

# SISTEM PAKAR UNTUK MENDETEKSI KERUSAKAN AMPLIFIER MENGUNAKAN METODE CASE BASED REASONING BERBASIS ANDROID

Wiranto<sup>1</sup>, Nelly Astuti Hasibuan<sup>2</sup>, Surya Darma Nasution<sup>3</sup>

Program Studi Teknik Informatika STMIK Budi Darma, Medan, Indonesia  
Jl. Sisingamangaraja No. 338 Simpang Limun, Medan

## ABSTRAK

*Sistem pakar bagian dari kecerdasan buatan yang dikembangkan manusia untuk memanfaatkan pengetahuan yang diperoleh melalui penginputan data dari kemampuan pakar agar dapat menyelesaikan suatu masalah, terutama dalam mendeteksi kerusakan suatu alat. Salah satunya yaitu alat elektronik berupa Amplifier, sistem pakar tersebut akan membantu user dalam menangani kerusakan Amplifier. Metode yang digunakan adalah metode CBR (Case Based Reasoning) dan algoritma Nearest Neighbor, metode ini bekerja dengan cara mengolah data basis pengetahuan kasus lama untuk menyelesaikan kasus baru dengan cara mencocokkan gejala yang tersimpan pada basis pengetahuan. Sistem pakar ini dirancang berbasis Android, diharapkan sistem pakar berbasis Android ini user dapat dengan mudah mendeteksi kerusakan dan mencegah kerusakan yang fatal serta mudah dan efisien dalam menggunakannya.*

**Kata kunci :** Sistem Pakar, Amplifier, Case Based Reasoning, Nearest Neighbor

## I. PENDAHULUAN

Kemajuan teknologi mendorong masyarakat untuk terus menciptakan suatu terobosan di segala bidang, terutama pada bidang kecerdasan buatan (*Artificial Intelligence*). Salah satu bentuk dari teknologi kecerdasan buatan adalah sistem pakar, sistem ini memiliki kemampuan untuk mengadopsi suatu dasar pengetahuan (*knowledge base*) yang diperoleh melalui pakar yang digunakan untuk memecahkan masalah yang biasanya hanya dapat diselesaikan oleh seorang pakar dalam bidang tertentu. Penggunaan sistem pakar akan berpengaruh pada bidang tertentu, salah satunya adalah sistem pakar yang dapat diaplikasikan ke perbaikan alat-alat elektronik.

Pada masa ini sudah banyak masyarakat yang menggunakan peralatan elektronik dalam kehidupan sehari-hari, salah satunya adalah *Amplifier* yang sudah menjadi kebutuhan sekunder. *Amplifier* adalah alat elektronik yang digunakan sebagai penguat daya dalam audio. Elektronik merupakan ilmu atau kemampuan yang hanya dimiliki oleh beberapa individu saja, hal ini yang dapat menyulitkan masyarakat umum yang tidak memiliki pengetahuan atau pengalaman dibidang elektronika dalam mendeteksi kerusakan *Amplifier*. Minimnya pengetahuan dibidang elektronik menyebabkan masyarakat bergantung pada teknisi elektronik, untuk perbaikannya juga dapat memakan biaya yang cukup besar apabila sudah mengalami kerusakan fatal.

Hal ini yang mendasari diperlukannya suatu perancangan aplikasi sistem pakar pada kerusakan *Amplifier*, terdapat beberapa metode yang dapat digunakan dalam sistem pakar, salah satunya adalah metode CBR (*Case Based Reasoning*). Pada penelitian sebelumnya metode CBR sudah pernah digunakan dalam sistem pakar untuk mendiagnosa

suatu penyakit diantaranya yaitu, pengembangan sistem cerdas menggunakan penalaran berbasis kasus (*Case Based Reasoning*) untuk diagnosa penyakit akibat virus Eksantema yang diteliti oleh Agus Sasmito Aribowo[1]. Dan penelitian *Case Based Reasoning* untuk diagnosa penyakit THT (telinga hidung dan tenggorokan) yang diteliti oleh Tedy Rismawan dan Sri Hartati[2].

## II. TEORITIS

### A. Sistem Pakar

Sistem pakar (*Expert System*) merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*), sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar[6]. Dengan kata lain sistem pakar mampu memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar yang dapat menyelesaikan masalah. Agar sistem pakar lebih menyerupai seorang pakar maka harus dilengkapi dengan fasilitas seperti fasilitas penjelasan (*explanation facility*) dan fasilitas akuisisi pengetahuan (*knowledge acquisition facility*). Konsep sistem pakar berbasis pengetahuan dapat diilustrasikan pada gambar di bawah ini[3]:

### B. Metode Case Based Reasoning

*Case Based Reasoning* (CBR) merupakan metode yang menitikberatkan pemecahan masalah dengan berdasarkan pada pengetahuan dari kasus-kasus terdahulu [4]. Dengan kata lain metode *Case Based Reasoning* menyelesaikan masalah dengan mengadaptasi solusi-solusi yang pernah digunakan pada kasus sebelumnya yang sudah tersimpan di basis pengetahuan. secara umum metode *Case Based*

*Reasoning* memiliki empat langkah tahapan proses, yaitu[4]:

1. *Retrieve*, pada proses ini dilakukan pencarian kembali kasus yang paling menyerupai antara kasus lama dengan kasus yang baru.
2. *Reuse*, pada langkah ini menggunakan kembali pengetahuan dan informasi kasus lama berdasarkan bobot kemiripan yang relevan untuk mengatasi kasus yang baru.
3. *Revise*, melakukan peninjauan kembali solusi yang diberikan oleh sistem, apabila solusi kurang tepat maka dilakukan perbaikan kembali.
4. *Retain*, melakukan analisa bagian dari solusi yang baru untuk selanjutnya disimpan ke dalam *knowledge base* untuk digunakan dalam pemecahan masalah selanjutnya.

### C. Algoritma Nearest Neighbor

Algoritma *Nearest Neighbor Retrieval* adalah sebuah algoritma untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut[5]. Langkah *retrieve* pada metode *Case Based Reasoning* menggunakan algoritma *Nearest Neighbor*, dilakukan dengan cara pembobotan lalu mencari kelompok objek dalam data lama yang paling dekat dengan data baru untuk menghitung kemiripan kasus. Adapun rumus untuk menghitung bobot kemiripan menggunakan algoritma *Nearest Neighbor Retrieval* adalah sebagai berikut[4]:

$$\text{Similarity}(p, q) = \frac{S_1 \times W_1 + S_2 \times W_2 + \dots + S_n \times W_n}{W_1 + W_2 + \dots + W_n}$$

Keterangan :

p : Kasus baru

q : Kasus yang tersimpan pada (*case*)

w : *weight* (bobot yang diberikan pada atribut ke-1)

s : *similarity* (nilai kemiripan) yaitu 1 (sama) dan 0 (tidak sama)

## III. ANALISA

### A. Anaisa Masalah

Perancangan aplikasi sistem pakar untuk mendeteksi kerusakan *Amplifier* diawali dengan pengumpulan data gejala serta melakukan konsultasi kepada teknisi, selanjutnya data-data tersebut direpresentasikan dalam bentuk berbasis aturan (*rule*). Sistem konsultasi yang dirancang akan mendapatkan hasil diagnosa berdasarkan data-data yang sudah ditentukan dalam bentuk *rule* tersebut. Berikut ini adalah penentuan nilai pakar untuk perhitungan metode *Case Based Reasoning* :

Tabel 1 Kriteria Kemiripan Kasus

Kriteria Kemiripan	Nilai Desimal Kemiripan
Tinggi	0,80-1
Sedang	0,40-0,79
Rendah	0-0,39

Tabel 2 Tabel Bobot Jawaban *User*

Keterangan	Bobot
Ya	1
Tidak	0

Berikut ini merupakan tabel ketentuan pakar yang diperoleh dari teknisi untuk mendeteksi kerusakan *Amplifier* menggunakan metode *Case Based Reasoning* yang nantinya dapat digunakan oleh sistem untuk memecahkan masalah yang dialami pada user.

Tabel 3 *Rule Based* Mendeteksi Kerusakan *Amplifier*

No	Gejala Kerusakan	Bagian Kerusakan	Kode Solusi
1	Tidak ada suara (G1) Socket RCA longgar atau goyang (G2) Socket kabel <i>input</i> sinyal longgar (G3) Socket kabel arus pada <i>input</i> sinyal longgar (G4) Suara yang keluar serak (G5)	<i>Input</i> sinyal (K1)	S1
2	Suara yang keluar serak (G5) Suara bass tidak ada (G6) Suara hilang timbul (G7) Suara yang keluar pelan tidak bisa maksimal (G8) Suara pada <i>microfon</i> tidak keluar (G9) Socket kabel arus dari <i>power amplifier</i> ke <i>tune control</i> longgar (G10) Suara mic 1 dengan mic 2 tidak seimbang (G27)	<i>Pre-amp/Tune control</i> (K2)	S2
3	Suara yang keluar cacat seperti mendengung ataupun serak (G11) <i>Transistor</i> terminal akhir apabila disentuh panas ataupun mengalami kerusakan seperti gosong atau pecah (G12) <i>Amplifier</i> mati tetapi lampu <i>power</i> masih menyala (G13) <i>Resistor</i> batu apabila disentuh panas ataupun mengalami kerusakan seperti gosong atau pecah (G14) <i>Transistor</i> kecil apabila disentuh	<i>Power Amplifier</i> (K3)	S3

No	Gejala Kerusakan	Bagian Kerusakan	Kode Solusi
	terasa panas ataupun mengalami kerusakan seperti gosong atau pecah (G15) Suara vokal hilang (G16) <i>Resistor</i> 2w (besar) pada bagian <i>power amplifier</i> gosong (G17) Terdapat <i>dioda zener</i> yang pecah (G28)		
4	<i>Amplifier</i> mati total (G18) <i>Elco</i> mengalami kerusakan seperti gembung ataupun bocor (G19) Solderan pada kaki <i>elco</i> longgar (G20) <i>Resistor</i> kecil pada bagian <i>power supply</i> gosong (G21) Suara yang keluar serak (G5) <i>Resistor</i> 2w (besar) pada bagian <i>power supply</i> gosong (G22) Apabila <i>Amplifier</i> dihidupkan listrik lompat (mati) (G23) Skring ( <i>fuse</i> ) putus (G24) Lampu power tidak menyala (G25) Kipas fan pada <i>Amplifier</i> mati (G26) <i>Dioda</i> penyearah terasa panas (G29) Tembaga <i>trafo</i> gosong (G30) <i>Amplifier</i> mati tetapi lampu power masih menyala (G13)	<i>Power Supply</i> (K4)	S4

Untuk melakukan proses diagnosa maka dilakukan perhitungan berdasarkan kedekatan kasus lama (yang merupakan basis pengetahuan yang dimiliki sistem dan pernah ditangani oleh seorang teknisi), berikut ini adalah tabel kasus lama (*training*) yang akan dicari nilai kemiripannya dengan kasus baru yang dimasukkan oleh *user*.

Tabel 4 Tabel Kasus *Training*

ID Kasus Lama	Kode Gejala Kerusakan	Kode Bagian Kerusakan
ID001	G1, G3, G4	K1
ID005	G5, G19, G21, G22, G26, G29	K4
ID012	G11, G12, G14, G15, G28	K3
ID021	G5, G6, G8	K2

Kemudian menghitung tingkat kemiripannya dengan contoh kasus baru yang dimasukkan oleh *user*.

Tabel 5 Tabel Kasus Baru

Kasus Baru X
Gejala : Suara yang keluar serak (G5) <i>Elco</i> mengalami kerusakan seperti gembung ataupun bocor (G19) Kipas fan pada <i>Amplifier</i> mati (G26) <i>Dioda</i> penyearah terasa panas (G29)
Kerusakan : ?

Tahapan pertama adalah proses *retrieve* dengan menghitung kedekatan kasus lama dengan kasus baru menggunakan algoritma *Nearest Neighbor*. Berikut adalah contoh perhitungan kedekatan gejala antara kasus baru dengan kasus lama.

Tabel 6 Perhitungan Bobot Kasus Lama ID001 Dengan Kasus Baru

Gejala Kerusakan <i>Amplifier</i>		Kemiripan Gejala	Bobot	
Kasus Lama ID001	Kasus Baru X		MB	MD
G1	-	0	0.1	0.9
G3	-	0	0.03	0.97
G4	-	0	0.06	0.94

$$\begin{aligned}
 \text{Similarity (X, ID001)} &= \frac{[(0*0.1)+(0*0.03)+(0*0.6)]}{0.1+0.03+0.06} \\
 &= \frac{0+0+0}{0.1+0.03+0.06} = \frac{0}{0.19} = 0 \text{ (Rendah)}
 \end{aligned}$$

Tabel 7 Perhitungan Bobot Kasus Lama ID005 Dengan Kasus Baru

Gejala Kerusakan <i>Amplifier</i>		Kemiripan Gejala	Bobot	
Kasus Lama ID005	Kasus Baru X		MB	MD
G5	G5	1	0.13	0.87
G19	G19	1	0.06	0.94
G21	-	0	0.03	0.97
G22	-	0	0.06	0.94
G26	G26	1	0.03	0.97
G29	G29	1	0.3	0.7

$$\begin{aligned}
 \text{Similarity (X, ID005)} &= \frac{[(1*0.13)+(1*0.06)+(0*0.03)+(0*0.06)+(1*0.03)+(1*0.3)]}{0.13+0.06+0.03+0.06+0.03+0.3} \\
 &= \frac{0.13+0.06+0+0+0.03+0.3}{0.13+0.06+0.03+0.06+0.03+0.3} = \frac{0.52}{0.61} \\
 &= 0.85 \text{ (Tinggi)}
 \end{aligned}$$

Selanjutnya adalah tahapan *reuse*, yaitu menampilkan kemiripan kasus. Hasil dari perhitungan kedekatan antara kasus lama dengan baru ada pada tabel berikut ini :

Tabel 8 Hasil Perhitungan *Similarity*

ID KASUS	Bagian Kerusakan	Nilai Kedekatan
ID001	Input sinyal	0
ID005	Power supply	0.85
ID012	Power Amplifier	0
ID021	Pre-amp/Tune control	0.33

Pada proses ini solusi yang diberikan adalah solusi dengan bobot kemiripan kasus yang paling tinggi, dalam contoh kasus ini adalah kasus yang paling tinggi yaitu kasus ID005 dengan bagian kerusakan *power supply* dengan nilai kedekatan 0,85.

#### IV. IMPLEMENTASI

##### A. Implementasi Sistem

Implementasi kebutuhan sistem merupakan langkah yang digunakan untuk mengoperasikan sistem yang dibangun. Spesifikasi *software* dan *hardware* minimum pada saat membangun dan pada saat menggunakan yang dibutuhkan dalam perancangan aplikasi sistem pakar ini yaitu sebagai berikut :

1. Kebutuhan sistem pembangun aplikasi
  - a. Perangkat keras (*hardware*)
    - 1) *Processor* : Intel® Celeron®
    - 2) *Memory RAM* : 2 GB
    - 3) *Harddisk* : 320 GB
  - b. Perangkat lunak (*software*)
    - 1) Sistem operasi : *Windows 7*
    - 2) Aplikasi : *Eclipse Juno*
    - 3) *Database* : *SQLite*
2. Kebutuhan sistem pengguna aplikasi
  - a. Perangkat keras (*hardware*)
    - 1) *Processor* : *Single Core*
    - 2) *Memory RAM* : 512 MB
    - 3) *Memory Internal* : 2 GB
  - b. Perangkat lunak (*software*)
3. Sistem operasi : *Android 2.3.3 (Gingerbread)*

Tampilan program terdiri dari beberapa *printscreen* aplikasi dari tampilan *input*, *output* dan proses.

Tampilan *input* dari aplikasi sistem pakar yang dirancang adalah sebagai berikut:

*Form* Menu utama adalah sebagai *user interface* yang muncul pada saat aplikasi dijalankan.



Gambar 1. Tampilan *Form* Menu Utama

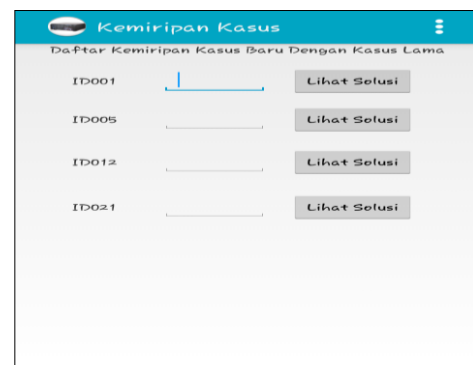
*Form* konsultasi digunakan untuk melakukan konsultasi mendeteksi kerusakan *Amplifier* dengan menginputkan gejala-gejala kerusakan yang terjadi.



Gambar 2. Tampilan *Form* Konsultasi

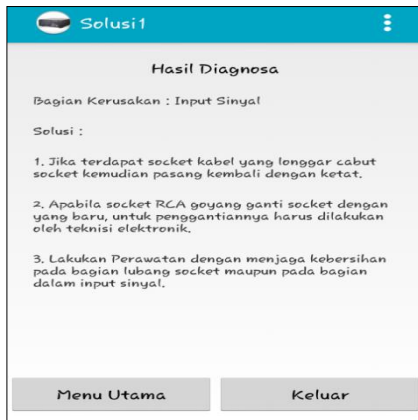
Tampilan *output* aplikasi sistem pakar yang dirancang dengan menggunakan metode *Case Based Reasoning* adalah sebagai berikut:

*Form* daftar kemiripan kasus berisi nilai kemiripan antara kasus baru dengan kasus lama, *user* akan memilih nilai kemiripan paling tinggi secara manual untuk selanjutnya akan tampil *form* hasil diagnosa.



Gambar 3. Tampilan *Form* Kemiripan Kasus

*Form* hasil diagnosa adalah untuk menampilkan hasil diagnosa dari konsultasi mendeteksi kerusakan *Amplifier*.



Gambar 4. Tampilan *Form* Hasil Diagnosa

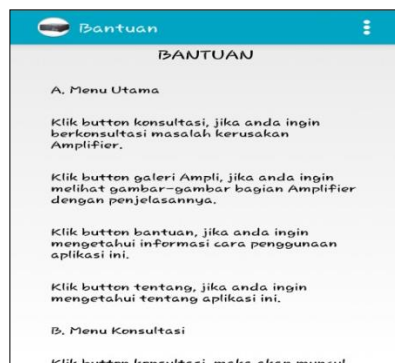
Tampilan menu aplikasi berikut ini merupakan menu yang dapat digunakan *user* agar bisa memahami tentang aplikasi sistem pakar yang dirancang.

*Form* Menu Galeri dapat digunakan untuk mengetahui gambar bagian-bagian *Amplifier* beserta dengan penjelasannya.



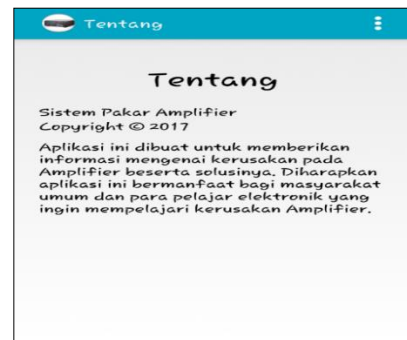
Gambar 5. Tampilan *Form* Menu Galeri

*Form* Menu bantuan berisi mengenai informasi cara penggunaan aplikasi sistem pakar ini



Gambar 6. Tampilan *Form* Menu Bantuan

*Form* Menu tentang berisi mengenai informasi aplikasi ini.



Gambar 7. Tampilan *Form* Menu Tentang

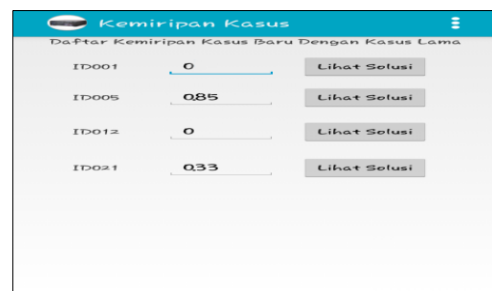
Hasil pengujian program menampilkan hasil output dari sebuah *input* data pada aplikasi yang sudah selesai.

1. Pengujian program pada kasus pertama
  - a. *User* memilih beberapa gejala kerusakan yang terjadi pada *Amplifier*



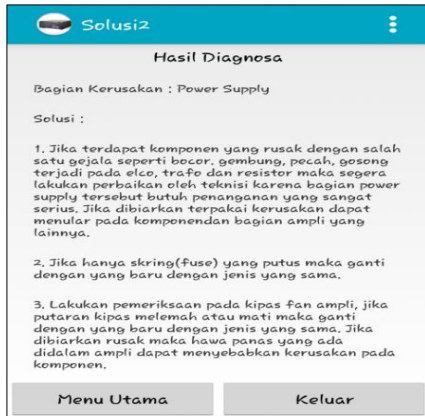
Gambar 8. *Form* Diagnosa Kasus Pertama

- b. Setelah *user* menekan tombol proses maka sistem akan menampilkan hasil kemiripan antara kasus baru dengan kasus lama yang tersimpan di dalam basis data. Setelah itu *user* memilih secara manual nilai kemiripan yang paling tinggi dengan menekan tombol *button* lihat solusi.



Gambar 4.9  
*Form* Daftar Nilai Kemiripan Kasus Pertama

- c. Setelah *user* menekan tombol *button* lihat solusi maka akan tampil *form* hasil diagnosa yang berisi bagian kerusakan dan solusinya.

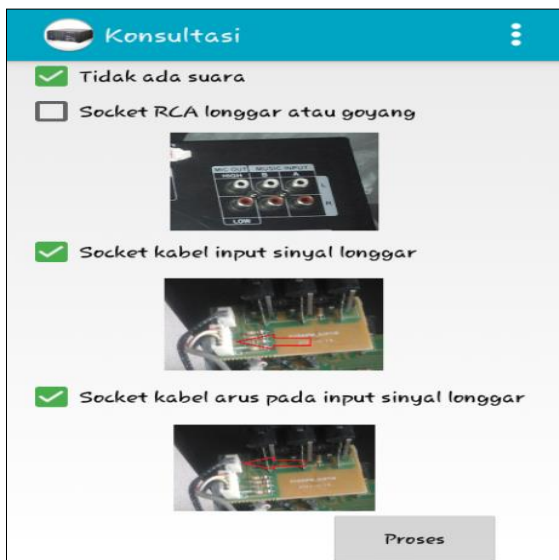


Gambar 4.10

Form Hasil Diagnosa Kasus Pertama

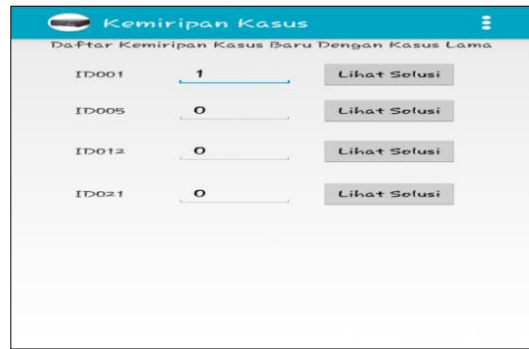
Pada pengujian kasus pertama dapat dihasilkan nilai kemiripan paling tinggi dengan kasus lama ID005 yaitu bernilai 0.85 dengan tingkat persentasi kemiripan tinggi. Bagian kerusakan yang dialami pada kasus pertama adalah *power supply*.

2. Pengujian program pada kasus baru kedua  
 a. *User* memilih beberapa gejala kerusakan yang terjadi pada *Amplifier*



Gambar 11. Form Diagnosa Kasus Kedua

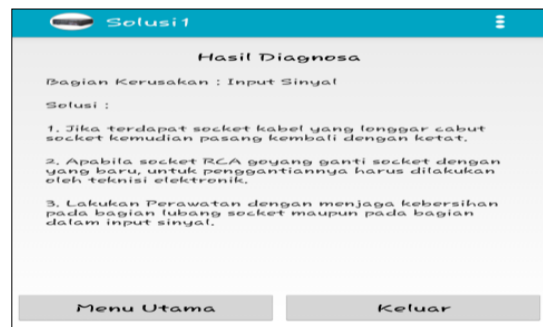
- b. Setelah *user* menekan tombol proses maka sistem akan menampilkan hasil kemiripan antara kasus baru dengan kasus lama yang tersimpan di dalam basis data. Setelah itu *user* memilih secara manual nilai kemiripan yang paling tinggi dengan menekan tombol *button* lihat solusi.



Gambar 12.

Form Daftar Niai Kemiripan Kasus Kedua

- c. Setelah *user* menekan tombol *button* lihat solusi maka akan tampil *form* hasil diagnosa yang berisi bagian kerusakan dan solusinya.



Gambar 13. Form Hasil Diagnosa Kasus Kedua

Pada pengujian kasus baru kedua dapat dihasilkan nilai kemiripan paling tinggi dengan kasus lama ID001 yaitu bernilai 1 dengan tingkat persentasi kemiripan tinggi. Bagian kerusakan yang dialami pada kasus baru kedua adalah *input sinyal*.

## V. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah dilakukan, maka diambil kesimpulan yaitu :

1. Kerusakan *Amplifier* dapat dideteksi dengan menggunakan sistem pakar dimana gejala/kerusakan di representasikan ke dalam *rule*.
2. Metode Case Based Reasoning dapat diterapkan dalam mendeteksi kerusakan *Amplifier* dengan cara menghitung kemiripan gejala pada kasus baru dengan kasus lama satu persatu.
3. Dengan adanya aplikasi sistempakarini, pengguna dapat memperoleh pengetahuan tentang beberapa jenis kerusakan dan tips penanganan pada kerusakan tersebut.

## REFERENCES

- [1] A.S. Aribowo, "Pengembangan Sistem Cerdas Menggunakan Penalaran Berbasis Kasus (*Case Based Reasoning*) untuk

- Diagnosa Penyakit Akibat Virus Eksantema”, UPN Veteran Yogyakarta, Vol. VII, pp.01, 2010.
- [2] T. Rismawan dan S. Hartati, “*Case Based Reasoning* untuk Diagnosa Penyakit THT (Telinga Hidung dan Tenggorokan)”, IJCCS, Vol. VI, pp.01, 2012.
- [3] R. Rosnelly, **Sistem Pakar Konsep dan Teori**, Yogyakarta : C.V Andi Offset. 2012.
- [4] T.E. Putri, D. Andreswari, R. Efendi, “Implementasi Metode CBR (*Case Based Reasoning*) Dalam Pemilihan Pestisida Terhadap Hama Padi Sawah Menggunakan Algoritma *K-Nearest Neighbor* (KNN) (Studi Kasus Kabupaten Seluma)”, Jurnal Rekursif, Vol. IV, pp.82-83, 2016.
- [5] A.A. Suryanto, I. Rosyidi, M. Ulum, A. Wendra, “Penerapan *Case Based Reasoning* (CBR) untuk Mendiagnosa Jenis Pecandu Narkoba”, Seminar Nasional Ilmu Komputer, pp.316, 2016.
- [6] Drs. Buntarto, MPd, **Teknik Audio Video Mobil**, Yogyakarta : PUSTAKABARUPRESS. 2014.
- [7] Purnama, (2012, April, 24). *Power Amplifier* [Online]. Available : <http://elektronika-dasar.web.id/power-amplifier/>.
- [8] A.R. Pratama, (2016, Agust, 10). Belajar UML-Sequence Diagram [Online]. Available : <https://www.codepolitan.com/belajar-uml-sequence-diagram-57fdb1a5ba777-17044>.
- [9] Z.P. Juhara, **Panduan Lengkap Pemograman Android**, Yogyakarta : C.V Andi Offset. 2016.