BAB 4

IMPLEMENTASI DAN PENGUJIAN

Bab ini merupakan tahap menterjemahankan kebutuhan pembangunan aplikasi ke dalam representasi perangkat lunak sesuai dengan hasil analisis yang telah dilakukan. Setelah implementasi maka dilakukan pengujian sistem yang baru dimana akan dilihat kekurangan-kekurangan pada aplikasi untuk selanjutnya dievaluasi.

4.1 Implementasi Sistem

Setelah sistem dianalisis dan didesain secara rinci, maka akan menuju tahap implementasi. Implementasi sistem merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehingga pengguna dapat memberi masukan kepada pengembangan sistem.

Setelah melakukan tahap perancangan sistem dan implementasi perangkat lunak, maka tindakan selanjutnya yang dilakukan adalah penerapan hasil perangkat lunak tersebut.

4.1.1 Implementasi Perangkat Keras

Sedangkan spesifikasi minimum untuk implementasi aplikasi perangkat lunak yaitu :

1. *Processor* dengan kecepatan 1,6 GHz.
2. Kapasitas *harddisk* minimal 20 GB.
3. *Memory* 256 MB.
4. Kartu grafis 64 MB.
5. Monitor beresolusi 1024x600 piksel.
6. *Keyboard* dan *mouse.*

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang dipergunakan untuk membangun aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian pada permasalahan ruang keadaan (*state and space*) yaitu :

1. *Windows 7 Home Premium* sebagai sistem operasi.
2. *Microsoft Visual Basic 6.0* sebagai aplikasi pembangun.
3. *Microsoft Access 2007* sebagai DBMS.

4.1.3 Implementasi Basis Data

Pembuatan database dilakukan dengan menggunakan aplikasi DBMS *Microsoft Access 2007.* Implementasi database dalam bahasa SQL seperti pada Tabel 4.1

Tabel 4.1. Implementasi database

No	Keterangan	Uraian
1	Nama Tabel	Problem
	Nama File	StateAndSpace.mdb
	Implementasi	CREATE TABEL `Problem` (`ProblemID` Text(10) NOT NULL DEFAULT '', `NamaProblem` Text(100) NOT NULL DEFAULT '', PRIMARY KEY (`ProblemID`)) ENGINE= ADODB ROW_FORMAT=DEFAULT
2	Nama Tabel	Variabel
	Nama File	StateAndSpace.mdb
	Implementasi	CREATE TABEL `Variabel` (`ProblemID` Text(10) NOT NULL DEFAULT '', `VariabelID` Text(10) NOT NULL DEFAULT '', `NamaVariabel` Text(50) NOT NULL DEFAULT '', `Driver` Boolean(Yes/No) NOT NULL DEFAULT '', `Foto` LongBinary NOT NULL DEFAULT '', PRIMARY KEY (`VariabelID`), FOREIGN KEY (‘ProblemID’)) ENGINE= ADODB ROW_FORMAT=DEFAULT
3	Nama Tabel	Control
	Nama File	StateAndSpace.mdb
	Implementasi	CREATE TABEL `Control` (`ProblemID` Text(10) NOT NULL DEFAULT '', `Variabel1` Text(50) NOT

		NULL DEFAULT '', `Variabel1` Text(50) NOT NULL DEFAULT '', `AttackID` Text(50) NOT NULL DEFAULT '', `Variabel2` Text(50) NOT NULL DEFAULT '', `Variabel3` Text(50) NOT NULL DEFAULT '', `Variabel4` Text(50) NOT NULL DEFAULT '', `Variabel5` Text(50) NOT NULL DEFAULT '', PRIMARY KEY ('AttackID'), FOREIGN KEY ('ProblemID')) ENGINE= ADODB ROW_FORMAT=DEFAULT

4.1.4 Implementasi Form

Implementasi *form* dilakukan untuk mengetahui setiap halaman program yang dibuat dan pengkodeannya dalam bentuk *file* program. Implementasi *form* dapat dilihat pada tabel 4.2

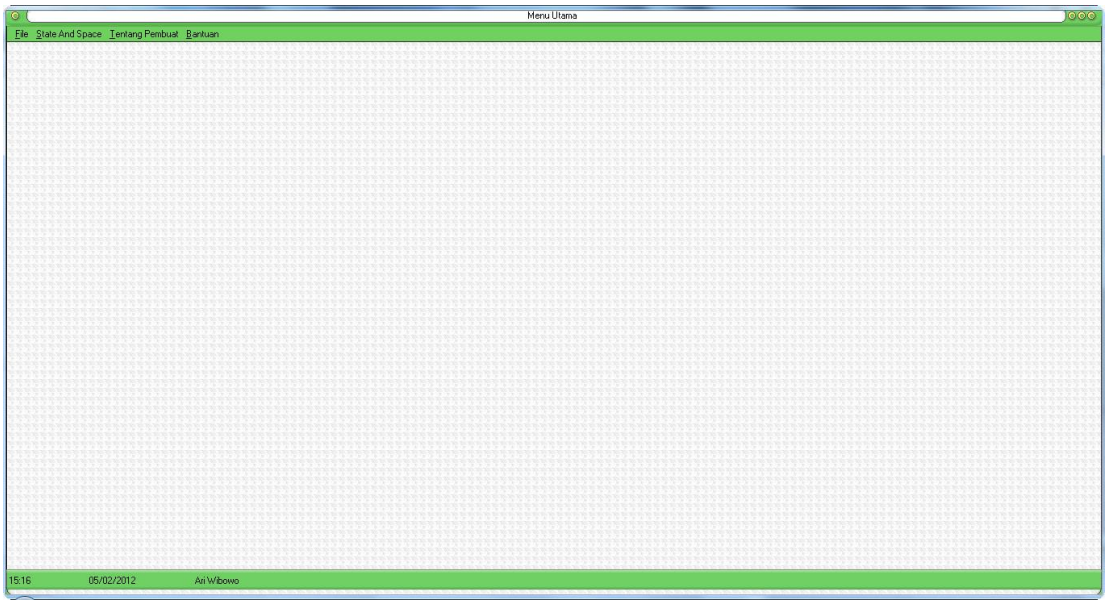
Tabel 4.2. Implementasi Form

Menu	Deskripsi	Nama File
Halaman Utama	Tampilan awal antarmuka dan menu utama	frmUtama.frm
Problem	Tampilan untuk mengolah data *problem* seperti tambah, edit, hapus.	frmProblem.frm
Variabel	Tampilan untuk mengolah variabel seperti tambah, edit, hapus.	frmVariabel.frm

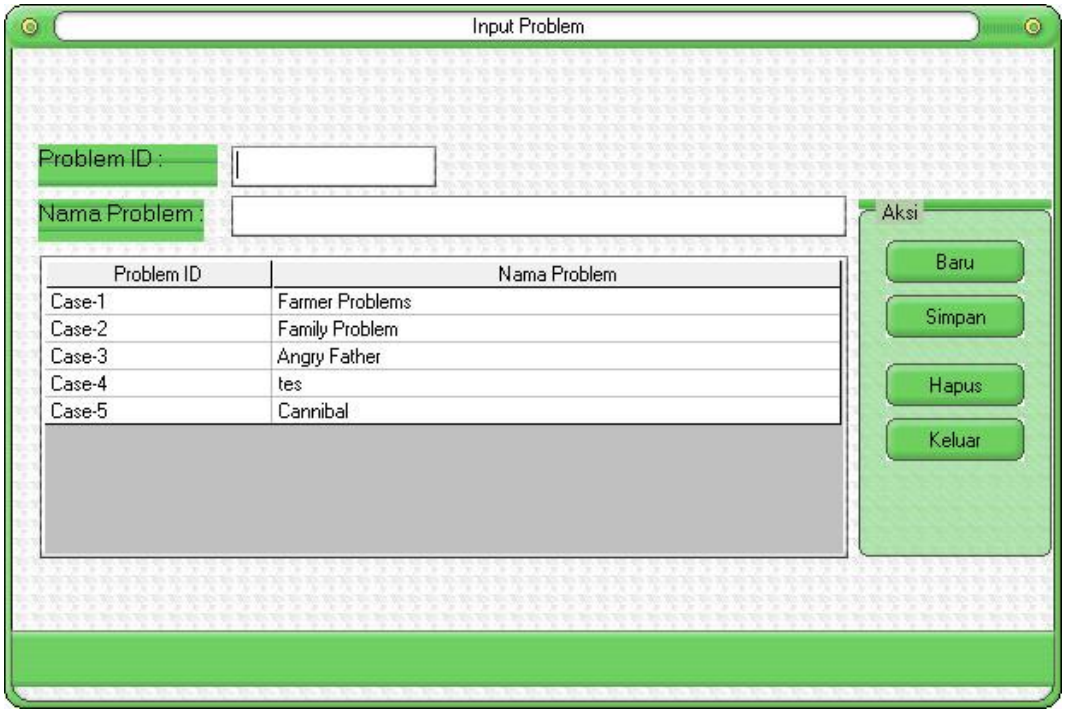
Control	Tampilan untuk menambah strategi control seperti tambah, edit, hapus.	frmRule.frm
Solusi	Tampilan untuk memilih problem yang akan diselesaikan secara manual atau secara otomatis (AI)	frmMain.frm
Solusi Manual	Tampilan untuk melihat dan menjalankan pencarian solusi serta simulasi langkah penyelesaian secara manual	frmSolusiManual.frm
Solusi AI	Tampilan untuk melihat dan menjalankan pencarian solusi serta simulasi langkah penyelesaian secara otomatis.	frmSolusiAI.frm
Materi *State And Space*	Tampilan untuk melihat teori state and space	frmMateriState.frm
Materi *Breadth First Search*	Tampilan untuk melihat teori *Breadth First Search*	frmMateriBFS.frm
Contoh Soal	Tampilan untuk melihat Contoh Soal	frmContoh.frm
Bantuan	Tampilan Untuk melihat bantuan penggunaan aplikasi	frmBantuan.frm
About	Tampilan untuk melihat tentang pembuat aplikasi	frmAbout.frm

4.1.5 Implementasi Antarmuka

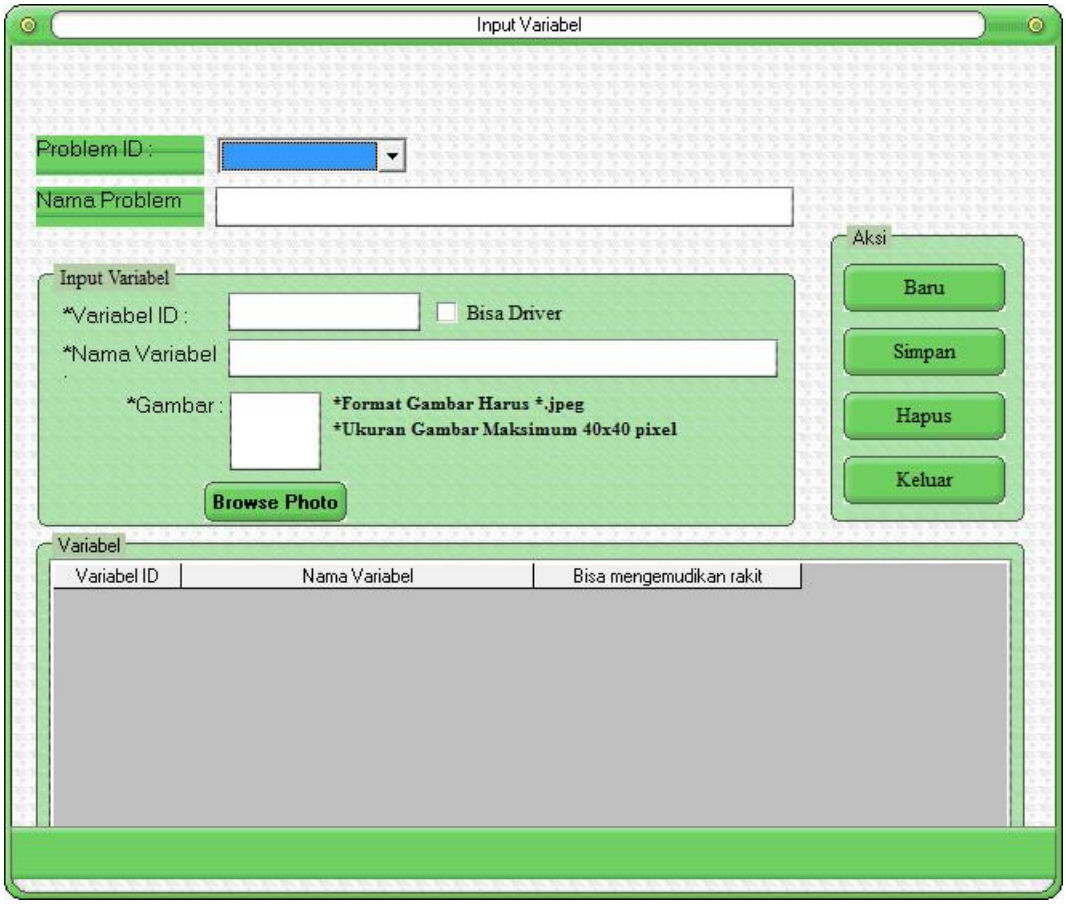
Implementasi antarmuka dilakukan dengan setiap halaman program yang dibuat dan pengkodeannya dalam bentuk file program. Implementasi antarmuka yang dibangun yaitu sebagai berikut :



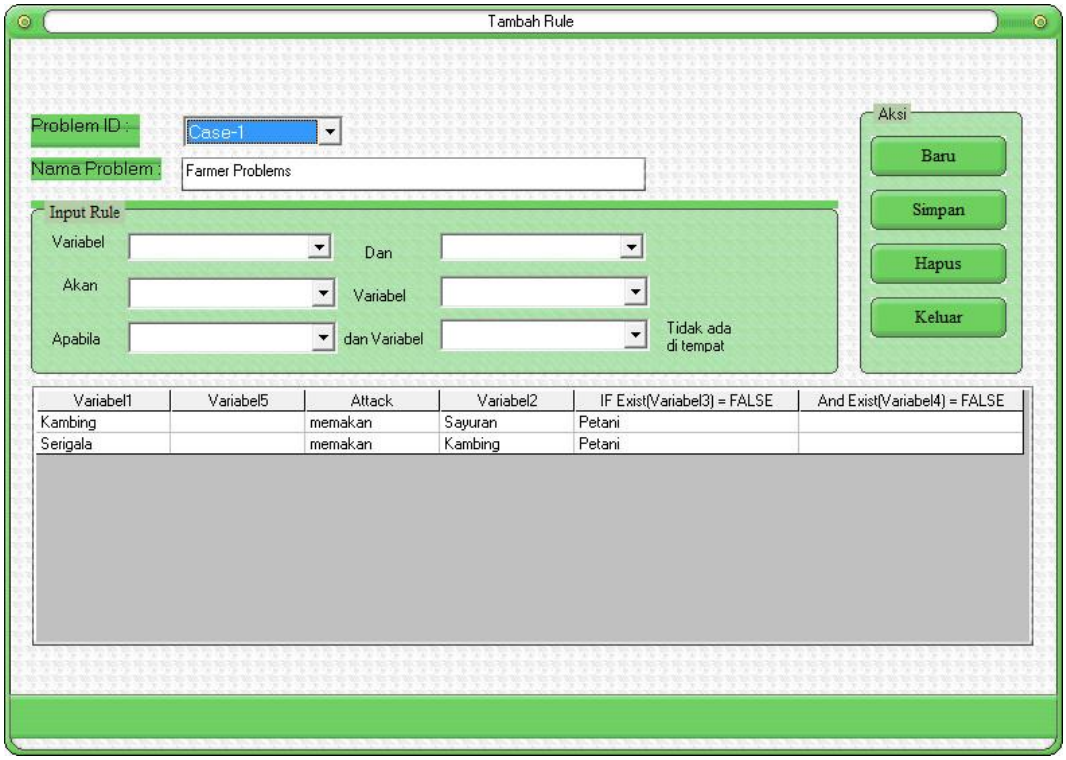
Gambar 4.1. Halaman Utama



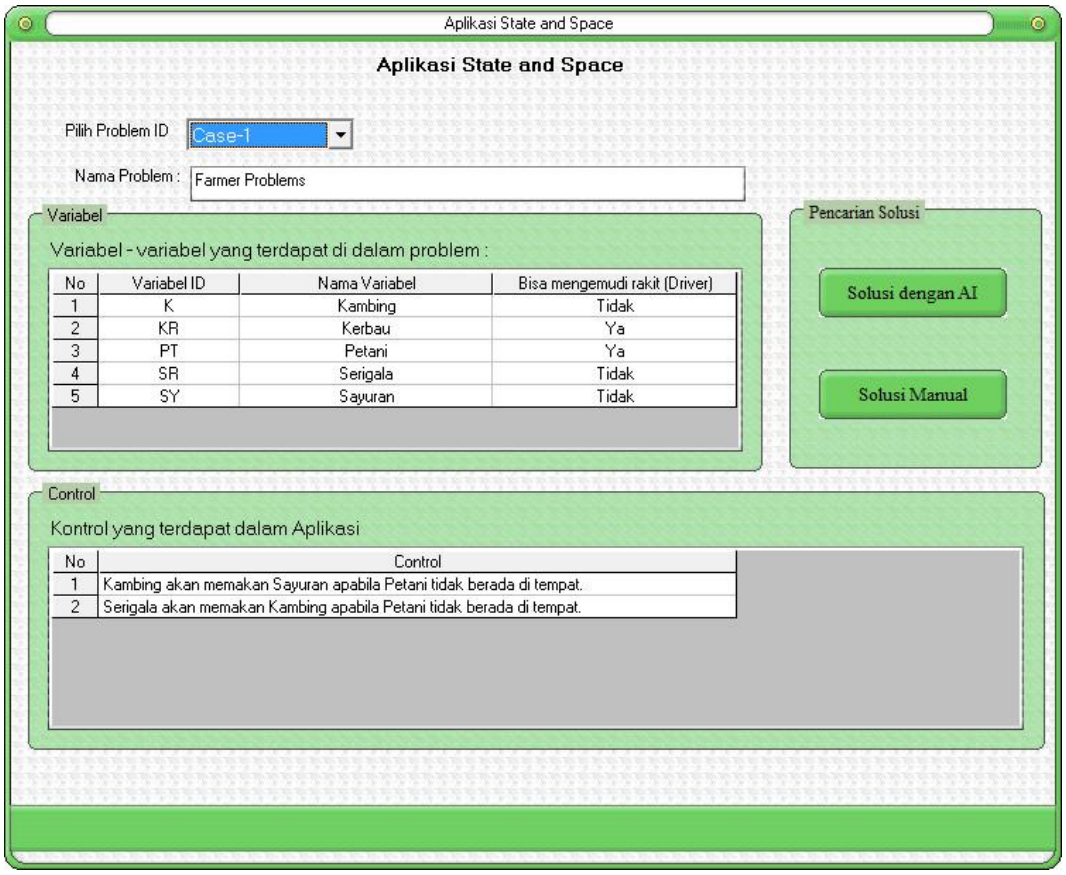
Gambar 4.2. Halaman kelola *Problem*



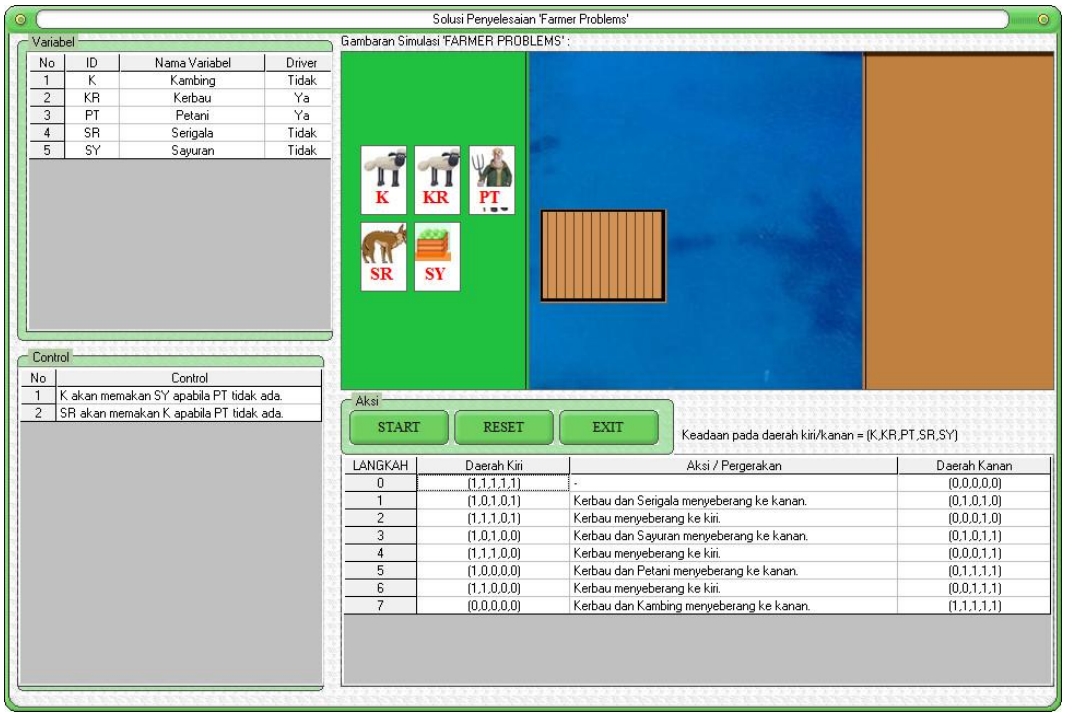
Gambar 4.3. Halaman kelola *variabel*



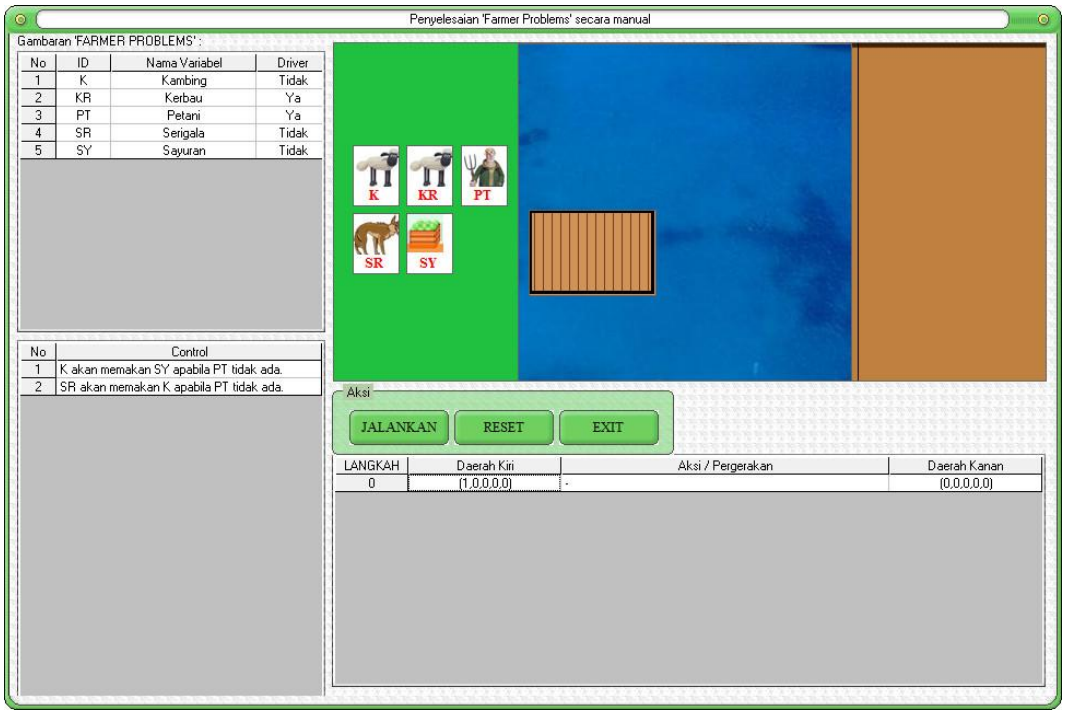
Gambar 4.4. Halaman kelola *rule*



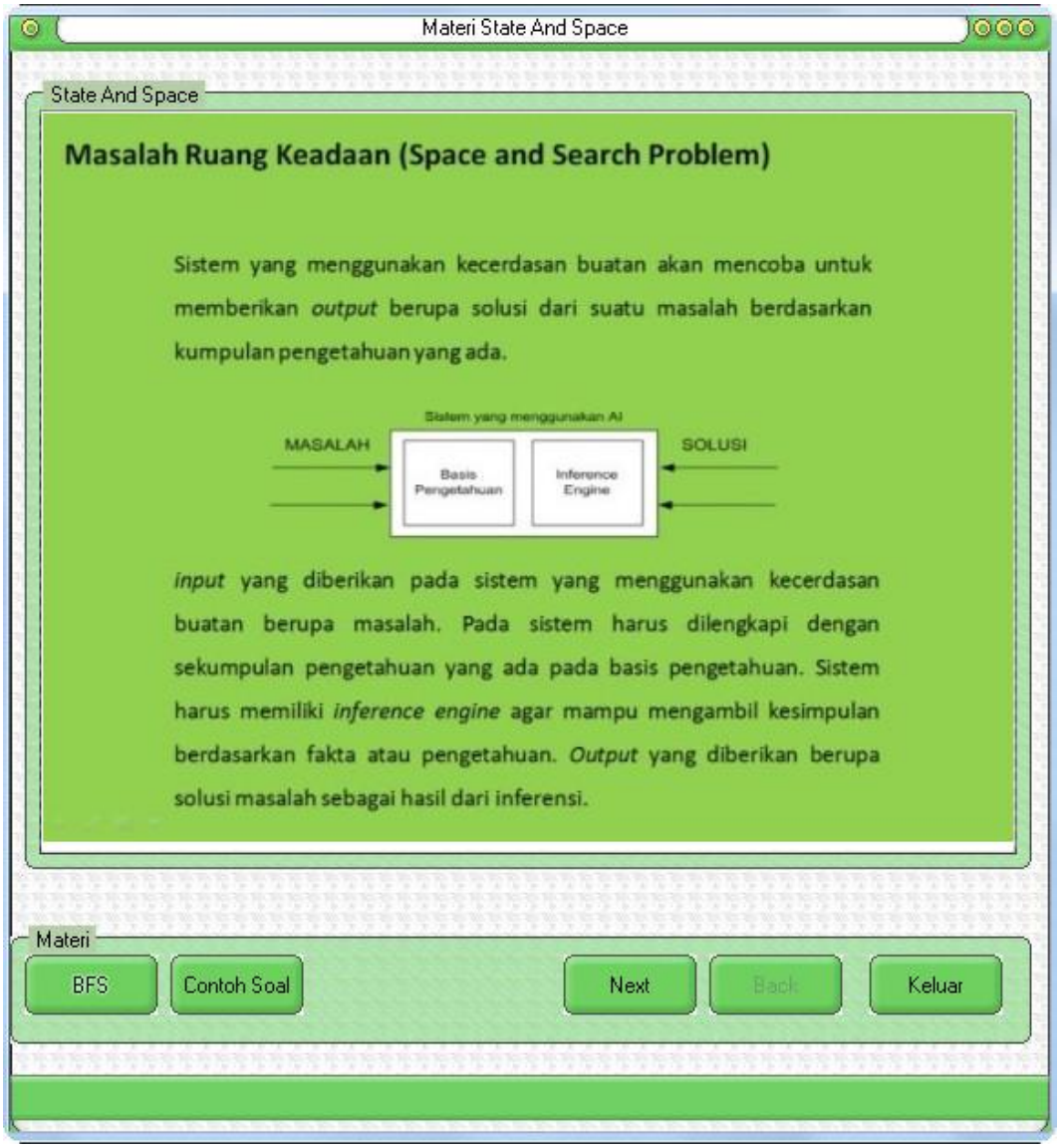
Gambar 4.5. Halaman pilih solusi Aplikasi *State and Space*



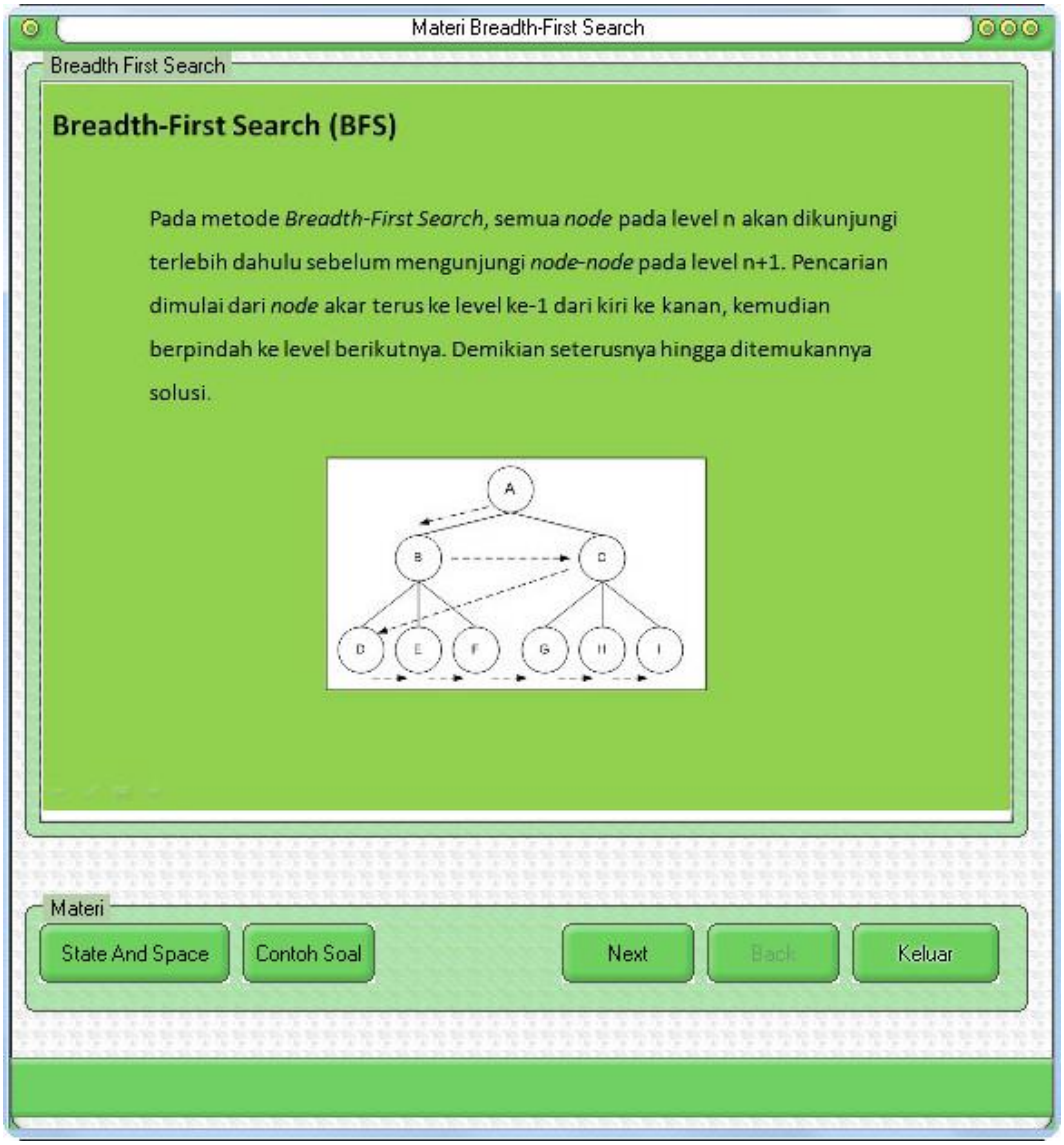
Gambar 4.6. Halaman Solusi AI



Gambar 4.7. Halaman Solusi Manual



Gambar 4.8. Halaman Materi *State and Space*



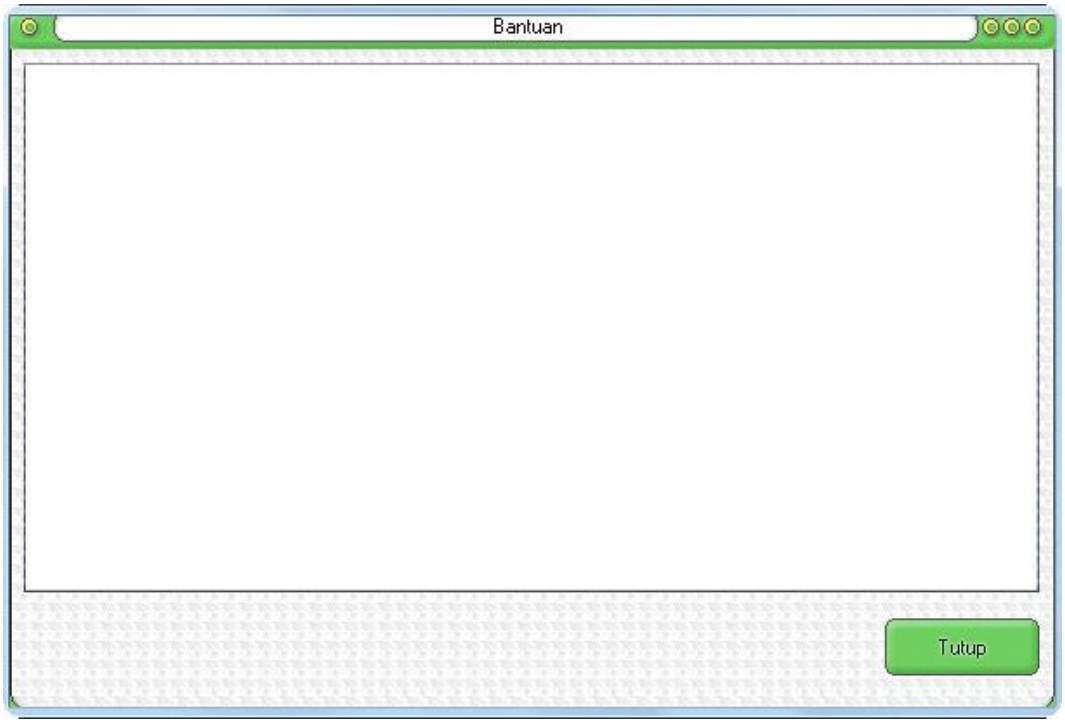
Gambar 4.9. Halaman Materi *Breadth-First Search*



Gambar 4.10. Halaman Contoh Soal



Gambar 4.11. Halaman Tentang Pembuat



Gambar 4.12. Halaman Bantuan

4.1.6 Implementasi Algoritma Pohon Pelacakan

Algoritma penyelesaian masalah *state and space* menggunakan pohon pelacakan untuk mencari solusi dari setiap permasalahan. Akar dari pohon merupakan keadaan awal. Akar pohon kemudian membentuk cabang-cabang baru dengan melakukan semua aksi-aksi yang dapat terjadi pada keadaan awal. Keadaan-keadaan baru yang dihasilkan merupakan cabang baru dari pohon. Proses pengembangan dan pencarian dilanjutkan dengan membentuk keadaan baru (cabang baru) dari cabang-cabang yang sudah ada hingga didapatkan solusi. Proses pembentukan keadaan baru juga melakukan prosedur pengecekan apakah keadaan tersebut sudah pernah dibentuk sebelumnya pada pohon. Apabila sudah ada, maka cabang baru tidak jadi dibentuk. Apabila tidak ada cabang yang dapat dikembangkan lagi dan ternyata belum ditemukan solusi, maka permasalahan yang sedang diselesaikan tidak memiliki solusi. Penelusuran pohon pelacakan dilakukan secara per level, dari level-1 (keadaan awal), kemudian ke level-2 (keadaan yang dapat dicapai oleh level-1) dan seterusnya. Metode penelusuran seperti ini disebut metode pencarian melebar pertama (*Breadth-First Search*).

Algoritma pencarian juga melakukan prosedur pengecekan apakah setiap aksi yang dilakukan pada setiap keadaan melanggar aturan yang telah dibuat sebelumnya. Prosedur ini memastikan bahwa keadaan baru yang akan dibentuk tidak melanggar aturan yang telah ada. Adapun struktur algoritma pencarian solusi adalah sebagai berikut:

1. Panggil *record* aturan (*rule*) dari *database* dan simpan ke dalam variabel *array* ‘Rule’. Seperti yang telah dijelaskan sebelumnya, bentuk aturan adalah variabel1 akan menyerang variabel2 apabila variabel3 dan variabel4 tidak

berada di tempat. Untuk I1 = 1 sampai rs.RecordCount, lakukan prosedur berikut:

a. I2 = (indeks *array* tertinggi dari variabel Rule) + 1.
b. Control(I2).Var1 = rs!Variabel1.
c. Control(I2).Var2 = rs!Variabel2.
d. Control(I2).Var3 = rs!Variabel3.
e. Control(I2).AttID = rs!AttackID.
f. Control(I2).Var4 = rs!Variabel4.
g. rs.MoveNext.

2. Panggil *record* variabel dari *database* dan simpan ke dalam variabel *array* 'Variabel'. Untuk I1 = 1 sampai rs.RecordCount, lakukan prosedur berikut:
   a. I2 = (indeks *array* tertinggi dari variabel Rule) + 1.
   b. Variabel(I2).VarID = rs!VariabelID.
   c. Variabel(I2).NamaVar = rs!NamaVariabel.
   d. Variabel(I2).Driver = rs!Driver.
   e. KAwal = KAwal & IIf(I1 = 1, "", ",") & "1".
   f. sT1 = sT1 & IIf(I1 = 1, "", ",") & "0".
   g. KTujuan = KTujuan & IIf(I1 = 1, "", ",") & "1".
   h. rs.MoveNext.
3. Hasilkan semua kombinasi variabel pada rakit yang mungkin terjadi dan tidak menyalahi aturan. Untuk I1=1 sampai batas atas *array* Variabel,
   a. Jika Variabel(I1).Driver = *True* maka, lanjutkan ke algoritma berikutnya. Jika tidak, maka lanjutkan *looping* berikutnya untuk variabel I1.

b. Hasilkan kombinasi *single* (hanya terdapat satu buah variabel pada rakit sehingga tidak akan pernah menyalahi aturan manapun).
    i. I4 = UBound(Kombinasi) + 1.
    ii. ReDim Preserve Kombinasi(I4).
    iii. Kombinasi(I4).Var1 = Variabel(I1).NamaVar.
    iv. Kombinasi(I4).Var2 = “”.
    v. Tambah ke list. sList = sList & “{“ & Kombinasi(I4).Var1 & “}”.
c. Cari kombinasi *double* (terdapat dua buah variabel pada rakit). Untuk I2 = 1 sampai Ubound(Variabel), lakukan prosedur berikut.
    i. Jika I1 <> I2 dan kombinasi(Variabel(I1).NamaVar dan Variabel(I2).NamaVar) belum pernah ada pada variabel sList, maka lanjutkan ke poin berikutnya. Jika tidak maka lanjutkan *looping* untuk variabel I2.
    ii. Cek apakah kombinasi variabel menyalahi aturan yang telah ada. Set B1 = *False*. Untuk I3 = 1 sampai UBound(Control), lakukan prosedur berikut.
        1) Jika ((Variabel(I1).NamaVar = Control(I3).Var1) dan (Variabel(I2).NamaVar = Control(I3).Var2)) Atau ((Variabel(I2).NamaVar = Control(I3).Var1) dan (Variabel(I1).NamaVar = Control(I3).Var2)) maka set B1 = *True* dan keluar dari looping.
    iii. Jika B1 = False (ini artinya tidak menyalahi aturan), maka hasilkan kombinasi *double*.

1) I4 = UBound(Kombinasi) + 1.
2) ReDim Preserve Kombinasi(I4).
3) Kombinasi(I4).Var1 = Variabel(I1).NamaVar.
4) Kombinasi(I4).Var2 = Variabel(I2).NamaVar.
5) Tambah ke list. sList = sList & "{" & Kombinasi(I4).Var1 & "," & Kombinasi(I4).Var2 & "}"& "{" & Kombinasi(I4).Var2 & "," & Kombinasi(I4).Var1 & "}".

4. Memulai pencarian dari keadaan awal.
   a. ReDim Keadaan(1).
   b. Keadaan(1).DKiri = KAwal.
   c. Keadaan(1).DKanan = sT1.
   d. Keadaan(1).Rakit = "KIRI".
   e. Keadaan(1).IsExpand = *False*.
   f. Keadaan(1).Induk = 0.
   g. Keadaan(1).Aksi = "".
5. Tambahkan list keadaan. sList = "{Kr=" & KAwal & ",Kn = " & sT1 & "R=KIRI}".
6. N = 0.
7. Selama belum semua keadaan dikembangkan / Not IsAllExpand(Keadaan), maka lakukan prosedur berikut:
   a. N = N + 1.
   b. Simpan keadaan pada daerah asal dan keadaan pada daerah seberang pada variabel sT1 dan sT2.

i. Jika Keadaan(N).Rakit = “KIRI” maka sT1 = Keadaan(N).DKiri, sT2 = Keadaan(N).DKanan.

ii. Jika tidak, maka sT1 = Keadaan(N).DKanan, sT2 = Keadaan(N).DKiri.

c. Lakukan semua kombinasi yang mungkin untuk keadaan pada daerah asal (sT1) dan daerah seberang (sT2). Untuk I1 = 1 sampai UBound(Kombinasi), lakukan prosedur berikut:

i. Penggal setiap karakter sT1 dengan pemisah koma dan masukkan ke variabel IsiT1. IsiT1 = Split(“,” & sT1, “,”). IsiT1(0) = “1”.

ii. Penggal setiap karakter sT2 dengan pemisah koma dan masukkan ke variabel IsiT2. IsiT2 = Split(“,” & sT2, “,”). IsiT2(0) = “1”.

iii. I2 = GetIndex(Kombinasi(I1).Var1).

iv. I3 = GetIndex(Kombinasi(I1).Var2).

v. Jika variabel1 dan variabel2 pada kombinasi(I1) terdapat pada keadaan daerah asal / Jika IsiT1(I2) = “1” dan IsiT1(I3) = “1”, maka lakukan prosedur berikut.

1) Hilangkan keberadaan 2 variabel pada daerah asal. IsiT1(I2) = “0”. IsiT1(I3) = “0”.

2) Munculkan keberadaan 2 variabel pada daerah tujuan. IsiT2(I2) = “1”. IsiT2(I3) = “1”.

3) Cek kombinasi(I1) apakah menyalahi aturan. Set B1=*False*. Untuk I2 = 1 sampai Ubound(Control), lakukan prosedur dari poin (3) sampai poin (9).
4) I3 = GetIndex(Control(I2).Var1).
5) I4 = GetIndex(Control(I2).Var2).
6) I5 = GetIndex(Control(I2).Var3).
7) I6 = GetIndex(Control(I2).Var4).
8) Cek apakah keadaan asal setelah pergerakan menyalahi aturan. Jika IsiT1(I3) = "1" dan IsiT1(I4) = "1" dan IsiT1(I5) = "0" dan IsiT1(I6) = "0" maka B1 = True dan keluar dari *looping* (lanjutkan ke poin (10)).
9) Cek apakah keadaan tujuan setelah pergerakan menyalahi aturan. Jika IsiT2(I3) = "1" dan IsiT2(I4) = "1" dan IsiT2(I5) = "0" n IsiT1(I6) = "0" maka B1 = *True* dan keluar dari *looping* (lanjutkan ke poin (10)).
10) Apabila tidak menyalahi aturan (B1 = *False*), maka:
    a) Set sT3 = keadaan asal setelah pergeseran (diambil dari array IsiT1).
    b) Set sT4 = keadaan tujuan setelah pergeseran (diambil dari array IsiT2).
    c) Cek keadaan ke variabel sList. Jika keadaan belum pernah ada maka bentuk satu keadaan baru. Lanjutkan ke poin berikutnya. Jika tidak maka lanjutkan *looping* untuk variabel I1.

d) Set I3 = UBound(Keadaan) + 1.

e) ReDim Preserve Keadaan(I3).

f) Jika Keadaan(N).Rakit = “KIRI” maka Keadaan(I3).DKiri = sT3, Keadaan(I3).DKanan = sT4 dan Keadaan(I3).Rakit = “KANAN”.

g) Jika Keadaan(N).Rakit = “KANAN” maka Keadaan(I3).DKanan = sT3, Keadaan(I3).DKiri = sT4 dan Keadaan(I3).Rakit = “KIRI”.

h) Keadaan(I3).Aksi = Kombinasi(I1).Var1 & “,” & Kombinasi(I1).Var2.

i) Keadaan(I3).Induk = N.

j) Keadaan(I3).IsExpand = *False*.

k) Masukkan ke sList. sList = sList & "{Kr=" & Keadaan(I3).DKiri & "," & "Kn=" & Keadaan(I3).DKanan & "," & "R=" & Keadaan(I3).Rakit & "}".

l) Jika Keadaan(I3).DKanan = KTujuan (keadaan tujuan tercapai) maka set IsExpand = *True* untuk semua variabel *array* keadaan.

d. Tandai bahwa keadaan sudah ditelusuri / dikembangkan. Keadaan(N).IsExpand = *True*.

8. Ambil indeks array keadaan yang merupakan keadaan tujuan dan simpan ke variabel I2.
9. Simpan semua langkah-langkah dari keadaan tujuan ke keadaan awal.

a. ReDim Temp(0).

b. Selama Keadaan(I2).Induk > 0,

    i. Simpan ke variabel Temp.

        1) I1 = UBound(Temp) + 1.

        2) ReDim Preserve Temp(I1).

        3) Temp(I1) = Keadaan(I2).

    ii. Gerakkan satu level ke atas. I2 = Keadaan(I2).Induk.

c. Simpan keadaan awal pada variabel Temp.

    i. I1 = UBound(Temp) + 1.

    ii. ReDim Preserve Temp(I1).

    iii. Temp(I1) = Keadaan(1).

10. Simpan semua langkah-langkah dari keadaan awal ke keadaan tujuan ke *array* Solusi (Balikkan isi variabel *array* Temp).

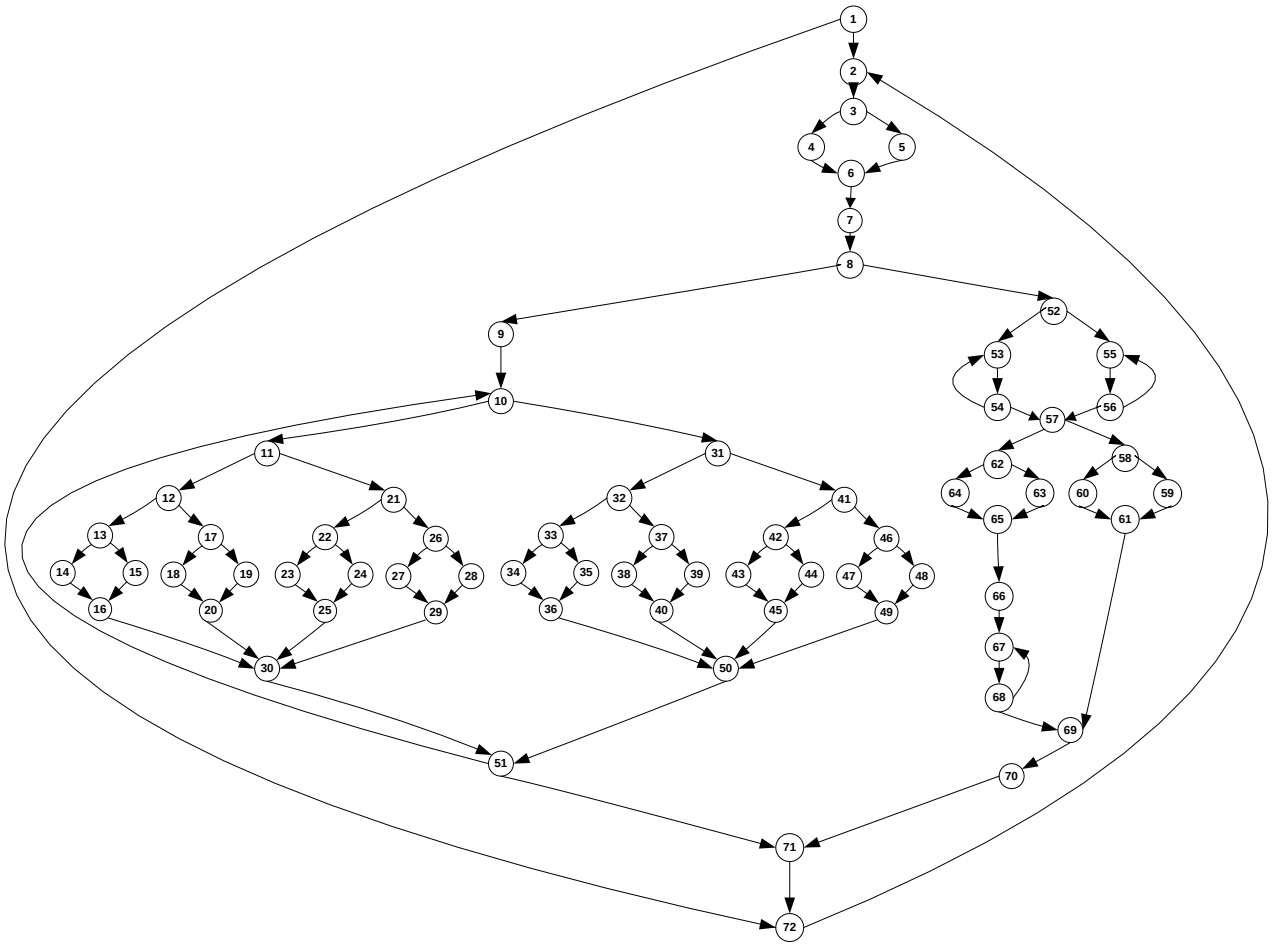
    a. ReDim Solusi(UBound(Temp) – 1).

    b. Untuk I1 = 0 Sampai Ubound(Temp)-1, set Solusi(I1) = Temp(UBound(Temp) – I1).

4.2 Pengujian *White Box*

Proses pengujian adalah menggunakan basis path yaitu menterjemahkan logika algoritma dengan diagram alir. Pengujian dilakukan untuk menguji apakah logika algoritma yang diterapkan dapat berjalan sesuai dengan yang diinginkan.

4.2.1 Diagram Alir



Gambar 4.13. Diagram alir dari logika algoritma program

4.2.2 Algoritma Program

```
1:   N = 0
2:   While Not (IsAllExpand(Keadaan))
3:       N = N + 1
4:       If Keadaan(N).Rakit = "KIRI" Then
           sT1 = Keadaan(N).DKiri
           sT2 = Keadaan(N).DKanan
5:       Else
           sT1 = Keadaan(N).DKanan
           sT2 = Keadaan(N).DKiri
6:       End If
7:       For I1 = 1 To UBound(Kombinasi)
           IsiT1 = Split("," & sT1, ",")
           IsiT2 = Split("," & sT2, ",")
           IsiT1(0) = "1"
           IsiT2(0) = "1"
           I2 = GetIndex(Kombinasi(I1).Var1)
           I3 = GetIndex(Kombinasi(I1).Var2)
8:         If IsiT1(I2) = "1" And IsiT1(I3) = "1" Then
               IsiT1(I2) = "0"
               IsiT1(I3) = "0"
               IsiT2(I2) = "1"
               IsiT2(I3) = "1"
9:      B1 = False
10:           For I2 = 1 To UBound(control)
                  I3 = GetIndex(control(I2).Var1)
                  I4 = GetIndex(control(I2).Var2)
                  I5 = GetIndex(control(I2).Var3)
                  I6 = GetIndex(control(I2).Var4)
                  I7 = GetIndex(control(I2).Var5)
11:                If control(I2).Var4 = ""
12:                  Then
13:                    If IsiT1(I3) = "1" And _
                         IsiT1(I4) = "1" And _
                         IsiT1(I5) = "0" Then
14:                     B1 = True
15:                         Exit For
16:                    End If
17:                    If IsiT2(I3) = "1" And _
                              IsiT2(I4) = "1" And _
                              IsiT2(I5) = "0" Then
18:                           B1 = True
19:                           Exit For
20:                    End If
21:              Else
22:                   If IsiT1(I3) = "1" And _
                              IsiT1(I4) = "1" And _
                              IsiT1(I7) = "1" And _
                              IsiT1(I5) = "0" And _
                              IsiT1(I6) = "0" Then
23:                         B1 = True
24:                         Exit For
25:                    End If
26:                    If IsiT2(I3) = "1" And _
                              IsiT2(I4) = "1" And _
                              IsiT2(I7) = "1" And _
                              IsiT2(I5) = "0" And _
                              IsiT2(I6) = "0" Then
27:                           B1 = True
28:                       Exit For
29:                    End If
30:              End If
31:              If control(I2).Var3 = ""
32:                   Then
33:                    If IsiT1(I3) = "1" And _
                              IsiT1(I4) = "1" And _
                              IsiT1(I5) = "0" Then
34:                         B1 = True
35:                         Exit For
36:                    End If
37:                    If IsiT2(I3) = "1" And _
                              IsiT2(I4) = "1" And _
                              IsiT2(I5) = "0" Then
38:                         B1 = True
39:                         Exit For
40:                    End If
41:               Else
42:                    If IsiT1(I3) = "1" And _
                              IsiT1(I4) = "1" And _
                              IsiT1(I7) = "1" And _
                              IsiT1(I5) = "0" And _
                              IsiT1(I6) = "0" Then
43:                         B1 = True
44:                         Exit For
45:                    End If
46:                    If IsiT2(I3) = "1" And _
                              IsiT2(I4) = "1" And _
                              IsiT2(I7) = "1" And _
                              IsiT2(I5) = "0" And _
                              IsiT2(I6) = "0" Then
47:                         B1 = True
48:                         Exit For
49:                    End If
50:              End If
51:              Next I2
52:      If B1 = False Then
             sT3 = ""
53:          For I4 = 1 To UBound(IsiT1)
54:            If I4 > 1 Then sT3 = sT3 & ","
                 sT3 = sT3 & IsiT1(I4)
             Next I4
             sT4 = ""
55:          For I4 = 1 To UBound(IsiT2)
56:             If I4 > 1 Then sT4 = sT4 & ","
                  sT4 = sT4 & IsiT2(I4)
             Next I4
57:         Else
58:         If Keadaan(N).Rakit = "KIRI" Then
59:           sT5 = "{Kr=" & sT3 & "," & _
                       "Kn=" & sT4 & "," & _
                       "R=KANAN}"
60:          Else
                sT5 = "{Kr=" & sT4 & "," & _
                       "Kn=" & sT3 & "," & _
                       "R=KIRI}"
61:          End If
62:          If InStr(1, sList, sT5) = 0 Then

                 I3 = UBound(Keadaan) + 1
                 ReDim Preserve Keadaan(I3)

63:              If Keadaan(N).Rakit = "KIRI" Then
                     'Update keadaan kiri = 0
                     Keadaan(I3).DKiri = sT3

                     'Update keadaan kanan = 1
                     Keadaan(I3).DKanan = sT4

                     'Posisi Rakit
                     Keadaan(I3).Rakit = "KANAN"
64:              Else
                     'Update keadaan kanan = 0
                     Keadaan(I3).DKanan = sT3

                     'Update keadaan kiri = 1
                     Keadaan(I3).DKiri = sT4

                     'Posisi Rakit
                     Keadaan(I3).Rakit = "KIRI"
65:              End If

                  'Aksi
                  Keadaan(I3).Aksi =
Kombinasi(I1).Var1 & "," & _

Kombinasi(I1).Var2

                  Keadaan(I3).Induk = N
                  Keadaan(I3).IsExpand = False
                  sList = sList & "{Kr=" &
Keadaan(I3).DKiri & "," & _
                  "Kn=" & Keadaan(I3).DKanan & "," &
_
                  "R=" & Keadaan(I3).Rakit & "}"

                  'Goal State?
66:               If Keadaan(I3).DKanan = KTujuan
Then
                      'Expand all
67:                  For I5 = N To UBound(Keadaan)
                         Keadaan(I5).IsExpand =
True
68:                   Next I5
                      Exit For
                  End If

69:          End If

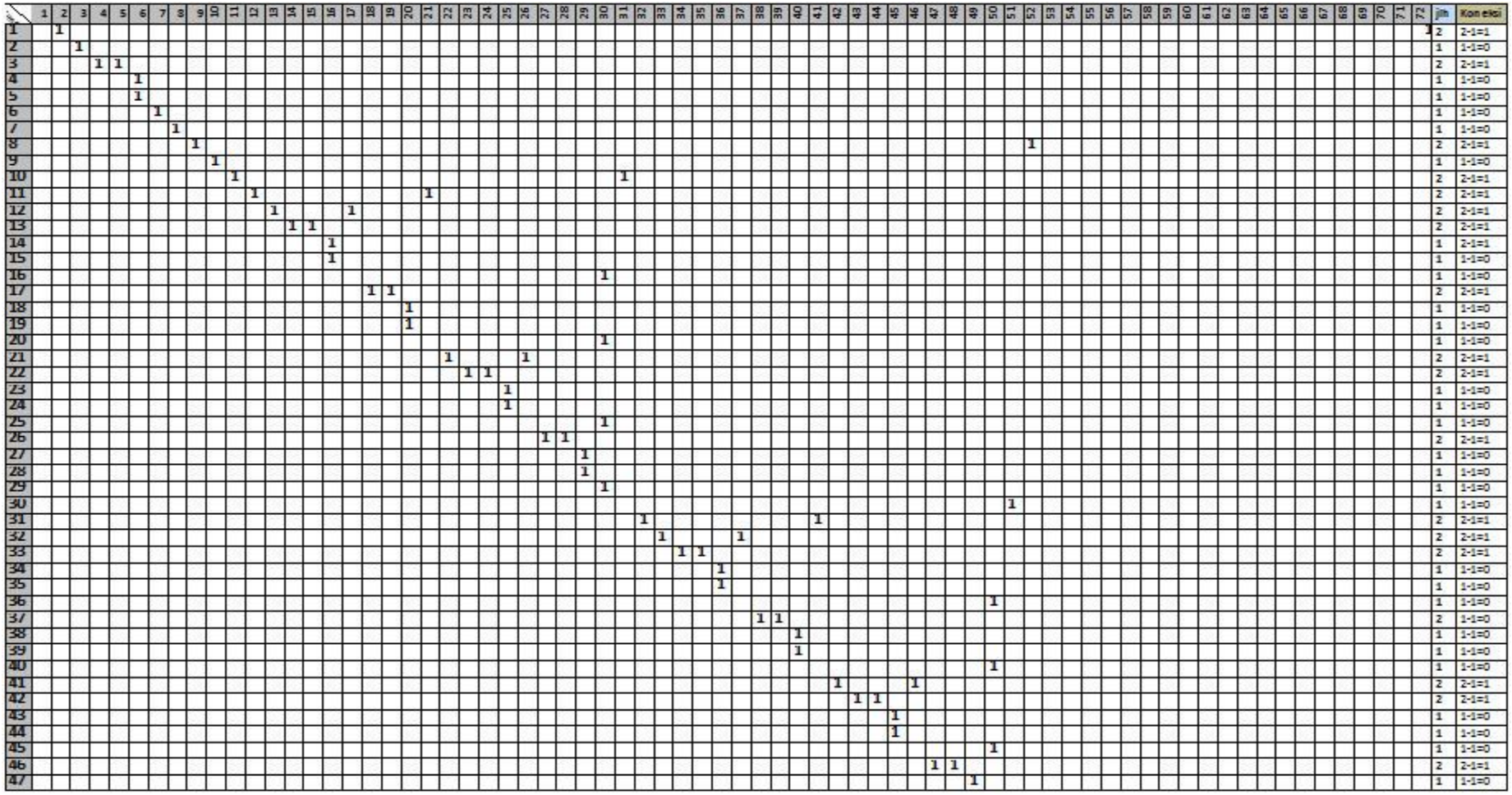
70:      End If

71: End If

72:   Next I1
```

4.2.3 Matriks Grafik

Tabel 4.3. Matriks Grafik



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
48																								
49																								
50																								
51										1														
52																								
53																								
54																								
55																								
56																								
57																								
58																								
59																								
60																								
61																								
62																								
63																								
64																								
65																								
66																								
67																								
68																								
69																								
70																								
71																								
72		1																						

	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
48																								
49																								
50																								
51																								
52																								
53																								
54																								
55																								
56																								
57																								
58																								
59																								
60																								
61																								
62																								
63																								
64																								
65																								
66																								
67																								
68																								
69																								
70																								
71																								
72																								

	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	jlh	Koneksi
48	1																								1	1-1=0
49		1																							1	1-1=0
50			1																						1	1-1=0
51																							1		2	2-1=1
52					1		1																		2	2-1=1
53						1																			1	1-1=0
54					1				1																2	2-1=1
55								1																	1	1-1=0
56							1		1																2	2-1=1
57										1				1											2	2-1=1
58											1	1													2	2-1=1
59													1												1	1-1=0
60													1												1	1-1=0
61																					1				1	1-1=0
62															1	1									2	2-1=1
63																	1								1	1-1=0
64																	1								1	1-1=0
65																		1							1	1-1=0
66																			1						1	1-1=0
67																				1					1	1-1=0
68																			1		1				2	2-1=1
69																						1			1	1-1=0
70																							1		1	1-1=0
71																								1	1	1-1=0
72																									1	1-1=0
Total Edge																									98	27

Jumlah Region (V(G)) = **28**

Jumlah Simpul (N) = 72 Buah

Jumlah Link atau Edge (E) = 98 Buah

V(G) = (E-N) + 2

V(G) = (98-72) + 2 = **28** atau

V(G) = Simpul Predikat + 1 = 27+1= **28**

4.2.4 Kesimpulan Hasil Pengujian White Box

Hasil pengujian menunjukkan bahwa logika algoritma yang diterapkan pada program berjalan sesuai dengan yang diharapkan. Hal ini teruji dengan pengujian pada hasil pengujian jalur logika yang telah dilakukan diatas yaitu pengujian setiap logika yang saling berhubungan (Link/Edge) antar simpul logika (N) dan daerah (V(G)) yang dilalui oleh program.

4.3 Pengujian *Black Box*

Proses pengujian adalah mencoba program dengan memasukkan data kedalam form – form masukkan yang telah disediakan. Pada tahap ini merupakan kelanjutan dari tahap implementasi yaitu melakukan pengujian terhadap aplikasi yang dibangun. Pengujian dilakukan dengan pengujian *black box* yang berfokus pada persyaratan fungsional perangkat lunak.

4.3.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan dengan menguji sistem secara *Alpha* dan *Betha*, dapat dilihat pada Tabel 4.3 berikut ini :

Tabel 4.4. Rencana Pengujian

Kelas uji	Butir Uji	Jenis Pengujian
Melihat Teori	Lihat teori *State And Space*	*Black Box*
	Lihat teori *Breadth First Search*	
	Lihat Contoh Soal	
Pengolahan *Problem*	Tambah *Problem*	*Black Box*
	Ubah *Problem*	
	Hapus *Problem*	
Pengolahan Variabel	Tambah *Variabel*	*Black Box*

	Ubah *Variabel*	
	Hapus *Variabel*	
Pengolahan Control	Tambah *Control*	*Black Box*
	Ubah *Control*	
	Hapus *Control*	
Melihat Solusi	Pilih Solusi	*Black Box*
	Pencarian Solusi Manual	
	Pencarian Solusi AI	

4.3.2 Kasus dan Pengujian *Alpha*

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian *Alpha* pada Aplikasi Bantu Kecerdasan Buatan Pada Permasalahan Ruang Keadaan (*State and Space*) sebagai berikut :

4.3.2.1 Pengujian Melihat Teori

1. Pengujian Lihat Teori *State And Space*

Tabel 4.5. Pengujian Lihat Teori *State And Space*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu materi *state and space*	Data materi *state and space* ditampilkan	Data materi *state and space* ditampilkan	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kosong tidak ada memilih menu materi *state and space*	Tidak menampilkan tampilan materi	Tidak menampilkan tampilan materi	[x]Diterima [ ]Tidak Diterima

2. Pengujian Lihat Teori *Breadth-First Search*

Tabel 4.6. Pengujian Lihat Teori *Breadth-First Search*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu materi *Breadth-First Search*	Data materi *Breadth-First Search* ditampilkan	Data Materi *Breadth-First Search* ditampilkan	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kosong tidak ada memilih menu materi *Breadth-First Search*	Tidak menampilkan tampilan materi	Tidak menampilkan tampilan materi	[x]Diterima [ ]Tidak Diterima

3. Pengujian Lihat Contoh Soal

Tabel 4.7. Pengujian Lihat Contoh Soal

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Pilih menu Contoh Soal	Data Contoh Soal ditampilkan	Data Contoh Soal ditampilkan	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Kosong tidak ada memilih menu Contoh Soal	Tidak menampilkan tampilan contoh soal	Tidak menampilkan tampilan contoh soal	[x]Diterima [ ]Tidak Diterima

4.3.2.2 Pengujian Pengolahan Problem

1. Pengujian tambah problem

Tabel 4.8. Pengujian tambah problem

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID *Problem* Nama *Problem*	Data tersimpan ke dalam database	Data tersimpan ke dalam database	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID Problem Nama *Problem* tetapi salah satunya kosong	Tidak terjadi proses simpan, menampilkan pesan 'ID *Problem* harus diisi' atau pesan 'Nama *Problem* harus diisi'	Tidak terjadi proses simpan, menampilkan pesan 'ID *Problem* harus diisi' atau pesan 'Nama *Problem* harus diisi'	[x]Diterima [ ]Tidak Diterima

2. Pengujian ubah *problem*

Tabel 4.9. Pengujian ubah *problem*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Problem yang akan diubah	Menampilkan *form* pengisian data	Menampilkan *form* pengisian data sesuai yang diharapkan	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID *Problem* Nama Problem tetapi salah satunya kosong	Tidak terjadi proses simpan, menampilkan pesan 'ID *Problem* harus diisi' atau pesan 'Nama *Problem* harus diisi'	Tidak terjadi proses simpan, menampilkan pesan 'ID *Problem* harus diisi' atau pesan 'Nama *Problem* harus diisi'	[x]Diterima [ ]Tidak Diterima

3. Pengujian hapus problem

Tabel 4.10. Pengujian hapus problem

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Problem yang akan dihapus	Data *Problem* dihapus	Data *problem* dihapus	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak memilih *problem* yang akan dihapus	Tidak terjadi proses penghapusan *problem*	Tidak terjadi proses penghapusan *problem*	[x]Diterima [ ]Tidak Diterima

4.3.2.3 Pengujian Pengolahan Variabel

1. Pengujian tambah *variabel*

Tabel 4.11. Pengujian tambah *variabel*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID *Problem* ID *Variabel* Nama *Variabel* *Driver* Foto	Data tersimpan ke dalam database	Data tersimpan ke dalam database	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID Problem ID Variabel Nama Variabel Driver Foto tetapi salah satu diantara ID Problem ID Variabel Nama Variabel Foto	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘ID Variabel harus diisi’ atau pesan ‘Nama Variabel harus diisi’ atau pesan ‘Gambar harus diisi’	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘ID Variabel harus diisi’ atau pesan ‘Nama Variabel harus diisi’ atau pesan ‘Gambar harus diisi’	[x]Diterima [ ]Tidak Diterima

kosong			

2. Pengujian ubah *variabel*

Tabel 4.12. Pengujian ubah *variabel*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Variabel yang akan diubah	Menampilkan form pengisian data	Menampilkan form pengisian data sesuai yang diharapkan	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID Problem ID Variabel Nama Variabel Driver Foto tetapi salah satu diantara ID Problem ID Variabel Nama Variabel Foto kosong	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘ID Variabel harus diisi’ atau pesan ‘Nama Variabel harus diisi’ atau pesan ‘Gambar harus diisi’	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘ID Variabel harus diisi’ atau pesan ‘Nama Variabel harus diisi’ atau pesan ‘Gambar harus diisi’	[x]Diterima [ ]Tidak Diterima

3. Pengujian hapus *variabel*

Tabel 4.13. Pengujian hapus *variabel*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Variabel yang akan dihapus	Data *Variabel* dihapus	Data *Variabel* dihapus	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak memilih variabel yang	Tidak terjadi proses penghapusan variabel	Tidak terjadi proses penghapusan variabel	[x]Diterima [ ]Tidak Diterima

akan dihapus			

4.3.2.4 Pengujian Pengolahan *Control*

1. Pengujian tambah *control*

Tabel 4.14. Pengujian tambah *control*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID Problem Variabel1 AttackID Variabel2 Variabel3 Variabel4 Variabel5	Data tersimpan ke dalam database	Data tersimpan ke dalam database	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID Problem Variabel1 AttackID Variabel2 Variabel3 Variabel4 Variabel5 tetapi salah satu diantara ID Problem Variabel1 AttackID Variabel2 Variabel3 Kosong atau data Variabel1 Variabel2 Variabel3 sama	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘Variabel1 harus diisi’ atau pesan “AttackID harus diisi’ atau pesan ‘Variabel2 harus diisi” atau pesan “variabel3 harus diiisi” Atau pesan “terdapat variabel yang sama !”	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘Variabel1 harus diisi’ atau pesan “AttackID harus diisi’ atau pesan ‘Variabel2 harus diisi” atau pesan “variabel3 harus diiisi” Atau pesan “terdapat variabel yang sama !”	[x]Diterima [ ]Tidak Diterima

2. Pengujian ubah *control*

Tabel 4.15. Pengujian ubah *control*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Control yang akan diubah	Menampilkan form pengisian data	Menampilkan form pengisian data sesuai yang diharapkan	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
ID Problem Variabel1 AttackID Variabel2 Variabel3 Variabel4 Variabel5 tetapi salah satu diantara ID Problem Variabel1 AttackID Variabel2 Variabel3 Kosong atau data Variabel1 Variabel2 Variabel3 sama	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘Variabel1 harus diisi’ atau pesan “AttackID harus diisi’ atau pesan ‘Variabel2 harus diisi” atau pesan “variabel3 harus diiisi” Atau pesan “terdapat variabel yang sama !”	Tidak terjadi proses simpan, menampilkan pesan ‘ID Problem harus diisi’ atau pesan ‘Variabel1 harus diisi’ atau pesan “AttackID harus diisi’ atau pesan ‘Variabel2 harus diisi” atau pesan “variabel3 harus diiisi” Atau pesan “terdapat variabel yang sama !”	[x]Diterima [ ]Tidak Diterima

3. Pengujian hapus *control*

Tabel 4.16. Pengujian hapus *control*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
control yang akan dihapus	Data control dihapus	Data control dihapus	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak memilih control yang akan dihapus	Tidak terjadi proses penghapusan control	Tidak terjadi proses penghapusan control	[x]Diterima [ ]Tidak Diterima

4.3.2.5 Pengujian Melihat Solusi

1. Pengujian memilih solusi

Tabel 4.17. Pengujian memilih solusi

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Problem yang akan diselesaikan dan sisimulasikan	Data problem akan diperiksa pada tabel control, dan variabel dan akan menampilkan from penyelesaian solusi AI atau form penyelesaian Solusi Manual	Data problem akan diperiksa pada tabel control, dan variabel dan akan menampilkan from penyelesaian solusi AI atau form penyelesaian Solusi Manual	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kosong tidak ada Problem yang dipilih	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan “tidak ditemukan solusi pada permasalahan ini !”	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan “tidak ditemukan solusi pada permasalahan ini !”	[x]Diterima [ ]Tidak Diterima

Data control yang dimiliki oleh Problem tidak memenuhi format kombinasi yang berlaku	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan “tidak ditemukan solusi pada permasalahan ini !”	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan “tidak ditemukan solusi pada permasalahan ini !”	[x]Diterima [ ]Tidak Diterima

2. Pengujian solusi AI

Tabel 4.18. Pengujian solusi AI

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Problem Data Variabel Data Control	Data problem akan diperiksa pada tabel control, dan variabel dan akan menampilkan pesan “Keadaan tujuan tercapai” apabila keadaan tujuan tercapai. Dan menampilkan tabel langkah penyelesaian	Data problem akan diperiksa pada tabel control, dan variabel dan akan menampilkan pesan “Keadaan tujuan tercapai” apabila keadaan tujuan tercapai. Dan menampilkan tabel langkah penyelesaian	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data control yang dimiliki oleh Problem tidak memenuhi format kombinasi yang berlaku	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan “tidak ditemukan solusi pada permasalahan ini !”	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan “tidak ditemukan solusi pada permasalahan ini !”	[x]Diterima [ ]Tidak Diterima

3. Pengujian Solusi Manual

Tabel 4.19. Pengujian Solusi Manual

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Problem Data Variabel Data Control Pilih Variabel yang akan dijalankan	Data problem akan diperiksa pada tabel control, dan variabel dan akan menampilkan pesan "Keadaan tujuan tercapai" apabila keadaan tujuan tercapai. Dan menampilkan tabel langkah penyelesaian	Data problem akan diperiksa pada tabel control, dan variabel dan akan menampilkan pesan "Keadaan tujuan tercapai" apabila keadaan tujuan tercapai. Dan menampilkan tabel langkah penyelesaian	[x]Diterima [ ]Tidak Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Variabel yang dipilih menyalahi format control yang telah ditentukan	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan "Variabel1 akan menyerang variabel2 apabila variabel3 tidak ada ditempat !" **atau** pesan "Variabel1, Variabel5 akan menyerang variabel2 apabila variabel3 atau Variabel4 tidak ada ditempat !" **atau** pesan "Variabel1, Variabel5 akan menyerang variabel2 apabila variabel3 dan Variabel4 tidak ada ditempat !" **atau** "Tidak ada yang dapat mengemudikan rakit"	Tidak terjadi proses pencarian solusi dan akan menampilkan pesan "Variabel1 akan menyerang variabel2 apabila variabel3 tidak ada ditempat !" **atau** pesan "Variabel1, Variabel5 akan menyerang variabel2 apabila variabel3 atau Variabel4 tidak ada ditempat !" **atau** "Tidak ada yang dapat mengemudikan rakit"	[x]Diterima [ ]Tidak Diterima

4.3.3 Kesimpulan Hasil Pengujian *Alpha*

Berdasarkan hasil pengujian *Alpha* yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi sudah berjalan sesuai yang diharapkan, tetapi tidak menutup kemungkinan dapat terjadi kesalahan suatu saat pada saat aplikasi digunakan, sehingga membutuhkan proses maintenance untuk lebih mengetahui kekurangan dari aplikasi.

4.3.4 Pengujian *Betha*

Pengujian betha merupakan pengujian yang dilakukan secara langsung pada *user*, yaitu untuk mengetahui kualitas dari aplikasi yang telah dibangun, apakah sudah sesuai dengan yang diharapkan atau belum. Pengujian dilakukan dengan memberikan kuisioner kepada masing-masing pemakai.

Berikut ini adalah daftar pertanyaan pada kuisioner yang diberikan kepada *user* sebagai *operator*. :

1. Apakah aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian pada permasalahan ruang keadaan ini mudah dipelajari dan mudah digunakan?

   a. Sangat Setuju

   b. Setuju

   c. Cukup Setuju

   d. Kurang Setuju

   e. Tidak Setuju

2. Apakah tampilan aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian pada permasalahan ruang keadaan ini menarik?

   a. Sangat Setuju

b. Setuju

c. Cukup Setuju

d. Kurang Setuju

e. Tidak Setuju

3. Apakah aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian pada permasalahan ruang keadaan ini dapat mensimulasikan permasalahan ruang keadaan?

a. Sangat Setuju

b. Setuju

c. Cukup Setuju

d. Kurang Setuju

e. Tidak Setuju

4. Apakah aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian pada permasalahan ruang keadaan ini dapat membantu akademisi dalam kegiatan belajar mengajar dan memahami permasalahan ruang keadaan?

a. Sangat Setuju

b. Setuju

c. Cukup Setuju

d. Kurang Setuju

e. Tidak Setuju

5. Apakah aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian pada permasalahan ruang keadaan ini dapat membantu menyelesaikan permasalahan ruang keadaan?

a. Sangat Setuju

b. Setuju

c. Cukup Setuju

d. Kurang Setuju

e. Tidak Setuju

Berdasarkan data hasil kuisioner **(terlampir)**, dapat dicari persentasi dari masing -

masing jawaban dengan menggunakan rumus:

Y = P/Q * 100%.....(iiii)

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai Persentase

Tabel 4.20. Hasil Jawaban kuisioner *user* sebagai operator, dengan Q = 20.

No Pertanyaan	Keterangan	Jawaban					Jumlah
		a	b	c	d	e	
1	Responden (P)	4	15	1	-	-	20
	Responden (Y)	20%	75%	5%	-	-	100%
2	Responden (P)	6	11	3	-	-	20
	Responden (Y)	30%	55%	15%	-	-	100%
3	Responden (P)	3	16	1	-	-	20
	Responden (Y)	15%	80%	5%	-	-	100%
4	Responden (P)	3	15	2	-	-	20
	Responden (Y)	15%	75%	10%	-	-	100%

5	Responden (P)	2	17	1	-	-	20
	Responden (Y)	10%	85%	5%	-	-	100%

4.3.5 Kesimpulan Pengujian Betha

Berdasarkan hasil dari pengujian betha, maka dapat disimpulkan bahwa Aplikasi Bantu Kecerdasan Buatan Pembelajaran dan Penyelesaian Pada Permasalahan Ruang Keadaan (*State and Space*) ini mudah digunakan digunakan oleh pengguna, dapat mensimulasikan penyelesaian permasalahan, membantu pengguna untuk mempelajari permasalahan ruang keadaan pada bidang keilmuan kecerdasan buatan.