat sejumlah kode atau fungsi *JavaScript* yang akan dipakai untuk memudahkan untuk membuat kode *JavaScript*. Sejumlah efisien yang dipunyai *jQuery* seperti:

1. memanipulasi elemen *HTML*
2. memanipulasi *CSS*
3. penanganan *event* pada *HTML*
4. efek-efek *JavaScript* dan animasi

## 2.4 Penelitian Terdahulu

Penelitian terdahulu adalah penelitian yang dilakukan oleh para ahli dan peneliti yang terdahulu sebelum penelitian ini. Hasil dari penelitian dijadikan sebagai pijakan pustaka acuan untuk penelitian ini, Mulai dari variabel yang terkait hingga hipotesis yang signifikan dari penelitian yang tercatat. Deskripsi lebih lanjut tentang penelitian terdahulu dilakukan dan berhubungan dengan penelitian ini adalah sebagai berikut:

1. Judul “*KOMUNIKASI PSIKIATER DAN PASIEN PENDERITA BIPOLAR*”.

Nama penulis: Merizha Yamudaha, Femi Oktaviani, Reza Rizkina Taufik.

Jurnal Ilmu Komunikasi (J-IKA) Vol.2 2015. ISSN: 2355-0287. Kesimpulan:

Bahwa komunikasi antarpribadi yang dilakukan oleh psikiater dan pasien penderita bipolar sangat membantu memudahkan psikiater untuk berinteraksi dengan pasien pada proses pemulihan, dimana psikiater dan keluarga adalah tokoh sentral yang ikut andil dalam proses pemulihan.

2. Judul “*DIAGNOSIS AND TREATMENT OF PATIENTS WITH BIPOLAR DISORDER: A REVIEW FOR ADVANCED PRACTICE NURSE*”. Penulis:

Bethany Murray dan Brittany McNew. Jurnal *The American Association of Nurse Practitioners*. ISSN: 23276924 DOI: 10.1002/2327-6924.12275.

Kesimpulan: Bipolar Disorder memiliki prevalensi seumur hidup substansial dalam populasi sebesar 4%. Karena gejala mania atau depresi bipolar cenderung menjadi berat dan berulang selama masa hidup pasien, kondisi ini dikaitkan dengan beban yang signifikan terhadap individu, pengasuh, dan

masyarakat. Kesadaran dokter bahwa bipolar dapat muncul meningkatkan kemungkinan sukses dalam pemulihan serta pengobatan yang tepat. Sejumlah perawatan farmakologis dan nonfarmakologis tersedia untuk perawatan akut dan pemeliharaan, dengan prospek mencapai pengurangan beban gejala dan peningkatan fungsi bagi banyak pasien.

3. Judul “*SISTEM PAKAR KEJIWAAN DENGAN FORWARD CHAINING BERBASIS WEB*”. Penulis: Jesreel Surbakti, Aqwan Rosadi Kardian. Jurnal Ilmiah Komputasi Vol.15 2016. ISSN: 1412-9434.

Kesimpulan: Dalam proses perancangan dan pembuatan sistem pakar kejiwaan dengan forward chaining berbasis web ini, ada beberapa kesimpulan yang dapat disampaikan sebagai hasil dari evaluasi pengembangan sistem sebagai berikut:

1. Pembuatan algoritma *forward chaining* sudah sesuai dengan rancangan.
2. Penerapan siklus *forward chaining* ke dalam *website* sudah sesuai dengan rancangan.
3. Pengimplementasian sistem pakar pada sebuah *website* dengan membuat menu diagnosis yang memungkinkan pengguna untuk memilih gejala. Gejala yang dipilih oleh pengguna akan diproses dengan *forward chaining* sehingga sistem pakar akan memberikan *output* penyakit yang diderita oleh pengguna.
4. *Output* sistem pakar terdiri dari biodata pengguna, nama penyakit, isi penyakit, pencegahan, pengobatan, persentase kemungkinan terkena penyakit berdasarkan dari input gejala oleh pengguna, dan tanggal diagnosa.

5. Menu lokasi bekerja dengan baik. Data lokasi di input oleh admin dengan memilih menu lokasi pada menu pakar. Cara memasukkan data lokasi dapat dilihat di menu bantuan.
6. Menu artikel bekerja dengan baik. Data artikel di input oleh admin dengan cara memilih menu artikel pada menu pakar.
7. Dengan menggunakan sistem ini diharapkan pengetahuan masyarakat tentang penyakit kejiwaan dapat bertambah sehingga masyarakat dapat bekerjasama untuk menghilangkan diskriminasi pada penderita penyakit jiwa karena lingkungan sangat berpengaruh terhadap kesembuhan.
4. Judul “*SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN PADI BERBASIS WEB DENGAN FORWARD DAN BACKWARD CHAINING*”. Penulis: Anton Setiawan Honggowibowo. Jurnal Telkomnika Vol.5 2009. ISSN: 1693-6930. Kesimpulan: Hasil implementasi sistem pakar diagnosa penyakit tanaman padi dengan metode inferensi *forward chaining* dan *backward chaining* berbasis *web* mempermudah untuk diakses oleh siapa saja (khususnya petani) dan dimana saja (asalkan tersedia jaringan internet). Sistem ini dibangun untuk menyimpan pengetahuan keahlian seorang pakar pertanian khususnya tanaman padi, sehingga sistem dapat dijadikan asisten pandai di bidangnya sebagai sumber pengetahuan oleh *user*. Pembangunan sistem dirancang sedemikian rupa sehingga dapat mengadopsi perkembangan penyakit penalaran yang digunakan berbasis aturan (*Rule-Based Reasoning*) dengan metode inferensi *forward chaining* dan *backward chaining*. Implementasi sistem pakar dalam

bentuk *web* sangat membantu memberikan kemudahan bagi user dalam mengaksesnya.

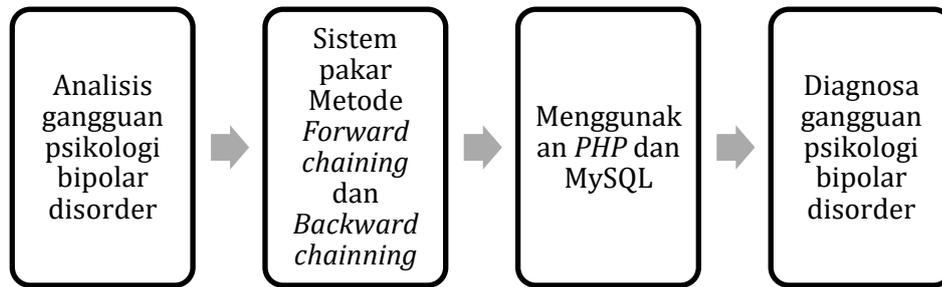
5. Judul“*APLIKASI SISTEM PAKAR UNTUK MENGIDENTIFIKASI PENYAKIT PADA HEWAN HAMSTER (CRICETINAE) DAN SARAN PENGOBATANNYA MENGGUNAKAN METODE BACKWARD CHANING*

Penulis: Susi Kurniasih, Dwi Aryanto, Agung Purwo Wicaksono. Jurnal Juita Vol 2 2012. *ISSN: 2086-9398*. Kesimpulan:

1. Telah dibangun suatu program aplikasi sistem pakar hamster yang dapat membantu masyarakat, terutama pecinta hamster, dalam memahami penyakit hamster dan memberikan saran pengobatan.
2. Dalam aplikasi sistem pakar ini, data yang terdapat pada aplikasi dapat diubah atau ditambah jika ditemukan data baru.

## **2.5 Kerangka Pemikiran**

Kerangka pemikiran merupakan penjelasan secara teoritis hubungan variabel yang diteliti. Kerangka pemikiran juga penghubung antara bermacam konsep yang dasari teori. Hubungan antara variabel terikat dan bebas harus dijelaskan secara teoritis. Syarat utama agar kerangka pemikiran dapat dipercayai adalah alur-alur pemikiran harus logis untuk membangun sebuah pemikiran yang logis dan dapat membuahkan kesimpulan. Untuk mempermudah dalam penelitian ini, Maka disusun bagan alur kerangka pemikiran sebagai berikut:



**Gambar 2.3** Kerangka Pemikiran  
**Sumber:** Data Penelitian, 2019

Keterangan gambar kerangka pemikiran:

1. Analisis data jenis bipolar disorder bersumber data dari buku dan jurnal ilmiah (penelitian terdahulu) serta wawancara langsung dengan pakar yang berkaitan (dokter spesialis kejiwaan).
2. Pembuatan sistem pakar ini menggunakan metode *forward chaining* dan *backward chaining* proses diagnosa dapat berupa konsultasi yang dimulai dari menanyakan gejala-gejala pada penderita bipolar, dan yang kedua proses diagnosa dapat dengan memilih daftar tingkatan bipolar sehingga akan memunculkan informasi tentang penyebab dan langkah-langkah penanganan gangguan psikologi tersebut.
3. Dalam perancangan sistem pakar ini menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai manajemen *database*.
4. Keluaran (*output*) dari proses perancangan sistem pakar dalam penelitian ini berupa aplikasi berbasis *web* diagnosis gangguan psikologi bipolar disorder menggunakan metode *forward chaining* dan *backward chaining*.