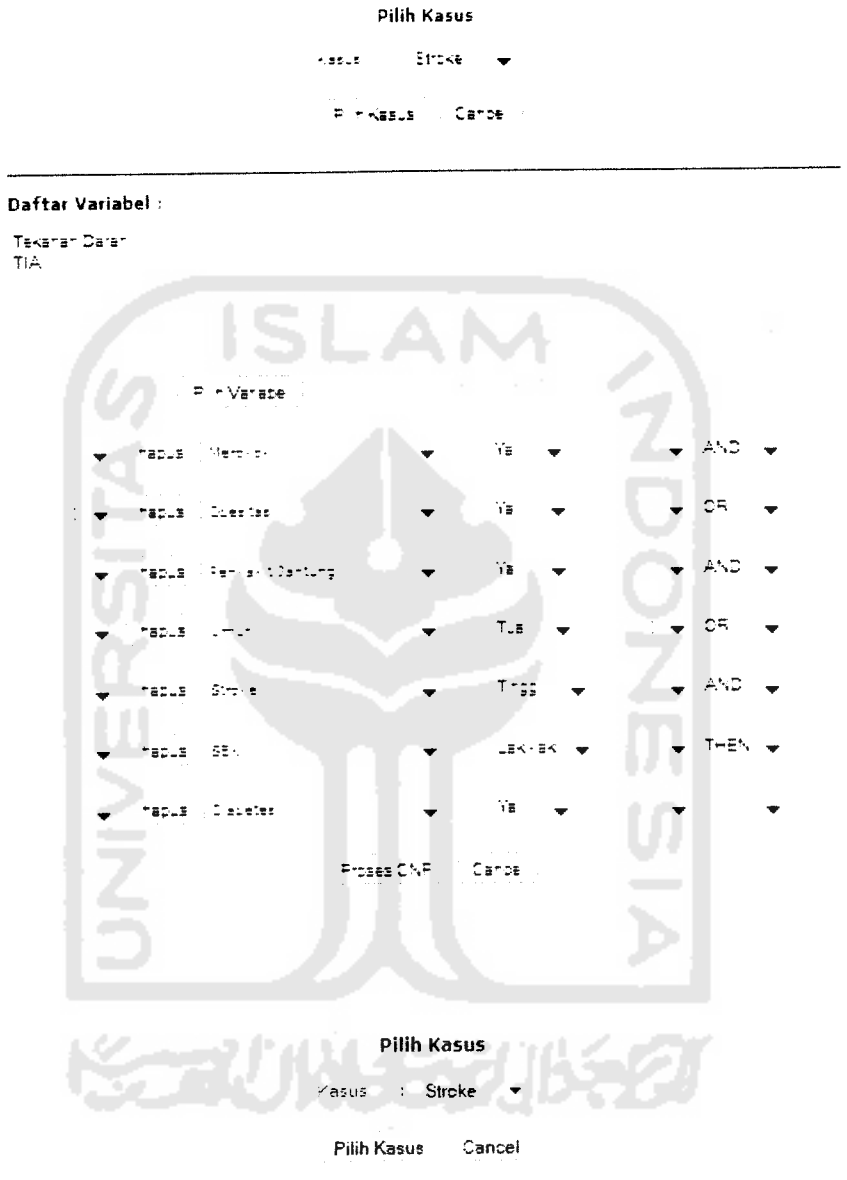
# BAB V
# PENGUJIAN

Dalam bab ini akan dijelaskan tentang pengujian Aplikasi Representasi Basis Pengetahuan Berbasis Aturan yang telah dibangun. Dengan pengujian ini diharapkan tingkat kesalahan baik dalam proses pengolahan data maupun dari sistem itu sendiri menjadi sangat kecil.

## 5.1 Pengujian pada Proses Konfigurasi Basis Pengetahuan

Halaman basis pengetahuan merupakan halaman untuk memproses peryataan-peryataan yang dimasukkan oleh *user* menjadi aturan-aturan yang lebih sederhana. *User* memilih terlebih dahulu kasus yang ingin dibuat pernyataannya, setelah *user* memilih kasus akan tampil daftar variabel-variabel yang berhubungan dengan kasus tersebut. Untuk membentuk variabel-variabel menjadi peryataan *user* memilih variabel-variabelnya, setelah variabel dipilih, *user* menentukan tanda kurung buka, tanda kurung tutup, negasi, nilai variabel dan operator yang dipilih pada tiap variabel. Setelah pernyataan terbentuk dengan benar sistem akan memproses peryataan tersebut menjadi aturan-aturan yang lebih sederhana. Aturan-aturan ini dapat dilihat pada 3 tabel yaitu tabel aturan, tabel blok aturan dan tabel kesimpulan Implementasi proses basis pengertahuan dapat dilihat pada gambar 5.1.




Pernyataan :

IF Merokok Ya AND ( Obesitas Ya OR Penyakit Jantung Ya AND Umur Tua ) OR Stroke Tinggi AND SEX Laki-laki THEN Diabetes Ya

Aturan Yang Terbentuk :

IF Merokok Ya AND Obesitas Ya THEN Diabetes Ya

IF Stroke Tinggi AND SEX Laki-laki THEN Diabetes Ya

IF Merokok Ya AND Penyakit Jantung Ya AND Umur Tua THEN Diabetes Ya

**Gambar 5.1** Gambar Proses Penyerdehanaan Aturan

- IF Merokok Ya AND Penyakit Jantung Ya AND Umur Tua Ya THEN Diabetes Ya
- IF Stroke Tinggi AND SEX Laki-Laki THEN Diabetes Ya

Sehingga dari pernyataan :

- IF Merokok Ya AND ( Obesitas Ya OR Penyakit Jantung Ya AND Umur Tua ) OR Stroke Tinggi AND SEX Laki-Laki THEN Diabetes Ya

Dapat disederhanakan menjadi aturan :

- IF Merokok Ya AND Obesitas Ya THEN Diabetes Ya
- IF Stroke Tinggi AND SEX Laki-Laki THEN Diabetes Ya
- IF Merokok Ya AND Penyakit Jantung Ya AND Umur Tua Ya THEN Diabetes Ya

Pseudocode dari proses penyerdehanaan proses pernyataan menjadi aturan sebagai berikut :

```
stringOr ← "OR";
stringAnd ←"AND";
stringAnswer ;
debugCount ←0;

Fungsi orOperation(varA, varB) {
      if(isAbleToParse (varA) = true){
            parsingString(varA);
      }
      Else{
            stringAnswer[] ← varA;
      }

      if(isAbleToParse (varB) = true){
            parsingString(varB);
      }
      Else{
            stringAnswer[] ← varB;
      }
}
```

```
Fungsi andOperation(varA, varB) {
      stringAns ← null;
      stringAnswer[] ← varA
      stringAnd ← varB;
}

Fungsi solve(string) {
      if (ada kurung buka, pada string){
            newVar ← parsingBracket(string);
      } else {
            newVar[] ← string;
      }

      for (i = 0 to  jumlah newVar){
            if (ada kurung buka, pada newVar[i]) {
                  solve(newVar[i]);
            } else {
                  parsingString(newVar[i]);
            }
      End for
      }
}


Fungsi parsingBracket(string) {
      newVar;
      arrStrSplit;
      count ← 0;
      stringTemp ←null;
      skip ← false;

      for (i = 0 to  jumlah string){
            char = pisahkan string dari huruf ke 1;
            if (char = kosong dan skip = false) {
                  arrStrSplit[] ← stringTemp;
                  stringTemp ← kosong;
            } else
                  if char = kurung tutup dan skip = false) {
                        arrStrSplit[] ← stringTemp;
                        stringTemp ← kosong;
                        skip ←true;
                  } else {
                        stringTemp ← char;
                  }
            }

            arrStrSplit[] ← stringTemp;
            bracket ←checkBracket(arrStrSplit[0],
                  arrStrSplit[1]);
```

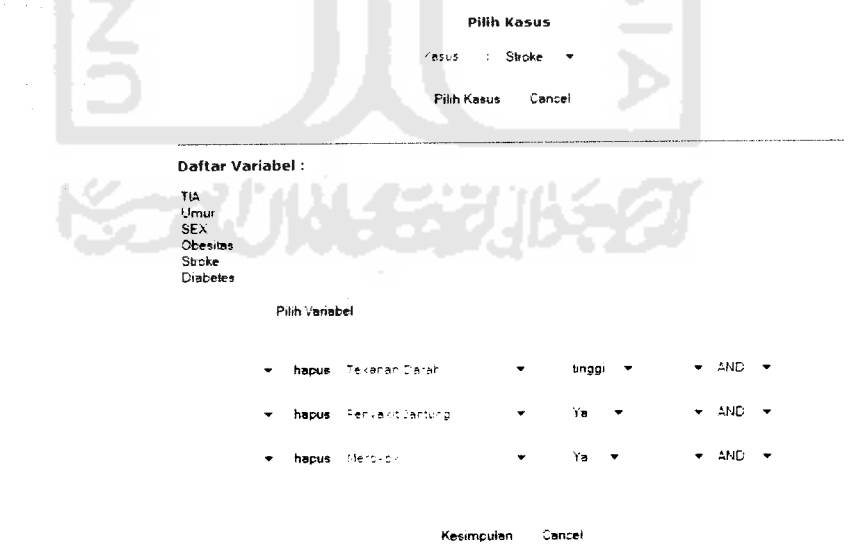
```
            if (bracket = 0) {
                  nonBracket ← 1;
            } else {
                  nonBracket ← 0;
            }

            if (cek jika dalam arrStrSplit[bracket] masih
                  terdapat Or) {
                  var ← arrStrSplit[bracket];
                  if (cek jika jumlah nonBracket < bracket)
                  {
                        newVar[] ← arrStrSplit[nonBracket] .
                           " " . var[0] . " " .
                           arrStrSplit[2];
                        newVar[] ←$arrStrSplit[nonBracket] .
                           " " . var[1] . " " .
                           arrStrSplit[2];
                  } else {
                        newVar[] ← var[0] . " " .
                           arrStrSplit[nonBracket] . " " .
                           arrStrSplit[2];
                        newVar[] ← var[1] . " " .
                           arrStrSplit[nonBracket] . " " .
                           arrStrSplit[2];
                  }
            } else {
                  if (cek jika jumlah nonBracket < bracket)
                  {
                        newVar[] ← arrStrSplit[nonBracket] .
                           " " . arrStrSplit[bracket] . " "
                           . arrStrSplit[2];
                  } else {
                        newVar[] ← arrStrSplit[bracket] . "
                           " . arrStrSplit[nonBracket] . " "
                           . arrStrSplit[2];
                  }
            }
      return newVar;
      End For
}

Fungsi checkBracket(varA, varB) {
      bracket ← -1;
      isEmpty ← false;
      varBSplit ← varB;

      if (cek jika varBSplit[0] = Or atau  varBSplit[0]  =
          And ) {
            bracket ← 0;
      } else {
            bracket ← 1;
      }
      return bracket;
}
```

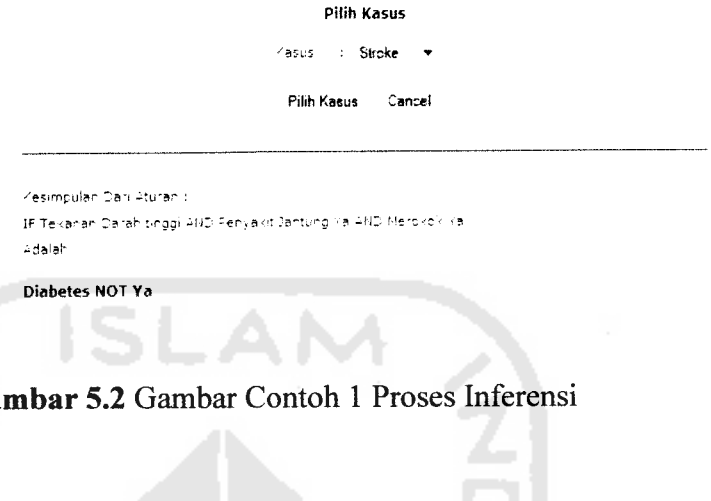
```
Fungsi parsingString(string) {
      if (cek jika dalam string masih terdapat Or) {
            var ← stringOr;
            orOperation(var[0], var[1]);
      } else
            if (cek jika dalam string masih terdapat And) {
                  var ← string;
                  andOperation(var[0], var[1]);
            } else {
                  stringAnswer[] ← string;
            }
      }
}

Fungsi isAbleToParse(string) {
      able ← false;
      if (cek jika dalam string masih terdapat Or) {
            able ← true;
      }
      if (cek jika dalam string masih terdapat And) {
            able ← false;
      }
      return able;
}

Fungsi removeSpace(string) {
      space ← false;
      stringSebelumnya ← kosong;
      for (i = 0 to  jumlah string){
            char ← string;

            if (cek jika char = null dan string Sebelumnya =
                null){
            } else {
                  Temp ← char;
            }
            stringSebelumnya  ←char;
      }
      return temp;
      End For
}

Fungsi showAnswer() {
      stringAnswer ←  stringAnswer;

      for (i = stringAnswer  to  no = val){
            temp[] ← removeSpace(val);
      }
```

```
        stringAnswer ← temp;
        jawab ← stringAnswer;
        stringAnswer ← null;
        return jawab;
}
```

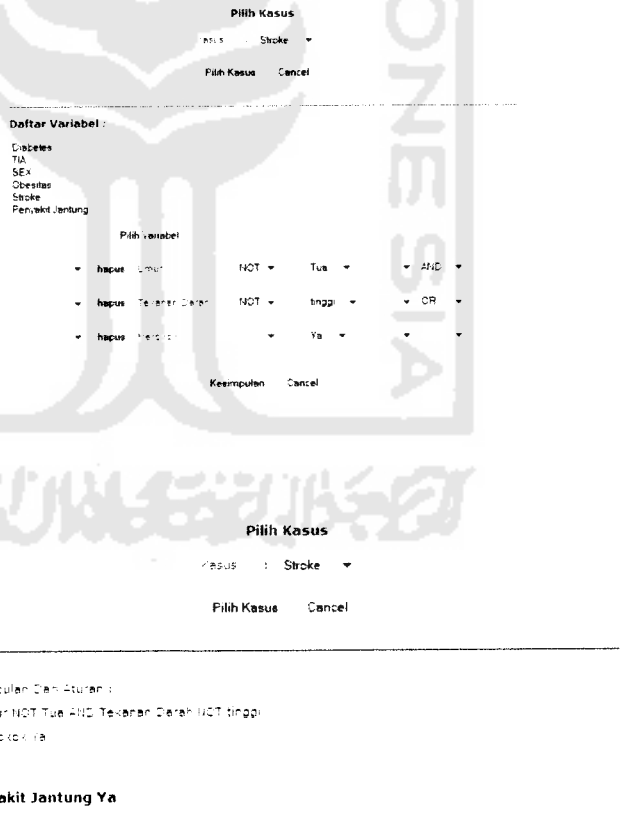
## 5.2 Proses Inferensi

Pada proses inferensi aturan, *user* akan menguji aturan-aturan yang terdapat dalam basis pengetahuan yang diperoleh pada proses sebelumnya. Pada proses inferensi ini *user* memasukkan pernyataan seperti pada proses sebelumnya tetapi tanpa menyertakan kesimpulan dari pernyataan tersebut ,kemudian pernyataan tersebut akan dibentuk aturan-aturan yang lebih sederhana melalui bentuk *CNF*. Dari aturan tersebut akan dicocokan dengan aturan-aturan yang terdapat pada basis pengetahuan. Hasil akhir dari proses inferensi ini adalah kesimpulan dari tiap aturan. Implemetasi proses inferensi dapat dilihat pada gambar 5.2 dan 5.3






**Gambar 5.2** Gambar Contoh 1 Proses Inferensi




**Gambar 5.3** Gambar Contoh 2 Proses Inferensi

```
for (i = 0 to count($_POST[id_variabel])) {
      stringVar2  ← $_POST[not][i] $_POST[nilai_variabel][i];
}

lihat_id_aturan ← mysql_query("select DISTINCT(id_aturan)
                                from blok_aturan");

while (id_aturan = mysql_fetch_array(lihat_id_aturan)) {
      lihat_blok_aturan ← mysql_query("
            select negasi, id_nilai_variabel from blok_aturan
            where id_aturan = 'id_aturan[0]'
      ");

      while (blok_aturan =
            mysql_fetch_assoc(lihat_blok_aturan)) {
                  temp_aturan[] = blok_aturan;
      }

      data_blok_aturan[] = array("id_aturan" => id_aturan[0],
                                 "isi" => temp_aturan);
      temp_aturan = null;
}

foreach (jawaban2 as no => val) {
      if (is_array(val)) {
            foreach (val as n => v) {
                  if (count(v) == 2) {
                        variable = v[1];
                        not = 0;
                  } else {
                        variable = v[0];
                        not = 1;
                  }

                  temp_aturan_uji[] =
                        array("negasi" => not,
                              "id_nilai_variabel" =>
                              variable);
            }

      } else {
            if (count(v) == 2) {
                  variable = v[1];
                  not = 0;
            } else {
                  variable = v[0];
                  not = 1;
                  }
            temp_aturan_uji[] =
                  array("negasi" => not, "id_nilai_variabel"
                        => variable);
      }
```

```
        data_blok_aturan_uji[] =
        array("isi" => temp_aturan_uji);
        temp_aturan_uji = null;
}


// pengecekan dari data base

foreach (data_blok_aturan_uji as no => blok_aturan_uji) {
        foreach (data_blok_aturan as id => blok_aturan) {
                if (blok_aturan_uji[isi] == blok_aturan[isi]) {
                        array_kes[] ← blok_aturan["id_aturan"];
                } else {
                        array_kes[] ← "-1";
                }
        }
}


// tampil jawaban

foreach (array_kes as no => val) {
        id_aturan ← val;

        if (id_aturan ← "-1") {
                string_kesimpulan ← " kesimpulan tidak ada ";
        } else {
                lihat_kesimpulan = mysql_query("
                        select  * from kesimpulan, aturan
                        where
                                kesimpulan.id_kesimpulan =
                                        aturan.id_kesimpulan and
                                aturan.id_aturan = 'id_aturan'
                ");


                kesimpulan =
                  mysql_fetch_array(lihat_kesimpulan);
                negasi_kesimpulan ← "0";
                negasi_kesimpulan ←kesimpulan[negasi];
                id_variabel_kesimpulan ←
                        kesimpulan[id_nilai_variabel];
                string_kesimpulan ← "";

                if (negasi_kesimpulan == "0") {
                        string_kesimpulan ← "NOT";
                } else {
                        string_kesimpulan ← "";
                }
```

```
            lihat_id_nama_variabel = mysql_query("
                  select * from nilai_variabel, variabel,
                         kasus
                  where
                        nilai_variabel.id_variabel =
                              variabel.id_variabel and
                        variabel.id_kasus = kasus.id_kasus
                              and
                        kasus.id_kasus = $ksss and
                        nilai_variabel.id_nilai_variabel =
                              '$id_variabel_kesimpulan'
            ");


            variabel =
                  mysql_fetch_array($lihat_id_nama_variabel);

            string_kesimpulan =  variabel[nama_variabel]
                  variabel[nama_nilai_variabel];
      }
}
```

**Tabel 5.1** Tabel Hasil Pengujian

| No | Pernyataan | Aturan |
|---|---|---|
| 1 | If A And B And C Then D | • If A And B And C Then D |
| 2 | If A And B Or C Then D | • If A And B Then D<br>• If C Then D |
| 3 | If A Or B then C | • If A Then C<br>• If B Then C |
| 4 | If A and B and C or D and E and F or G and H then K | • If A and B and C then K<br>• If D and E and F then K<br>• If G and H then K |
| 5 | IF A and ( B or C ) then D | • If A and B then D<br>• If B and C then D |
| 6 | IF (A and B) or ( C and D and E ) then F | • If A and B then F<br>• If C and D and E then F |

| | | |
|---|---|---|
| 7 | If A and B and ( C or D ) and E and ( F or G ) and H then I | • If A and B and C and E and F and H then I<br>• If A and B and C and E and G and H then I<br>• If A and B and D and E and F and H then I<br>• If A and B and D and E and G and H then I |
| 8 | If A and ( C or D ) and E or H then B | • If A and C and E then B<br>• If A and D and E then B<br>• If H the B |
| 9 | If A Or ( B And C Or D) And E Or ( F And G Or H ) And I Then J | • If A Then J<br>• If B And C And E Then J<br>• If F And G And I Then J<br>• If H And I Then J<br>• If D And E Then J |
| 10 | If A And B And ( C Or D And E ) Or G And F Or H And I Then J | • If A And B And C Then J<br>• If G And F Then J<br>• If H And I Then J<br>• If A And B And D And E Then J |