

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Teori merupakan penalaran atau alur logika dari seperangkat konsep, definisi, dan proposisi yang disusun secara sistematis. Teori memiliki tiga fungsi yaitu menjelaskan, meramalkan, dan mengendalikan suatu gejala (Sudaryono, 2015: 14).

Bab ini menjelaskan beberapa teori dasar yaitu kecerdasan buatan atau *Artificial Intelligence (AI)* dan subdisiplin ilmunya seperti Logika *Fuzzy (Fuzzy Logic)*, Jaringan Saraf Tiruan (*Artificial Neural Network*), Sistem Pakar (*Expert System*); *Android*, Basis Data dan Validitas Sistem.

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence (AI)*

Menurut Budiharto dan Suhartono (2014: 2) kecerdasan buatan merupakan bidang ilmu komputer yang berkembang beberapa tahun terakhir dengan pertumbuhan kebutuhan akan perangkat cerdas untuk memecahkan berbagai masalah, membangun sebuah sistem, untuk industri maupun kebutuhan rumah tangga. Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia seperti penalaran, pembelajaran, pemecahan masalah, dan sebagainya.

Intelligence berasal dari kata *Intelligo* yang berarti saya paham atau cerdas. Jadi *intelligence* merupakan kemampuan memahami dan melakukan aksi. Sedangkan *artificial* artinya adalah buatan. Kecerdasan buatan bermula dari kemunculan komputer sekitar tahun 1940-an. Pada zaman saat ini, masyarakat terfokus terhadap kemampuan komputer mengerjakan suatu hal yang dapat dilakukan manusia yang berarti komputer dapat meniru kemampuan kecerdasan dan perilaku dari manusia. Sebuah mesin dikatakan cerdas maka harus diberi bekal pengetahuan, pengalaman dan diberi kemampuan untuk menalar dalam melakukan sebuah tindakan. Kecerdasan buatan mencakup bidang yang besar, mulai dari yang umum hingga khusus. Kecerdasan buatan relevan dengan berbagai macam *task* kecerdasan, merupakan sebuah ilmu yang universal (Budiharto dan Suhartono, 2014: 3).

Kecerdasan buatan diawali ketika tahun 1950, Alan Turing mencoba menjawab pertanyaan dalam pikirannya “dapatkah komputer berpikir”. Alan Turing melakukan percobaan sederhana yang disebut dengan *Turing Test*. Kesimpulan pada *Turing Test* adalah jika sebuah mesin mampu berperilaku dan mengerjakan seperti yang dilakukan oleh manusia, maka mesin itu dianggap cerdas (*intelligence test*) (Budiharto dan Suhartono, 2014: 4).

Tahun 1956 John McCarthy melakukan penelitian dibantu oleh beberapa ahli lainnya tentang *automata*, jaringan syaraf, dan pembelajaran intelijensia. Hasilnya program mampu berpikir secara non-numerik dan menyelesaikan suatu masalah pemikiran (Suyanto, 2014: 5).

Tahun 1966, kecerdasan buatan yang dikatakan akan menjadi ilmu pengetahuan yang berkembang ternyata tidak sesuai. Perkembangan kecerdasan buatan justru melambat salah satunya dikarenakan program di dalam kecerdasan buatan hanya berisi sedikit pengetahuan atau informasi pada subjeknya. Namun tahun 1988, program mulai dioperasikan di *Digital Equipment Corporation* (DEC) yang membuat kecerdasan buatan kembali sering digunakan. Salah satu contohnya di McDermott, program yang dibuat berhasil menghemat pengeluaran perusahaan sekitar US\$40 juta per tahun. Hampir semua perusahaan khususnya di USA mulai mempelajari dan menggunakan sistem pakar agar bisa memberikan keuntungan lebih bagi perusahaan. Kecerdasan buatan pada zaman sekarang ini sangat marak digunakan, salah satunya sistem *Global Positioning Systems* (GPS). Ketika kebingungan terhadap arah atau tujuan cukup memasukkan tempat tujuan ke dalam sistem pakar dan akan terlihat rute jalan untuk sampai ke tempat tersebut. (Suyanto, 2014: 5-8).

Kombinasi antara kecerdasan buatan dengan bidang ilmu yang lainnya melahirkan subdisiplin ilmu dalam kecerdasan buatan. Beberapa diantaranya adalah *Natural Language Processing* (NLP), *computer vision*, robotika dan sistem navigasi, *game playing*, dan *expert system* (Budiharto dan Suhartono, 2014: 7-14).

Menurut Sutojo, dkk. (2011: 3) terdapat beberapa tujuan dari kecerdasan buatan, yaitu:

1. Membuat mesin menjadi lebih pintar.
2. Memahami apa itu kecerdasan.
3. Membuat mesin lebih bermanfaat.

2.1.1.1 Logika *Fuzzy* (*Fuzzy Logic*)

Menurut Sutojo, dkk. (2011: 211-212) logika *fuzzy* merupakan sistem pemecahan masalah yang bagus diimplementasikan pada sistem, seperti sistem kontrol, jaringan PC, *multi-channel*, dan jaringan komputer. Logika *fuzzy* memiliki nilai keanggotaan dalam rentang 0 (nol) hingga 1 (satu), yang di dalam logika klasik berarti memiliki dua kemungkinan ya atau tidak. Akan tetapi, dalam logika *fuzzy* bisa saja suatu keadaan mempunyai dua nilai ya dan tidak secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya.

Beberapa keuntungan yang ada ketika menggunakan logika *fuzzy* untuk memecahkan suatu masalah, yaitu (Budiharto dan Suhartono, 2014: 152):

1. Mudah dimengerti.
2. Sangat fleksibel.
3. Memiliki toleransi terhadap data-data yang tidak tepat.
4. Mampu memodelkan fungsi-fungsi non linear yang sangat kompleks.
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Dapat bekerja sama dengan teknik-teknik kendali secara konvensional.
7. Didasarkan pada bahasa alami.

Sistem inferensi *fuzzy* adalah bagaimana cara memetakan ruang *input* menuju ruang *output* menggunakan logika *fuzzy*. Empat elemen dasar sistem inferensi *fuzzy* antara lain (Sutojo, dkk., 2011: 232):

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

Beberapa metode yang digunakan dalam sistem inferensi *fuzzy* adalah (Sutojo, dkk., 2011: 233-237):

1. Metode Tsukamoto

Dalam inferensinya, tahapan yang ada di metode Tsukamoto, yaitu:

- a. Fuzzifikasi.
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*).
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*).
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*).

2. Metode Mamdani

Metode ini sering digunakan karena strukturnya yang sederhana. Pada metode ini, untuk mendapatkan *output* diperlukan beberapa tahapan yaitu:

- a. Fuzzifikasi.
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*).
- c. Aplikasi fungsi implikasi menggunakan fungsi *MIN* (*Minimum*) dan komposisi antar-*rule* menggunakan fungsi *MAX* (*Maximum*) dengan menghasilkan himpunan *fuzzy* baru.
- d. Defuzzifikasi menggunakan metode *Centroid* (Titik Tengah).

3. Metode Sugeno

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

- a. Fuzzifikasi.
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*).
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*).
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*).

2.1.1.2 Jaringan Saraf Tiruan (*Artificial Neural Network*)

Menurut Sutojo, dkk. (2011: 283-288) Jaringan saraf tiruan merupakan paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia.

Elemennya adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah elemen yang saling berhubungan (*neuron*), bekerja secara bersama untuk menyelesaikan suatu masalah. Cara kerja jaringan saraf tiruan sama sama halnya dengan cara kerja manusia, yaitu belajar melalui contoh (Sutojo, dkk., 2011: 283-288).

Kelebihan yang ada pada jaringan saraf tiruan adalah (Sutojo, dkk., 2011: 284):

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

Jaringan saraf tiruan juga mempunyai kelemahan-kelemahan, yaitu (Sutojo, dkk., 2011: 284-285):

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatika, operasi logika, dan simbolis.
3. Membutuhkan pelatihan agar dapat beroperasi sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan juga lama.

Terdapat beberapa bagian lapisan penyusun jaringan saraf tiruan, yaitu (Sutojo, dkk., 2011: 292):

1. Lapisan *Input* (*Input Layer*)

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

2. Lapisan Tersembunyi (*Hidden Layer*)

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

3. Lapisan *Output* (*Output Layer*)

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

Menurut Sutojo, dkk. (2011: 292-295) terdapat arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan antara lain:

1. Jaringan Lapisan Tunggal

Jaringan yang terdiri dari 1 lapisan *input* dan 1 lapisan *output*, yang di setiap unit dalam lapisan *input* terhubung dengan setiap unit yang terdapat pada lapisan *output*. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *ADALINE*, *Hopfield*, dan *Perceptron*.

2. Jaringan Lapisan Banyak

Jaringan yang mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Keuntungan pada jaringan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan

jaringan lapisan tunggal. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *MADALINE*, *backpropagation*, dan *Neocognitron*.

3. Jaringan dengan Lapisan Kompetitif

Jaringan yang memiliki bobot yang telah ditentukan dan tidak memiliki proses pelatihan. Jaringan ini sering digunakan untuk mengetahui *neuron* pemenang dari sejumlah *neuron* yang ada sehingga sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *Learning Vector Quantization* (LVQ).

2.1.1.3 Sistem Pakar (*Expert System*)

Menurut Hartati dan Iswanti (2008: 13) sistem pakar adalah sistem yang dibuat untuk menirukan keahlian dari seorang pakar untuk menjawab pertanyaan dan dapat memecahkan suatu masalah. Nantinya dengan sistem pakar seseorang yang bukan pakar dapat menjawab pertanyaan, memecahkan masalah dan mengambil keputusan layaknya seorang pakar.

Sistem pakar awalnya dimulai pada pertengahan 1960, ditandai dengan lahirnya sistem pakar pertama bernama *General-purpose Problem Solver (GPS)* yang dikembangkan oleh Newel dan Simon. Lalu bermunculan sistem pakar lain di berbagai bidang seperti MYCIN untuk diagnosis penyakit, DENDRAL untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON & XSEL untuk membantu konfigurasi sistem komputer besar, SOPHIE untuk analisis sirkuit elektronik, Prospector digunakan di bidang geologi untuk membantu

mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberikan keputusan bagi seorang manajer dalam masalah stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya (Sutojo, dkk., 2011: 159-160).

Menurut Hartati dan Iswanti (2008: 11) Pakar merupakan seorang yang memiliki pengetahuan khusus di bidangnya, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan masalah dalam bidang tertentu. Seorang pakar memiliki kemampuan kepakaran seperti :

1. Mengenali dan merumuskan suatu masalah.
2. Menyelesaikan masalah dengan cepat dan tepat.
3. Menjelaskan solusi dari suatu masalah.
4. Restrukturisasi pengetahuan.
5. Belajar dari pengalaman.
6. Memahami batas kemampuan.
7. Didasarkan pada bahasa alami.

Adapun ciri-ciri dari sistem pakar, yaitu (Sutojo, dkk., 2011: 162):

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami
4. Bekerja berdasarkan kaidah tertentu
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme inferensi diletakkan terpisah

7. Keluarannya (*output*) bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara terpisah secara searah, sesuai dengan dialog dengan pengguna

Adapun manfaat pada sistem pakar, yaitu (Sutojo, dkk., 2011: 160-161):

1. Meningkatkan produktifitas.
2. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar
3. Meningkatkan kualitas.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Handal (*realibility*).
8. Meningkatkan kapabilitas sistem komputer.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam latihan.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

Adapun kekurangan pada sistem pakar, yaitu (Sutojo, dkk., 2011: 161):

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

Menurut Hartati dan Iswanti (2008: 22) representasi pengetahuan berarti mengorganisasikan pengetahuan dalam bentuk dan format tertentu agar dimengerti oleh komputer.

Beberapa struktur sistem pakar yang berbasis kaidah produksi, yaitu (Hartati dan Iswanti, 2008: 10):

1. Antar Muka Pemakai

Sistem pakar menyediakan komunikasi antara sistem dan pemakai, yang disebut antar muka. Komponen antar muka harus efektif dan mudah digunakan agar bagi pengguna yang tidak ahli dalam bidang yang diterapkan pada sistem pakar bisa menggunakannya (Hartati dan Iswanti, 2008: 4-5).

2. Basis Pengetahuan

Basis pengetahuan merupakan sekumpulan pengetahuan tertentu pada tingkatan pakar dalam format tertentu. Pengetahuan didapat dari sekumpulan pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti buku, jurnal ilmiah, majalah, dan dokumentasi tercetak lainnya (Hartati dan Iswanti, 2008: 5).

3. Struktur Kontrol (Mesin Inferensi)

Struktur kontrol merupakan otak dari sistem pakar, yaitu berupa perangkat lunak yang berfungsi sebagai tugas inferensi penalaran sistem pakar. Intinya struktur kontrol mencari suatu permasalahan. (Hartati dan Iswanti, 2008: 5).

Beberapa konsep perunutan yang ada pada struktur kontrol, yaitu (Hartati dan Iswanti, 2008: 45-47):

a. Runut Maju (*Forward Chaining*)

Runut maju adalah proses perunutan (penalaran) dimulai dari data dan fakta yang ada (*IF*) menuju konklusi akhir atau *derived information* (*THEN*). Bentuk konsepnya sebagai berikut:

IF (informasi masukan)

THEN (konklusi)

Informasi masukan berupa data, bukti, temuan atau pengamatan sedangkan berupa tujuan, penjelasan, hipotesa atau diagnosis. Dalam metode ini, sistem akan menerima semua gejala yang diisikan pemakai lalu sistem akan memeriksa gejala-gejala dan selanjutnya mencocokkan dengan konklusi yang sesuai.

b. Runut Mundur (*Backward Chaining*)

Runut mundur adalah proses perunutan (penalaran) dimulai dari tujuan menuju ke jalur yang akan mengarahkan ke tujuan tersebut hingga bagian kondisi terpenuhi. Bentuk konsepnya sebagai berikut:

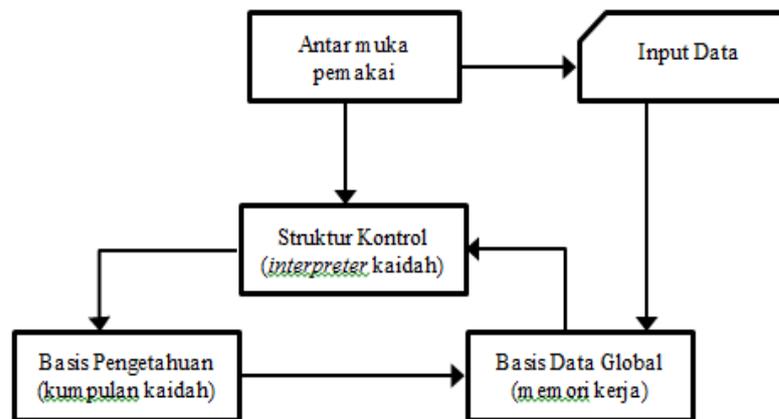
Tujuan,

IF (kondisi)

Pada perunutan ini dimulai dari tujuan lalu merunut balik ke jalur yang mengarah ke tujuan, untuk membuktikan bahwa kondisi pada aturan terpenuhi.

4. *Working Memory* (Memori Kerja) atau Basis Data Global

Memori kerja merupakan bagian yang menyimpan fakta-fakta saat dilakukan proses konsultasi. Fakta-fakta diolah oleh struktur kontrol untuk menentukan keputusan pemecahan masalah, biasanya berupa hasil diagnosa, akibat maupun tindakan (Hartati dan Iswanti, 2008: 6).



Gambar 2.1 Struktur Sistem Pakar Berbasis Kaidah Produksi
(Sumber: Firebaugh, 1988 dalam Hartati dan Iswanti, 2008: 10)

Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah *IF-THEN* menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Contoh struktur kaidah *IF-THEN* yang menghubungkan obyek (Adedeji, 1992 dalam Hartati dan Iswanti, 2008: 25):

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

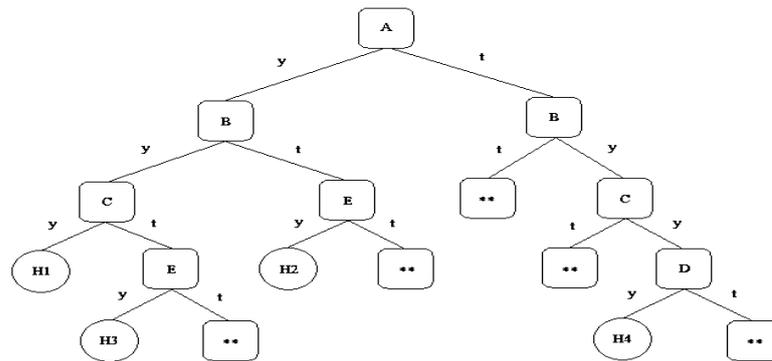
Menurut Hartati dan Iswanti (2008: 25-26) premis mengacu pada fakta yang benar sebelum konklusi diperoleh. Masukan mengacu pada data yang tersedia sebelum keluaran diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data berisikan informasi yang harus tersedia sehingga hasil dapat diperoleh. Tindakan berisi kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan yang mendorong adanya pemeriksaan.

Contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan, yaitu (Hartati dan Iswanti, 2008: 26-39) :

Tabel 2.1 Tabel Keputusan

Hipotesa <i>Evidence</i>	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	Ya	ya	tidak

Sumber: Hartati dan Iswanti (2008: 32)



Keterangan:

A = <i>evidence</i> A,	H1 = hipotesa 1,	y = ya
B = <i>evidence</i> B,	H2 = hipotesa 2,	t = tidak
C = <i>evidence</i> C,	H3 = hipotesa 3,	** = tidak menghasilkan hipotesa
D = <i>evidence</i> D,	H4 = hipotesa 4	

Gambar 2.2 Pohon Keputusan
(Sumber: Hartati dan Iswanti, 2008: 33)

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

Dalam sesi konsultasi pada sistem pakar, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna

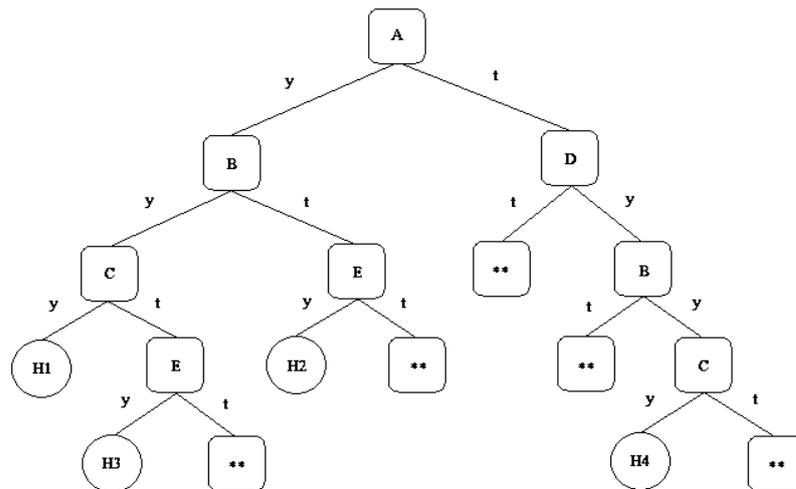
tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

Tabel 2.2 Alternatif Tabel Keputusan

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

Sumber: Hartati dan Iswanti (2008: 34)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



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 2.3 Alternatif Pohon Keputusan
(Sumber: Hartati dan Iswanti, 2008: 35)

Dilihat dari gambar 2.3, masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi “y” dan “t” sudah tidak mengarah pada *evidence* yang sama. Hal ini berarti jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula.

Kaidah yang dapat dihasilkan berdasarkan pohon keputusan pada gambar 2.3 adalah sebagai berikut:

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

Kaidah produksi banyak digunakan pada aplikasi sistem pakar karena lebih mudah dipahami dan bersifat deklaratif atau sesuai dengan jalan pikiran manusia dalam menyelesaikan suatu masalah dan mudah diinterpretasikan.

2.1.2 Android

Menurut Tim EMS (2015: 1-9) *Android* merupakan *software* yang digunakan pada perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh *Google*. Pengembangan aplikasi pada *platform android* menggunakan dasar bahasa pemrograman *Java*. *Platform android* bersifat *open-source*, yaitu bisa dikembangkan untuk membangun aplikasi yang kaya dan inovatif. Fitur yang ada pada *android*, yaitu:

1. *Android run-time*, yaitu terdiri atas *library java* dan *dalvik virtual machine*.
2. *Open graphics library*, yaitu *application program interface* yang digunakan untuk membuat grafis 2D dan 3D.
3. *Webkit*, yaitu *engine* dari *web browser* yang digunakan untuk menampilkan isi *website* dan menyederhanakan tampilan dari proses *loading*.
4. *SQLite*, yaitu *engine* dari relasional *database* yang dapat diintegrasikan dengan aplikasi.

Kelebihan dari penggunaan sistem *android*, yaitu:

1. *Multitasking*, dapat menjalankan aplikasi secara bersamaan.
2. Terdapat notifikasi ketika ada panggilan atau sms, yaitu ketika ada sms atau *email* yang masuk, akan terdapat notifikasi pada ponsel.
3. Dukungan ribuan aplikasi terpercaya melalui situs *Google Play*, yaitu berfungsi mendapatkan berbagai aplikasi yang diperlukan.
4. Penggunaan *widget* pada *home screen*, yaitu memudahkan dan mempercepat pengguna ketika membuka aplikasi.

Kelemahan dari penggunaan sistem *android*, yaitu:

1. Boros pada penggunaan baterai.
2. Sentralisasi *Google*.
3. Tidak mendukung penggunaan J2ME.
4. *Security* yang masih tergolong rendah.

2.1.3 Basis Data (*Database*)

Menurut A.S. dan Shalahuddin (2011: 43-44) sistem basis data merupakan sistem yang terkomputerisasi bertujuan untuk memelihara data yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. Basis data merupakan media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Contoh basis data dalam sebuah sistem yaitu *Database Management System (DBMS)*. *DBMS* adalah sistem aplikasi untuk menyimpan, mengelola, dan menampilkan data. Persyaratan yang harus dipenuhi agar bisa disebut dengan *DBMS*, yaitu (A.S. dan Shalahuddin, 2011: 45):

1. Menyediakan fasilitas mengelola akses data
2. Menangani integritas data
3. Menangani akses data yang dilakukan secara bersamaan
4. Menangani *backup* data

DBMS yang paling banyak digunakan saat ini, yaitu:

1. *DBMS* versi komersial, yaitu *Oracle*, *Microsoft SQL Server*, *IBM DB2*, dan *Microsoft Access*.
2. *DBMS* versi *open source*, yaitu *MySQL*, *PostgreSQL*, *Firebird*, dan *SQLite*.

2.1.4 Validasi Sistem

Menurut A.S. dan Shalahuddin (2011: 211-214) validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang

dibangun dapat digunakan dan telah sesuai dengan yang diharapkan. Adapun pendekatan dalam melakukan pengujian untuk validasi sistem yaitu:

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

Black-box testing adalah pengujian sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program yang bertujuan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak telah sesuai dengan spesifikasi yang dibutuhkan. Pengujian pada *black-box testing* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem apakah telah sesuai dengan spesifikasi yang dibutuhkan.

2. *White-Box Testing* (Pengujian Kotak Putih)

White-box testing adalah pengujian sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. Pengujian pada *white-box testing* dilakukan dengan memeriksa logika dari kode program.

2.2 Variabel Penelitian

Variabel penelitian adalah sesuatu yang berbentuk apa saja, bisa berbentuk orang, obyek atau kegiatan yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi, kemudian ditarik kesimpulannya (Sugiyono, 2012: 38). Obyek yang digunakan dalam penelitian ini adalah televisi tabung dan variabel

penelitian yang ditetapkan yaitu kerusakan pada televisi tabung.

2.2.1 Televisi

Televisi merupakan sistem elektronis yang menyampaikan suatu isi pesan dalam bentuk audio visual, dengan adanya televisi masyarakat akan mudah mengetahui apa saja informasi terbaru yang ada di belahan bumi lainnya (Murawan dan Sufa'atin 2015: 1).

Televisi dapat dinikmati oleh semua kalangan baik anak-anak maupun dewasa. Televisi pertama kali ditemukan pada tahun 1927 di San Fransisco. Perkembangan yang sangat pesat membuat televisi semakin populer dan berkembang. Terdapat macam-macam televisi dari televisi *LED*, televisi *LCD*, dan televisi *CRT*. Televisi berbentuk tipis atau *LCD* banyak digunakan oleh masyarakat karena lebih ringan dan lebih elegan. Namun televisi *CRT* (*Cathode-Ray Tubes*) atau biasa disebut televisi tabung juga banyak dipakai di kalangan masyarakat menengah ke bawah karena harga yang masih murah atau terjangkau dibandingkan dengan televisi *LCD*. Kerusakan pada televisi tabung harus diperbaiki melalui operator *service*, berbeda dengan televisi *LED* yang memberikan masa garansi kepada pelanggan. (Wulandari dan Destiani, 2015: 1).

Dari segi fisik, televisi terdiri dari layar dan tabung. Terdapat komponen-komponen di dalam tabung televisi yang saling berhubungan sehingga dapat menghasilkan gambar, suara dan video. Komponen pada televisi tabung terbagi menjadi 4 kelompok, yaitu (Rio dan Sawamura, 2007: 73-139):

1. Rangkaian Catu Daya

Menurut Rio dan Sawamura (2007: 138) rangkaian catu daya terdiri dari rangkaian penyearah untuk sumber dari tegangan bolak-balik dan rangkaian penyearah untuk sumber dari tegangan pulsa yang dibangkitkan oleh rangkaian defleksi horizontal. Rangkaian catu daya berfungsi untuk mengubah sinyal AC (*Alternating Current*) menjadi DC (*Direct Current*) yang selanjutnya didistribusikan ke seluruh rangkaian. Catu daya merupakan bagian yang penting dari semua rangkaian televisi, tanpa catu daya televisi tidak akan aktif. Rangkaian catu daya terdiri dari dua macam yaitu:

1.1. *Power Supply*

Menurut Rio dan Sawamura (2007: 73-74) rangkaian *power supply* atau *regulator* televisi adalah rangkaian yang paling penting dalam televisi. Dari rangkaian inilah segalanya bermula. Rangkaian *power supply* televisi harus memberikan supply tegangan ke seluruh rangkaian lainnya sesuai dengan yang dibutuhkan oleh komponen setiap rangkaian. Jika rangkaian *power supply* televisi tidak mampu memberi kebutuhan sesuai dengan yang diminta oleh rangkaian televisi maka akan terjadi ketidakwajaran di dalam rangkaian dan akan menghasilkan hal yang tidak diinginkan. Dalam teknologi televisi, gambaran lengkap secara teknis system televisi terdapat dalam skema rangkaian. Dengan bantuan skema rangkaian, kerusakan serumit apa pun akan dapat diatasi, terutama untuk mengetahui spesifikasi komponen yang tidak umum atau untuk mengetahui komponen yang telah terbakar.



Gambar 2.4 *Power Supply*
(Sumber: Data Penelitian: 2017)

1.2. *Flyback Transformer (FBT)*

Menurut Rio dan Sawamura (2007: 138) rangkaian *flyback transformer* adalah salah satu komponen monitor untuk menghasilkan tegangan tinggi yang dibutuhkan oleh monitor. *FBT* merupakan trafo untuk tegangan tinggi agar tabung bekerja menghasilkan gambar.



Gambar 2.5 *Flyback Transformer (FBT)*
(Sumber: Data Penelitian: 2017)

2. Rangkaian Penerima Gelombang

Rangkaian penerima gelombang adalah bagian yang berfungsi untuk menerima gelombang, memproses dan menampilkannya ke layar televisi. Pemilihan kanal nantinya terdiri dari selektor mekanik yang berbentuk tombol dan menggunakan selektor elektronika. (Rio dan Sawamura, 2007: 91-92). Rangkaian penerima gelombang terdiri dari dua macam yaitu:

2.1. *Tuner*

Menurut Rio dan Sawamura (2007: 91-92) *tuner* adalah alat untuk menerima sinyal yang disalurkan ke layar televisi. Untuk beberapa daerah, *tuner* terkadang tidak dimanfaatkan karena menggunakan saluran *AV (Audio Video)* dan *receiver digital* parabola. *Tuner* masih banyak dipakai di masyarakat terutama daerah yang mempunyai pemancar televisi lokal.

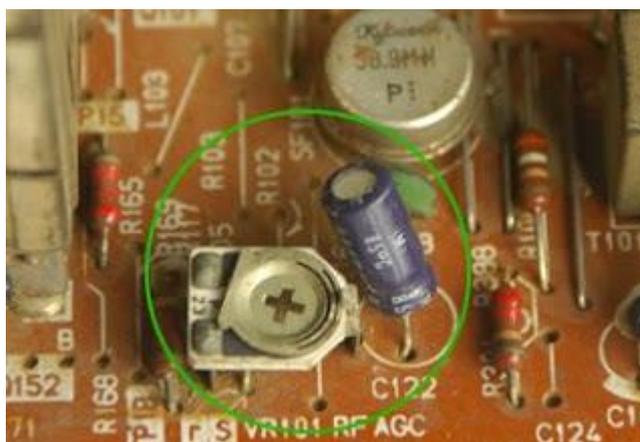


Gambar 2.6 *Tuner*
(Sumber: Data Penelitian: 2017)

2.2. Penguat Video

Menurut Rio dan Sawamura (2007: 66-67) rangkaian ini berfungsi sebagai penguat sinyal luminan yang berasal dari *detector* video sehingga dapat

menjalankan tabung gambar atau *CRT (Cathode Ray Tube)*. Nantinya sinyal sinkronisasi dan sinyal krominan dikeluarkan dan diberikan rangkaian pemroses selanjutnya. Agar menghasilkan gambar berwarna yang baik pada tabung gambar, sinyal luminan dari detektor video diperkuat oleh penguat video kira-kira seratus kali dari rangkaian tunda yang terpasang.



Gambar 2.7 Rangkaian Penguat Video
(Sumber: Data Penelitian: 2017)

3. Rangkaian Gambar

Menurut Rio dan Sawamura (2007: 83) rangkaian gambar adalah bagian yang berfungsi untuk menerima gelombang, memproses dan menampilkannya ke layar televisi. Hampir seluruh rangkaian penerima televisi tergantung pada penguat gambar yang berfungsi mengatur kualitas gambar yang ditampilkan. Rangkaian gambar biasanya berbentuk *IC monolitik*, yaitu dibuat pada keping semikonduktor. Rangkaian gambar terdiri dari dua macam yaitu:

3.1 *IC Program*

Menurut Rio dan Sawamura (2007: 84) rangkaian ini berfungsi untuk mengontrol seluruh sistem kerja yang terjadi pada penerima televisi

berwarna dan rangkaian elektronik lainnya yang menggunakan *IC program* dan memori. Pada pesawat televisi keluaran baru *IC program* sudah dibuat menyatu dengan *IC* utama . Namun untuk televisi tabung masih terpisah, tetapi meskipun terpisah maupun menyatu prinsip kerja *IC program* sama saja.



Gambar 2.8 Rangkaian *IC Program*
(Sumber: Data Penelitian: 2017)

3.2 *IC* Vertikal

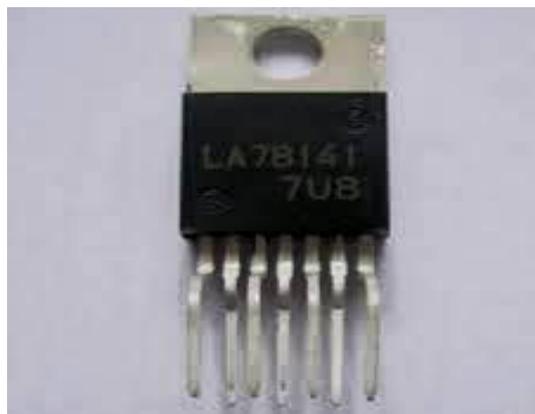
Menurut Rio dan Sawamura (2007: 67) rangkaian ini berfungsi untuk menguatkan sinyal gigi gergaji yang dikeluarkan oleh *IC* utama pin vertikal output ke *IC* penguat vertikal. Setelah dikuatkan, sinyal gergaji ini oleh *IC* vertikal akan diumpankan ke kumparan pembelok vertikal (defleksi yoke) yang akan membelokkan berkas elektron keatas dan kebawah *CRT* (membelokkan secara vertikal).



Gambar 2.9 IC Vertikal
(Sumber: Data Penelitian: 2017)

3.3 IC Horizontal

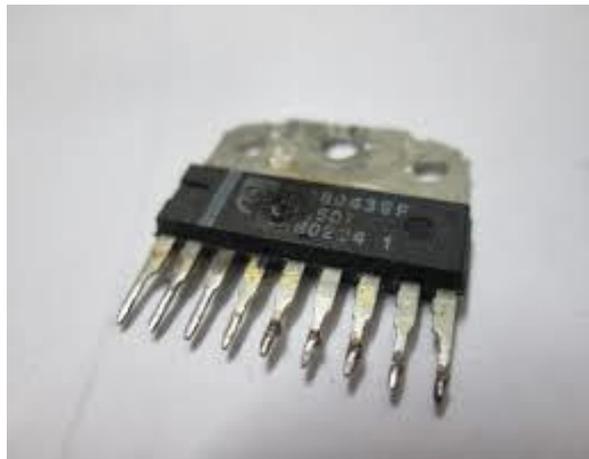
Menurut Rio dan Sawamura (2007: 67-68) rangkaian penguat horizontal bermula dari *H-Out IC* utama, dikuatkan oleh penguat *driver horizontal HDT* (*Horizontal Driver Transformer*) kemudian dikuatkan oleh transistor penguat akhir horizontal dan terakhir sinyal gigi gergaji yang diumpankan ke *FBT* (*Flyback Transformer*) dan defleksi yoke horizontal.



Gambar 2.10 IC Horizontal
(Sumber: Data Penelitian: 2017)

4. Rangkaian Suara

Rangkaian penerima suara adalah bagian yang berfungsi untuk menerima gelombang, memproses dan menghasilkan suara pada televisi. Pemilihan kanal nantinya terdiri dari selektor mekanik yang berbentuk tombol dan menggunakan selektor elektronika. Rangkaian penguat suara terdiri atas sinyal output suara dari IC utama, *volume* kontrol dan *mute* dari IC program, IC penguat suara, catu daya untuk rangkaian penguat suara dan speaker. *Output* adalah keluaran sinyal suara dari IC utama yang akan dikuatkan intensitasnya. *Volume* berfungsi untuk mengontrol besar kecilnya penguatan intensitas suara. *Mute* berfungsi untuk mematikan suara jika diperlukan. IC penguat berfungsi sebagai alat penguat suara. *Speaker* digunakan sebagai pengubah tenaga listrik menjadi tenaga mekanik, sehingga suara televisi dapat didengar (Rio dan Sawamura, 2007: 136-137).



Gambar 2.11 Rangkaian suara
(Sumber: Data Penelitian: 2017)

2.2.2 Kerusakan Televisi

Televisi dapat menjalankan fungsinya karena terdapat banyak komponen yang mendapatkan perawatan dan pemeliharaan yang baik. Tingginya penggunaan televisi dan buruknya perawatan televisi dapat menimbulkan permasalahan yang dapat mengganggu kinerja televisi. Adapun permasalahan yang sering terjadi pada perangkat televisi adalah (Rio dan Sawamura, 2007: 179-245):

1. Televisi mati namun lampu indikator menyala
Perangkat televisi mati namun lampu indikator menyala disebabkan oleh 2 kemungkinan, yaitu antara kerusakan *power supply* dan *flyback transformer*.
2. Kabel *power* ketika dialiri listrik memercikan api
Perangkat televisi Kabel *power* ketika dialiri listrik memercikan api disebabkan oleh 2 kemungkinan, yaitu antara kerusakan *power supply* dan *flyback transformer*.
3. *Miniatur Circuit Breaker (MCB) trip* atau turun
Perangkat televisi *Miniatur Circuit Breaker (MCB) trip* atau turun disebabkan oleh kerusakan *power supply*.
4. Terdengar bunyi meletup (meledak) dari dalam televisi
Perangkat televisi terdengar bunyi meletup (meledak) dari dalam televisi disebabkan oleh kerusakan *Flyback Transformer (FBT)*.
5. Terdengar bunyi berdesis pada televisi
Perangkat televisi terdengar bunyi berdesis pada televisi disebabkan oleh kerusakan *tuner*.

6. Gambar televisi tidak jernih
Perangkat televisi gambar tidak jernih disebabkan oleh kerusakan *tuner*.
7. Gambar televisi *noise* atau bintik-bintik
Perangkat televisi gambar *noise* atau bintik-bintik disebabkan oleh kerusakan *tuner*.
8. Gambar televisi tidak ada namun suara ada
Perangkat televisi gambar tidak ada namun suara ada disebabkan oleh kerusakan Rangkaian penguat video.
9. Layar gambar televisi gelap atau kurang terang
Perangkat televisi gambar gelap atau kurang terang disebabkan oleh kerusakan Rangkaian penguat video.
10. Pencahayaan televisi kurang
Perangkat televisi pencahayaan kurang terang disebabkan oleh kerusakan Rangkaian penguat video.
11. Tombol *setting* pada televisi tidak berfungsi
Perangkat televisi tombol *setting* tidak berfungsi disebabkan oleh kerusakan *IC* Program rusak.
12. Siaran televisi tidak tersimpan
Perangkat televisi siaran tidak tersimpan disebabkan oleh kerusakan *IC* Program rusak.
13. Video *input* tidak berfungsi
Perangkat televisi video *input* tidak berfungsi disebabkan oleh kerusakan *IC* Program rusak.

14. Suara ada namun gambar televisi bergaris

Perangkat televisi suara ada namun gambar televisi bergaris disebabkan oleh 2 kemungkinan, yaitu antara kerusakan *IC* vertikal dan kerusakan *IC* horizontal.

15. Gambar televisi bergerak-gerak dan bergemetar

Perangkat televisi bergerak-gerak dan bergemetar disebabkan disebabkan oleh 2 kemungkinan, yaitu antara kerusakan *IC* vertikal dan kerusakan *IC* horizontal.

16. Gambar televisi menyempit keatas atau ke bawah

Perangkat televisi menyempit keatas atau ke bawah disebabkan oleh 2 kemungkinan, yaitu antara kerusakan *IC* vertikal dan kerusakan *IC* horizontal.

17. Gambar televisi berbentuk oval atau trapesium

Perangkat televisi berbentuk oval atau trapesium disebabkan oleh kerusakan *IC* Horizontal.

18. Gambar televisi ada namun suara tidak ada

Perangkat televisi ada namun suara tidak ada disebabkan oleh kerusakan *IC* *Sound Amplifier*.

19. Suara televisi lemah dan cacat

Perangkat suara televisi lemah dan cacat disebabkan oleh kerusakan *IC* *Sound Amplifier*.

2.3 Software Pendukung

Software pendukung merupakan beberapa perangkat lunak yang digunakan untuk mendukung pembuatan sistem pakar dalam penelitian ini. Perangkat lunak tersebut antara lain: *Android*, *Java*, *Eclipse* dan *StarUML*.

2.3.1 Android



Gambar 2.12 Logo *Android*
(Sumber: Tim EMS, 2015: 1)

Android merupakan *software* yang digunakan pada perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh *Google*. Pengembangan aplikasi pada *platform android* menggunakan dasar bahasa pemrograman *Java*. *Platform android* bersifat *open-source*, yaitu bisa dikembangkan untuk membangun aplikasi yang kaya dan inovatif. Fitur yang ada pada *android*, yaitu (Tim EMS, 2015: 1-9):

1. *Android run-time*, yaitu terdiri atas *library java* dan *dalvik virtual machine*.
2. *Open graphics library*, yaitu *application program interface* yang digunakan untuk membuat grafis 2D dan 3D.
3. *Webkit*, yaitu *engine* dari *web browser* yang digunakan untuk menampilkan isi *website* dan menyederhanakan tampilan dari proses *loading*.

4. *SQLite*, yaitu *engine* dari relasional *database* yang dapat diintegrasikan dengan aplikasi.

Menurut Tim EMS (2015: 1-9) Kelebihan dari penggunaan sistem *android*, yaitu:

1. *Multitasking*, dapat menjalankan aplikasi secara bersamaan.
2. Terdapat notifikasi ketika ada panggilan atau sms, yaitu ketika ada sms atau *email* yang masuk, akan terdapat notifikasi pada ponsel.
3. Dukungan ribuan aplikasi terpercaya melalui situs *Google Play*, yaitu berfungsi mendapatkan berbagai aplikasi yang diperlukan.
4. Penggunaan *widget* pada *home screen*, yaitu memudahkan dan mempercepat pengguna ketika membuka aplikasi.

2.3.2 Java



Gambar 2.13 Logo *Java*
(Sumber: Bambang Hariyanto, 2014: 2)

Java adalah bahasa pemrograman yang dapat dijalankan di komputer maupun telepon genggam. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai *Mesin Virtual Java* (*JVM*). *Java* merupakan bahasa pemrograman yang bersifat umum dan

secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin (Bambang Hariyanto, 2014: 4).

Karena fungsionalitasnya yang memungkinkan aplikasi *java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda. *Java* memiliki beberapa kelebihan yaitu :

1. Bahasa sederhana.
2. Bahasa orientasi objek.
3. Bahasa *statically Typed*.
4. Bahasa dikompilasi.
5. Bahasa yang aman.
6. Bahasa *independen* terhadap *platform*.
7. Bahasa *multithreading*.
8. Bahasa yang didukung *garbage collector*.
9. Bahasa yang tangguh.
10. Bahasa yang mampu diperluas.

2.3.3 Eclipse

Menurut Nasruddin Safaat (2015: 4-7) *Eclipse* merupakan suatu *Integrated Development Environment* dalam mengembangkan perangkat lunak dan bisa dijalankan di semua *platform* atau *multiplatform*. Kelebihan *Eclipse* yaitu:

1. *Multi-platform*.
2. *Multi-language*.
3. *Multi-role*.



Gambar 2.14 Logo *Eclipse*
(Sumber: Nazruddin Safaat, 2015:4)

2.3.4 *StarUML*

Salah satu pemodelan yang saat ini paling banyak digunakan adalah *UML* (*Unified Modeling Language*). *UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S. dan Shalahuddin, 2011: 113).



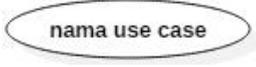
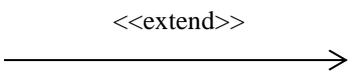
Gambar 2.15 Logo *StarUML*
(Sumber: A.S. dan Shalahuddin, 2011: 1)

Menurut A.S. dan Shalahudin (2011: 130-140) *use case* dan *sequence diagram* merupakan bagian dari desain sistem. Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:

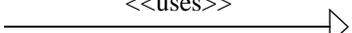
1. Use Case Diagram

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada 2 hal utama yang terdapat pada *use case* yaitu aktor dan *use case*. Berikut ini adalah simbol-simbol yang digunakan dalam *use case diagram* (A.S. dan Shalahuddin, 2011: 130-133).

Tabel 2.3 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>

Tabel 2.3 Lanjutan

Simbol	Deskripsi
<p>generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan/<i>include/uses</i></p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini. Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>

Sumber: A.S. dan Shalahuddin (2011: 131-133)

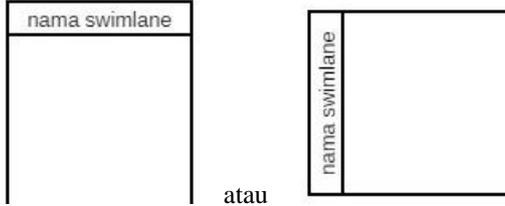
2. Activity Diagram

Activity diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Jadi dapat dikatakan bahwa *activity diagram* menggambarkan aktifitas sistem, bukan apa yang dilakukan oleh aktor. Simbol-simbol yang digunakan dalam *activity diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin: 2011: 134-135).

Tabel 2.4 Simbol Activity Diagram

Simbol	Deskripsi
<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal</p>
<p>Aktifitas</p> 	<p>Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja</p>

Tabel 2.4 Lanjutan

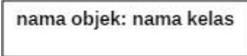
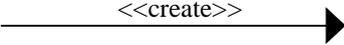
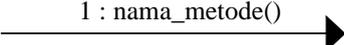
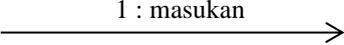
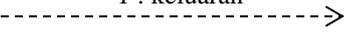
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi

Sumber: A.S. dan Shalahuddin (2011: 134-135)

3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup (*life cycle*) objek dan *message* (pesan) yang dikirimkan dan diterima antar objek. Jumlah *sequence diagram* yang harus digambar minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. Semakin banyak *use case* yang didefinisikan semakin banyak pula *sequence diagram* yang harus dibuat. Simbol-simbol yang digunakan pada *sequence diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin, 2011: 137-139).

Tabel 2.5 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor/actor</p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
<p>Garis hidup/lifeline</p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
<p>Pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat
<p>pesan tipe <i>call</i></p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.
<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju
<p>pesan tipe <i>return</i></p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek penerima
<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri

Sumber: A.S. dan Shalahuddin (2011: 137- 139)

2.4 Penelitian Terdahulu

Peneliti menggunakan telaah penelitian yang berasal dari penelitian terdahulu yang berkaitan dengan permasalahan pada penelitian yang dilakukan oleh peneliti. Penelitian terdahulu bertujuan untuk bahan perbandingan antara penelitian yang sudah dilakukan dan yang akan dirancang oleh peneliti. Beberapa telaah penelitian tersebut diantaranya :

1. **Prasetyo Adi Wibowo dan Sugeng Purbawanto** (2015: 27-34), *Edu Elekrika Journal*, Nomor 4, Volume 2, ISSN : 2252-6811, dengan judul **Sistem Pakar Untuk Mendiagnosa Kerusakan Televisi Dengan Metode *Forward Chaining* Menggunakan *PHP* Dan *MySQL***. Sistem pakar merupakan salah satu bagian dalam kecerdasan buatan. Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang dimasukkan ke dalam komputer untuk memecahkan masalah-masalah yang biasanya diselesaikan oleh pakar. Metode inferensi yang digunakan adalah *forward chaining*, yaitu proses peruntukan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Tujuan penelitian ini adalah merancang dan membangun aplikasi yang dapat membantu pengguna untuk mendiagnosa masalah atau kerusakan pada televisi, sehingga kerusakan yang dialami dapat diketahui dan dapat diatasi dengan solusi yang diberikan.
2. **Zuly Budiarmo, Hersatoto Listiyono dan Eddy Nur Raharjo** (2010: 10-15), *Jurnal Teknologi Informasi DINAMIK*, Nomor 1, Volume XV, ISSN :

0854-9524, dengan judul **Rancang Bangun *User Interface* Untuk Menentukan Tingkat Kerusakan Rangkaian Televisi Dengan Menggunakan Teori Faktor Keyakinan (*Confidence Factor*)**. Penyebab kerusakan televisi adalah karena adanya kerusakan pada suatu rangkaian. Untuk menentukan blok rangkaian yang mengalami kerusakan hanya bisa dilakukan oleh seorang pakar yang ahli di bidang televisi. Sehingga diperlukan suatu cara untuk menampung pengetahuan dari beberapa orang pakar untuk menentukan tingkat kerusakan sebuah rangkaian televisi. Salah satu metode yang digunakan untuk menentukan tingkat kerusakan suatu blok rangkaian televisi dengan menggunakan teori Faktor Kepastian (*Confidence Factor*).

3. **Murawan dan Sufa'atin** (2015: 1-6), Program Studi Teknik Informatika- Universitas Komputer Indonesia, Nomor 1, Volume 4, ISSN : 2089-9033, dengan judul **Pembangunan Aplikasi Media Pembelajaran Dan Simulasi *Troubleshooting* Televisi**. Media pembelajaran merupakan segala sesuatu yang dapat digunakan oleh seorang tenaga pendidik dalam menyampaikan materi pembelajaran kepada para peserta didiknya. simulasi merupakan satu metode pelatihan yang memperagakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan yang sesungguhnya. *Troubleshooting* adalah sebuah istilah dalam bahasa inggris, yang merujuk kepada sebuah bentuk penyelesaian sebuah masalah. Televisi adalah sistem elektronis yang menyampaikan suatu isi pesan dalam bentuk audiovisual, dengan adanya televisi semua orang akan lebih mudah mengetahui yang ada di belahan

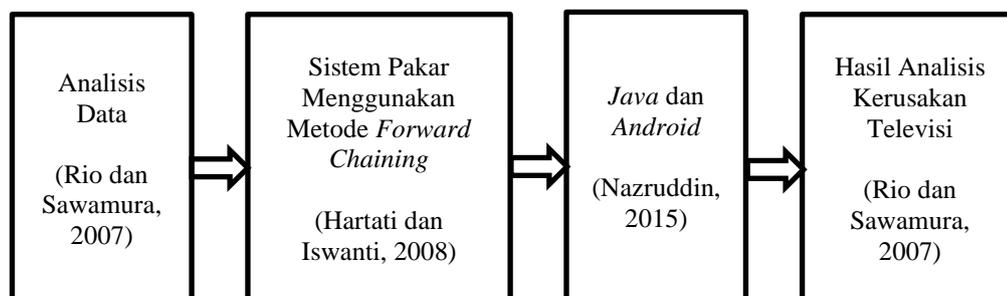
dunia lain. Ketika seseorang telah memiliki televisi, diharapkan tahu bagaimana cara memperbaiki televisi.

4. **Abid Yanuar Badharudin dan Dwi Aryanto** (2011: 133-141), Teknik Informatika – F. Teknik – Universitas Muhammadiyah Purwokerto, Nomor 4, Volume 1, ISSN : 2086-9398, dengan judul **Aplikasi Sistem Pakar untuk Mendiagnosa Kerusakan Monitor CRT (*Cathode Ray Tube*)**. Layar monitor merupakan komponen utama pendukung keutuhan suatu perangkat. Sistem pakar adalah salah satu cabang dari AI (*Artificial Intelligence*) yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. runut maju atau disebut penalaran dari bawah ke atas dikarenakan penalaran *evidence* (fakta) pada level bawah menuju konklusi pada level atas didasarkan pada fakta. Fakta merupakan satuan dari paradigma berbasis pengetahuan karena mereka tidak dapat diuraikan ke dalam satuan paling kecil yang mempunyai makna
5. **Deny Wiria Nugraha** (2014: 113-126), INFORMATIKA, Nomor 2, Volume 10, dengan judul **Sistem Pakar Untuk Mendiagnosa Kerusakan Perangkat Televisi Menggunakan Metode *Backward Chaining***. Sebagai peralatan elektronika, perangkat televisi tentu saja akan mengalami gangguan atau kerusakan. Hal ini akan sangat mengganggu pengguna televisi karena jika hal itu terjadi, maka program siaran televisi yang disukai akan terlewatkan tanpa bisa ditonton. Dari pengetahuan dan pengalaman yang dimiliki oleh teknisi senior, membuat teknisi senior tersebut mampu

dengan cepat mengetahui penyebab kerusakan peralatan elektronika, misalnya perangkat televisi berdasarkan jenis kerusakan yang ada. Tujuan dari penelitian ini adalah merancang dan membangun sistem pakar yang dapat mendiagnosa kerusakan pada perangkat televisi berdasarkan jenis dan ciri kerusakan yang ada dengan menggunakan metode *backward chaining*.

2.5 Kerangka Pemikiran

Kerangka pemikiran adalah model konseptual bagaimana teori berhubungan dengan faktor-faktor yang diidentifikasi sebagai masalah yang penting. Kerangka pemikiran menjelaskan secara teori yang saling berhubungan antar variabel yang diteliti (Sugiyono, 2012: 60). Bentuk kerangka pemikiran yang mendasari penelitian ini yaitu :



Gambar 2.16 Kerangka Pemikiran
(Sumber: Data Penelitian: 2017)

Data-data tentang permasalahan televisi tabung awalnya dianalisis terlebih dahulu agar mudah dilakukan proses pengolahan datanya. Data-data kemudian diolah menggunakan metode *forward chaining* untuk membuat aturan (*rule*) yang akan digunakan. Sistem pakar ini dibuat menggunakan bahasa pemrograman *Java*

dan *Android* sehingga nantinya dapat menampilkan hasil analisis kerusakan televisi dari aplikasi sistem pakar tersebut.