

SISTEM PAKAR PENDETEKSI MASALAH PADA HARDWARE KOMPUTER MENGGUNAKAN METODE BACKWARD CHAINING

Dewi Safutri¹, Dila Fadilah², Hasbi Daru Pratama³, Ikhfan Yusuf Arifin³, Perani Rosyani⁴
¹⁻⁴Universitas Pamulang; Jl. Surya Kencana No.1, Pamulang Barat, (021) 741-2566 atau 7470 9855
¹⁻⁴Jurusan Teknik Informatika, Fakultas Teknik, Universitas Pamulang

e-mail: *¹ dewisafutri24@gmail.com

Abstrak

Sistem pakar adalah salah satu cabang perangkat lunak dari AI (Artificial Intelligence) yang digunakan untuk menyelesaikan berbagai masalah diantaranya untuk mendeteksi masalah pada hardware komputer dengan menggunakan metode backward chaining. Salah satu teknik untuk membuat komputer mampu mengolah pengetahuan ini disebut teknik kecerdasan buatan (Artificial Intelligence Techique). Dalam penelitian ini komputer merupakan kebutuhan utama untuk menunjang kinerja manusia dan komputer juga sering merusak perangkat keras. Masalah pada hardware komputer bukanlah hal yang mudah untuk diatasi oleh pengguna awam. Dengan ini dapat membantu para user untuk dapat mendeteksi kemungkinan kerusakan yang terjadi, sehingga dapat mempersingkat waktu untuk proses perbaikan. Dengan pendekatan ini manusia mencoba membuat komputer dapat berpikir seperti cara yang dipakai manusia dalam memecahkan masalah computer dan tugas ini bertujuan untuk membangun sistem berbasis pengetahuan untuk mendeteksi kerusakan perangkat keras komputer dengan menggunakan metode backward chaining.

Kata kunci : Sistem Pakar, Metode Backward Chaining , Masalah Hardware Komputer

I. PENDAHULUAN

Penggunaan komputer pada masyarakat era digital ini sudah bisa dibilang menjadi kebutuhan primer. Kerusakan pada perangkat keras komputer merupakan hal yang mungkin terjadi suatu saat. Namun kebanyakan orang(awam) akan mudah panik saat terjadi kerusakan pada perangkat keras komputer dan cenderung takut untuk memperbaiki karena memang tidak mengetahui kerusakannya. Oleh karena permasalahan di atas maka dirasa perlu adanya software yang dpat mendeteksi kerusakakan pada hardware komputer tanpa harus tahu banyak tentang hardware komputer.

Sistem pakar dibuat untuk mengatasi permasalahan yang spesifik di suatu bidang berdasarkan cara-cara yang dilakukan seorang pakar, misalnya pada kasus kali ini yaitu bidang hardware komputer. Permasalahan biaya dan waktu perbaikan yang saat ini menjadi permasalahan jika terjadi kerusakan pada hardware komputer diharapkan bisa diatasi dengan adanya aplikasi sistem pakar untuk mendeteksi kerusakan

hardware komputer. Sehingga orang awam pun dapat memperbaiki masalah pada hardware komputer tanpa harus mengeluarkan biaya dan membuang waktu.

Pada tugas prosiding ini akan dirancang suatu perangkat lunak yang dapat membantu orang dengan pengetahuan hardware komputer terbatas, teknisi dan tempat pelatihan komputer untuk mengidentifikasi permasalahan hardware yang umum terjadi. Perangkat lunak ini dapat digunakan user untuk mengidentifikasi permasalahan hardware dengan cara user memilih masalah yang dialami, kemudian perangkata lunak akan akan melakukan pemeriksaan berdasarkan algoritma yang diterapkan dalam aplikasi dan pada akhirnya mendapatkan informasi penyebab permasalahan beserta solusinya.

II. METODE PELAKSANAAN

Metode Backward chaining

Backward chaining menggunakan pendekatan goal driven yang biasanya dimulai dari harapan apa yang akan terjadi dan selanjutnya mencari bukti yang

mendukung atau bertentangan dengan harapan (Minarni:2013). Backward chaining ini membutuhkan perumusan dan pengujian hipotesis sementara.

1. Basis Pengetahuan (Knowledge Base)

Basis pengetahuan adalah bagian penting dari sistem pakar. Kesuksesan atau keberhasilan dalam suatu sistem pakar tergantung pada basis pengetahuan yang ada didalamnya. Basis pengetahuan berisi tentang fakta-fakta, teori, pemikiran, maupun aturan-aturan yang mengarahkan penggunaan pengetahuan dalam suatu menyelesaikan masalah yang akan diterapkan ke dalam sistem komputer dengan menggunakan metode representasi pengetahuan tertentu.

Metode representasi pengetahuan itu sendiri adalah cara untuk menstrukturkan pengetahuan yang dimiliki oleh pakar agar mudah diolah oleh komputer. Jadi basis pengetahuan merupakan satu komponen yang sangat penting dalam sistem pakar karena menyimpan semua pengetahuan yang akan dipakai sebagai dasar pengambilan keputusan.

2. Inference Engine

Inference Engine merupakan bagian dari sistem pakar yang bertugas untuk menemukan solusi yang tepat dari banyaknya solusi yang ada. Fungsi Inferensi Engine itu sendiri ialah memberikan pertanyaan kepada user, menambah jawaban pada working memory (balckboard), menambahkan fakta baru dari suatu rule (hasil inferensi), menambahkan fakta baru tersebut pada working memory, dan mencocokkan fakta pada working memory dengan rule. Inference Engine merupakan otak dari sistem pakar, juga struktur control/interpreter. Komponen ini berupa suatu program komputer yang menyediakan suatu metodologi untuk mempertimbangkan informasi dalam knowledge base dan merumuskan kesimpulan. keahlian inferensi, dan kemampuan menjelaskan. Keahlian merupakan kelebihan penguasaan pengetahuan dibidang tertentu yang diperoleh dari pelatihan, membaca, atau pengalaman (Turban, Aronson, & Liang, 2005). Pengalihan keahlian dari para ahli komputer untuk dialihkan ke orang lain yang bukan ahli adalah tujuan utama dari sistem pakar. Proses ini membutuhkan empat aktivitas antara lain tambahan pengetahuan, representasi pengetahuan (ke komputer), inferensi pengetahuan, dan pengalihan pengetahuan ke user. Pengetahuan yang disimpan dikomputer disebut nama basis pengetahuan. Ada dua tipe basis pengetahuan yaitu fakta dan prosedur. Kemampuan untuk menalar merupakan kemampuan yang harus dimiliki oleh suatu sistem pakar. Semua

3. Explanation Subsystem

Explanation Subsystem digunakan untuk memberikan respon atau penjelasan tentang apa yang dilakukan sistem pakar. Digunakan untuk mengklarifikasi proses reasoning, rekomendasi dan tindakan lainnya (misalnya : melalui pertanyaan).

4. User Interface

User Interface merupakan media komunikasi antar pemakai (user) dengan program, yang menyediakan dan memberikan fasilitas informasi dan beberapa keterangan yang mengarah pada penelusuran masalah sampai ditemukan solusi.

5. Knowledge Base Editor

Knowledge Base Editor merupakan bagian yang digunakan untuk menambah, menghapus dan memperbaiki basis pengetahuan.

6. Learning

Learning adalah suatu proses belajar dari suatu sistem pakar apabila sistem tidak menemukan solusi masalah.

7. Certainty Factor

Certainty Factor merupakan faktor keyakinan atas fakta-fakta yang ada.

8. Explanation Subsystem

Explanation Subsystem Merupakan kemampuan untuk memberikan penjelasan dari kesimpulan atau keputusan yang diberikan.

9. Knowledge acquisition

Akuisisi Pengetahuan merupakan proses ekstraksi, strukturisasi, dan mengorganisasikan pengetahuan dari satu sumber atau lebih.

Konsep Dasar Sistem Pakar

Sistem pakar pada konsep dasarnya mengandung keahlian, ahli, aturan, pengalihan data keahlian-keahlian sudah tersimpan dalam database computer sebaai basis pengetahuan yang dibaca oleh program komputer. Aplikasi komputer tersebut harus di rancang agar dapat membentuk inferensi. Proses inferensi dapat dikemas dalam bentuk mesin inferensi (inference engine).

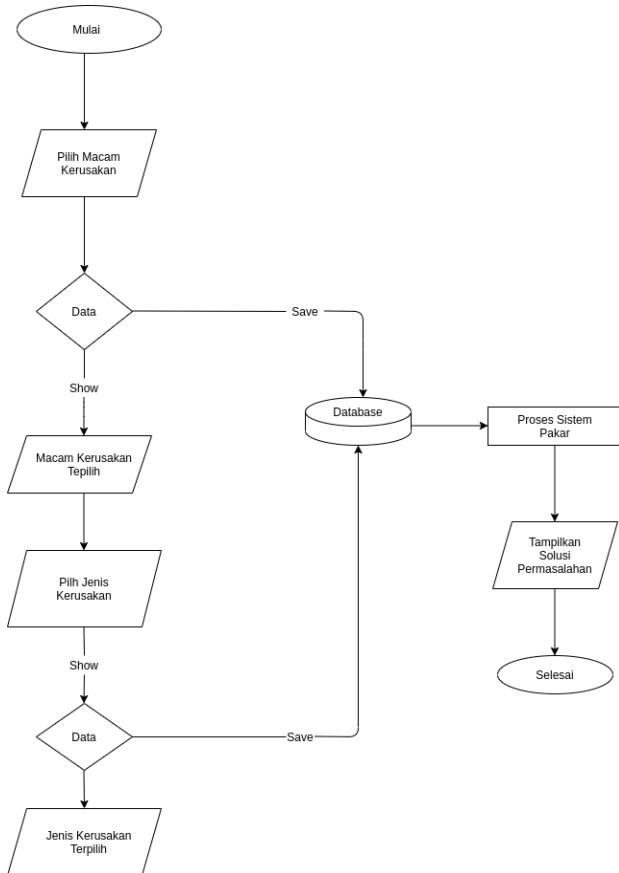
Algoritma

Menurut Goodman Hedet Niemi, algoritma adalah urutan terbatas dari operasi-operasi terdefinisi dengan baik, yang masing-masing membutuhkan memori dan waktu yang terbatas untuk menyelesaikan suatu masalah.

Pada sistem ini dirancang untuk mempermudah bagi orang awam dalam memperbaiki kerusakan komputer.

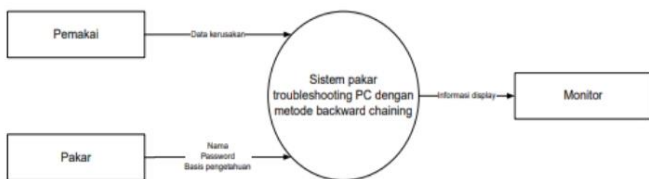
Namun ada alat bantu perancangan yang akan dipakai dalam sistem ini adalah:

Bagan alair sistem (system flowchart) adalah bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada dalam di dalam sistem, bagan alir sistem menunjukkan apa yang dikerjakan di system.

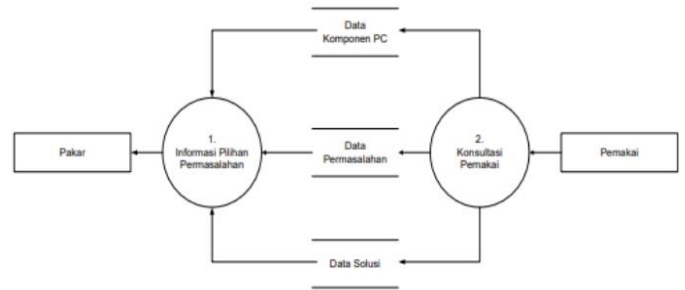


Gabar 1. Sistem Flowchart

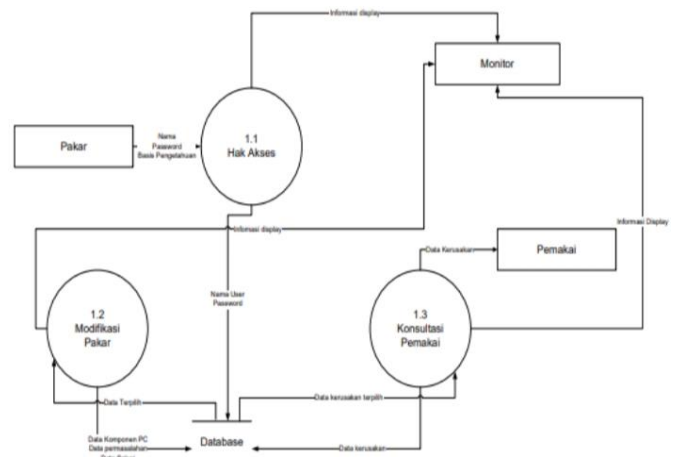
DFD (*Data Flow Diagram*) berfungsi untuk memperlihatkan alur sistem yang dibuat oleh penulis secara spesifik (perluasan dari DCD). Diagram alir data dalam sistem yang akan dibangun itu secara paralel maupun terstruktur dengan melibatkan komponen-komponen entitas-entitas yang terkait baik entitas luar maupun dalam, media penyimpanan (*storage*), proses-proses sistem maupun simbol panah yang menunjukkan hubungan alir data dari proses ke entitas.



Gambar 2. Diagram Konteks



Gambar 3. Data Flow of Diagram Zero



Gambar 4. Data Flow of Diagram Detail

Algoritma pada kerusakan komputer dan periferal

Untuk lebih jelasnya penulis akan memberikan proses pencarian solusi sebagai perbandingan pada tabel kerusakan komputer sesuai dengan jenis kerusakan masing-masing dengan menggunakan algoritma sebagai berikut :

Algoritma Name Menu Utama

- Start
- Var
- Pilih = chart
- Repeat
- Read (Pilih)
- Case Pilih Of
 - 1: Komputer tidak mau hidup
 - 2: Komputer tidak mau booting
 - 3: Motherboard bermasalah
 - 4: Keyboard bermasalah
 - 5: Mouse bermasalah
 - 6: Monitor bermasalah
 - 7: Printer bermasalah
 - 8: Harddisk bermasalah
 - 9: Keluar
- End Case
- Until pilih = 9
- Stop

Algoritma Name Komputer tidak mau hidup

```
Repeat
Read (Pilih)
Case Pilih Of
1: Tombol power tidak terpasang sempurna
2: Power supply rusak
3: Stabilizer rusak
0: Kembali
End Case
Until pilih=0
IF Tombol power tidak terpasang sempurna = True
THEN
```

```
    Write('Pasang kembali dengan benar')
```

```
ELSE Kabel power rusak = True THEN
```

```
    Write('Ganti kabel power')
```

```
ELSE
```

```
IF Power supply rusak = True THEN
```

```
    Write('Ganti power supply')
```

```
ELSE
```

```
IF Stabilizer rusak = True THEN
```

```
    Write('Ganti Stabilizer')
```

```
END IF
```

Algoritma Name Komputer tidak mau booting

```
Repeat
```

```
Read (Pilih)
```

```
Case Pilih Of
```

```
1: Kabel harddisk rusak
```

```
2: RAM bermasalah
```

```
3: VGA bermasalah
```

```
0: Kembali
```

```
End Case
```

```
Until pilih=0
```

```
IF Kabel harddisk rusak = True THEN
```

```
    Write('Ganti')
```

```
ELSE
```

```
IF RAM tidak terpasang sempurna = True THEN
```

```
    Write('Pasang kembali dengan benar')
```

```
ELSE
```

```
IF Slot RAM tergores atau cacat = True THEN
```

```
    Write('Ganti RAM')
```

```
ELSE
```

```
IF VGA rusak = True THEN
```

```
    Write('Ganti')
```

```
END IF
```

Algoritma Name Komputer sering hang

```
IF Jenis memori tidak cocok = True THEN
```

```
    Write('Ganti jenis memori')
```

```
ELSE
```

```
IF Harddisk pada harddisk = True THEN
```

```
    Write('Lakukan low level pada harddisk')
```

```
ELSE
```

```
IF Pengaruh overclocking = True THEN
```

```
    Write('Kembalikan setting komputer menjadi normal kembali')
```

```
END IF
```

Algoritma Name Proses komputer lambat

```
IF Memori kurang cukup besar = True THEN
```

```
    Write('Lakukan penambahan memori dan uninstall program yang tidak perlu')
```

```
ELSE
```

```
IF Prosesor terlalu panas = True THEN
```

```
    Write('Cek kembali kipas dan pendingin prosesor')
```

```
ELSE
```

```
IF Harddisk telah terinfeksi virus = True THEN
```

```
    Write('Scan harddisk dengan antivirus yang telah di-update')
```

```
END IF
```

Algoritma Name Komputer restart sendiri

```
IF Power supply tidak normal = True THEN
```

```
    Write('Ganti dengan yang baru atau dengan daya yang besar')
```

```
ELSE
```

```
IF Motherboard berdebu = True THEN
```

```
    Write('Bersihkan motherboard dengan kuas')
```

```
END IF
```

Algoritma Name Keyboard tidak terdeteksi BIOS

```
IF Konektor keyboard longgar = True THEN
```

```
    Write('Matikan komputer copot keyboard dan pasang kembali dengan benar')
```

```
ELSE
```

```
IF Keyboard rusak = True THEN
```

```
    Write('Ganti keyboard yang baru')
```

```
ELSE
```

```
IF I/O Port rusak = True THEN
```

```
    Write('Ganti')
```

```
END IF
```

Algoritma Name Tombol Keyboard tidak berfungsi

```
IF PCB kotor = True THEN
```

```
    Write('Bersihkan dengan cairan alkohol')
```

```
ELSE
```

```
IF IC Keyboard tidak berfungsi = True THEN
```

```
    Write('Ganti IC')
```

```
END IF
```

Algoritma Name Mouse tidak terdeteksi BIOS

```
IF Konektor mouse longgar = True THEN
```

```
    Write('Matikan mouse, copot keyboard dan pasang kembali dengan baik')
```

```
ELSE IF Mouse rusak = True THEN
```

```
    Write('Ganti mouse')
```

```
ELSE IF I/O Port mouse rusak = True THEN
```

```
    Write('Ganti')
```

```
END IF
```

Algoritma Name Monitor tidak menampilkan gambar

```
IF Kabel power bermasalah = True THEN
    Write('Cek arus kabel power')
ELSE
IF Kabel power rusak = True THEN
    Write('Ganti kabel power')
ELSE
IF Kabel monitor ke CPU kendor = True
THEN
    Write('Kencangkan dan baud konektor ke
CPU')
ELSE
IF VGA Rusak = True THEN
    Write('Ganti VGA')
ELSE
IF Indikator On layar gelap = True THEN
    Write('Ganti Flayback')
ELSE
IF Hanya ada garis pada monitor = True THEN
    Write('Atur vertical loop dan horizontal loop')
ELSE
IF Tampilan tidak sesuai = True THEN
    Write('Setting ulang resolusi monitor')
END IF
Algoritma Name Printer tidak mau beroperasi
IF Kabel power bermasalah = True THEN
    Write('Cek arus AC yang memasuki printer')
ELSE
IF Power supply printer rusak = True THEN
    Write('Ganti power supply printer')
END IF
Algoritma Name Printer tidak mau mencetak
IF Driver printer belum terinstall = True THEN
    Write('Install driver printer')
ELSE
IF Head printer rusak = True THEN
    Write('Ganti head printer')
ELSE
IF Motor printer rusak = True THEN
    Write('Ganti motor printer')
END IF
Algoritma Name Hasil print tidak normal
IF Tinta habis = True THEN
    Write('Isi ulang tinta printer')
ELSE
IF Head printer kotor = True THEN
    Write('Cleaning catridge')
ELSE
IF Catridge atau head printer rusak = True THEN
    Write('Ganti catridge printer')
END IF
Algoritma Name Harddisk tidak dikenali BIOS
IF Kabel tidak terpasang dengan benar = True THEN
```

```
    Write('Cek kabel data dan power pastikan
terpasang dengan baik')
ELSE
IF Kabel data atau power rusak = True THEN
    Write('Ganti Kabel data atau power')
ELSE
IF Jumper setting salah = True THEN
    Write('Lakukan setting jumper dengan benar')
ELSE
IF Setting BIOS tidak benar = True THEN
    Write('Lakukan setting BIOS dengan benar')
ELSE
IF Harddisk rusak = True THEN
    Write('Ganti harddisk')
END IF
```

IV. SIMPULAN

Pada sistem ini dirancang untuk mempermudah bagi orang awam dalam memperbaiki kerusakan komputer.

DAFTAR PUSTAKA

- Anton Mulyono. 1990. Pengantar Kecerdasan Buatan, Dinastindo. Jakarta.
- Deni Arifiyanto dan Ari Funatik. 2009 Antigaptek Hardware Komputer, Kawan Pustaka. Jakarta.
- Euis Marlina. 2009. 10 Jsenis Koneksi Delphi Ke Database. Gava Media. Yogyakarta.
- Farid Azin. Belajar Sendiri Pemrograman Sistem Pakar, PT. Elekmedia Komputindo – Kelompok Gramedia. Jakarta.
- Kusnassriyanto Saiful Bahri dan Wawan Sjachriyanto, 2008. Teknik Pemrograman Delphi, Informatika. Bandung.
- Raymond Mc Leoid, Jr. 2001. Sistem Informasi Manajemen, Prenhalindo. Jakarta.
- Rinaldi Munir. 2007. Algoritma Dan Pemrograman Dalam Bahasa Pascal Dan C. Inforamtika. Bandung.
- Suparman. 1991. Mengenal Artificial Intelligence. Andi Offset. Yogyakarta.
- Teguh Wahyono. 2005. Pc Troubleshooting Plus. Gava Media. Yogyakarta.
- Yogianto. HM. 1990. Analisa Dan Desain Sistem, Andi Offset. Yogyakarta.

- Wawan Kusdiawan, M.Kom, Cara Mudah Dan Cepat Membuat Program Aplikasi Database.
- Arhami, Muhammad. 2005. Konsep Dasar Sistem Pakar. Penerbit Andi. Yogyakarta.
- Wijayana Yenita. 2019. Sistem Pakar Kerusakan Hardware Komputer Dengan Metode Backward Chaining Berbasis Web, Semarang.
- Yanti, Sari Noorlima Dan Endah Budiyati. 2020. Aplikasi Sistem Pakar Untuk Mendiagnosa Virus Covid-19 Pada Manusia Berbasis Web Menggunakan Metode Forward Chaining. *Jurnal Informatika Universitas Pamulang*, 5(4), 451-458.
- Nugroho, Fajar Agung. 2018. Perancangan Sistem Pakar Diagnosa Penyakit Jantung Dengan Metode Forward Chaining. *Jurnal Informatika Universitas Pamulang*, 3(2), 75-79.
- Samsudin , Indrawan , dan Sri Mulyati. 2020. Perancangan Sistem Informasi Pembelajaran Algoritma dan Pemrograman Berbasis Web pada Program Studi Teknik Informatika STMIK ERESHA. *Jurnal Informatika Universitas Pamulang*, 5(4), 521-528.