BAB 4
IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi permainan yang dikembangkan, yaitu Implementasi Algoritma *Fuzzy Logic* Dalam Pengambilan Keputusan Berinvestasi Pada *Game Time Management* (*Me In Trouble*). Hasil perancangan pada tahap sebelumnya kemudian diimplementasikan ke dalam bahasa pemrograman. Setelah tahap implementasi selesai, maka dilakukan tahap pengujian terhadap aplikasi Implementasi Algoritma *Fuzzy Logic* dalam pengambilan keputusan berinvestasi pada *game time management* (*Me In Trouble*).

4.1 Implementasi

Tahap implementasi merupakan tahap penciptaan perangkat lunak dan juga tahap kelanjutan dari kegiatan perancangan aplikasi. Tahap ini merupakan tahap dimana aplikasi siap untuk dioperasikan, yang terdiri dari penjelasan mengenai lingkungan implementasi, baik itu lingkungan perangkat keras, maupun lingkungan perangkat lunak, serta implementasi program.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk menjalankan aplikasi Implementasi algoritma *Fuzzy Logic* dalam pengambilan keputusan berinvestasi pada *game time management* (*Me In Trouble*) ini tidak harus yang berspesifikasi tinggi. Spesifikasi minimum yang dapat digunakan untuk menjalankan aplikasi Implementasi algoritma *Fuzzy Logic* dalam pengambilan keputusan berinvestasi pada *game time management* (*Me In Trouble*) adalah sebagai berikut:

1. *Processor* dengan kecepatan 1.8 GHz ke atas.
2. RAM 256 MB
3. VGA 128 MB
4. *Harddisk available* 250MB
5. *Monitor*

6. Mouse

4.1.2 Implementasi Perangkat Lunak

Untuk menjalankan aplikasi ini tidak diperlukan software khusus karena aplikasi telah berformat .exe Implementasi Aplikasi. Aplikasi pengimplementasian ini menggunakan *Unity* dan teknik dengan grafik 2D yang dibangun dengan menggunakan bahasa pemrograman C#.

4.1.3 Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi Implementasi algoritma *Fuzzy Logic* dalam pengambilan keputusan berinvestasi pada *game time management* (*Me In Trouble*) adalah sebagai berikut:

1. Antarmuka Menu Utama

Antarmuka Menu Utama ini menggambarkan tampilan awal dari *game*, terdapat 3 pilihan menu yang tersedia pada antarmuka ini yaitu menu mulai, petujuk dan keluar. Tampilan antarmuka menu utama dapat dilihat pada lampiran A.

2. Antarmuka Menu Petunjuk

Antarmuka menu petunjuk ini menampilkan petunjuk dalam memainkan *game*. Tampilan antarmuka menu petunjuk dapat dilihat pada lampiran A.

3. Antarmuka Mulai Game

Antarmuka mulai *game* ini menampilkan tampilan saat *game* pertama kali dimulai. Tapilan antarmuka mulai *game* dapat dilihat pada lampiran A.

4. Antarmuka Pesanan

Antarmuka pesanan ini menampilkan munculnya pesanan yang muncul yang harus diingat oleh pemain, waktu yang diberikan untuk mengingat pesanan tersebut adalah 8 detik. Akan disediakan 8 pesanan yang harus diingat oleh pemain dan disajikan dalam waktu 120 detik namun tampilan antarmuka menu pesanan secara umum dapat dilihat pada lampiran A.

5. Antarmuka Penyajian Burger

Antarmuka penyajian burger ini menampilkan lapisan-lapisan burger yang tersedia dan pemain harus menyajikan burger sesuai pesanan dengan klik lapisan

lapisan yang tersedia. Tampilan antarmuka menu penyajian burger dapat dilihat pada lampiran A.

6. Antarmuka Penyajian Kopi

Antarmuka penyajian kopi ini menampilkan lapisan-lapisan kopi yang tersedia dan pemain harus menyajikan kopi sesuai pesanan dengan klik lapisan lapisan yang tersedia. Tampilan antarmuka menu penyajian kopi dapat dilihat pada lampiran A.

7. Antarmuka Keputusan Akhir

Antarmuka keputusan akhir akan menampilkan keputusan akhir yang didapat setelah melakukan penyajian pesanan selama 120 detik. Keputusan akhir ini menyatakan "Berinvestasi" atau "Tidak Investasi". Antarmuka keputusan akhir dapat dilhat pada lampiran A.

4.2 Pengujian

Tahap pengujian merupakan tahap pengujianperangkat lunak dan juga tahap kelanjutan dari kegiatan implementasi aplikasi. Tahap ini merupakan tahap dimana aplikasi yang sudah diimplementasikan siap untuk diuji.

4.2.1 Pengujian Sistem

Pengujian sistem dilakukan dengan 2 (dua) macam teknik pengujiana yaitu pengujian *white box*, pengujian *black box*. Pengujian yang menggunakan teknik *white box* lebih menitikberatkan pada struktur *internal* (*source code*) program, di mana setiap baris kode yang akan di compile dicek satu persatu untuk mengetahui apakah masih terjadi error. Pengujian pada black box tidak terlalu memperhatikan struktur internal (*source code*) dari program, tetapi lebih kepada menemukan kesalahan pada program. Pada perancangan fungsional digunakan pengujian *black box*.

4.2.1.1 Pengujian *Black Box*

Pengujian dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan.

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Berikut ini pengujian dari *game Me In Trouble* dapat dilihat pada tabel 4.1 sebagai berikut:

Tabel 4.1 Pengujian *Black Box*

No	Komponen Yang Di Uji	Jenis Pengujian
1.	Mulai Permainan	*Black Box*
2.	Penyajian Burger	*Black Box*
3.	Penyajian Kopi	*Black Box*

4.2.1.1.1 Kasus Dan Hasil Pengujian *Black Box*

Berikut ini adalah hasil dari pengujian perangkat lunak yang sudah dibangun menggunakan metode *black box* berdasarkan pengamatan yang sudah dilakukan dan menghasilkan kesimpulan dari pengujian tersebut.

1. Pengujian Mulai Permainan

   Pengujian mulai permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.2:

Tabel 4.2 Pengujian Mulai Permainan

Kasus Dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
- Score Burger = 0 - Score Kopi = 0 - Time Burger = 0 - Time Kopi = 0 - Score Investasi = 0 - Score Tidak Investasi = 0 - Score Hasil = 0 - Time Game = 120	Sistem menampilkan layar permainan dan atribut *score* dan *time* telah ter-*setting* sesuai dengan masukan.	Layar permainan muncul dan atribut *score* dan *time* telah ter-*setting* sesuai dengan masukan.	[√] Berhasil [ ] Tidak Berhasil

2. Pengujian Penyajian Burger

Pengujian penyajian burger dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.3:

Tabel 4.3 Pengujian Penyajian Burger

Kasus Dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
A. Jika Semua Lapisan Benar Dan Waktu Pelayanan Cepat - Score Burger = 50 - Time Burger = 2 B. Jika Semua Lapisan Benar Dan Waktu Pelayanan Sedang - Score Burger = 50 - Time Burger = 6 C. Jika Semua Lapisan Benar Dan Waktu Pelayanan Lambat - Score Burger = 50 - Time Burger = 8	Sistem mencatat dan menyimpan Score Burger dan Time Burger sesuai dengan kondisi masukan yang didapat.	Sistem berhasil mencatat dan menyimpan Score Burger dan Time Burger sesuai dengan kondisi masukan yang didapat.	[√] Berhasil [ ] Tidak Berhasil
Kasus Dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
A. Jika Satu Lapisan Salah Dan Waktu Pelayanan Cepat - Score Burger = 40	Sistem mencatat dan menyimpan Score Burger dan Time	Sistem berhasil mencatat dan menyimpan Score Burger	[√] Berhasil [ ] Tidak Berhasil

- Time Burger = 2 B. Jika Satu Lapisan Salah Dan Waktu Pelayanan Sedang - Score Burger = 40 - Time Burger = 6 C. Jika Satu Lapisan Salah Dan Waktu Pelayanan Lambat - Score Burger = 40 - Time Burger = 8 D. Jika Dua Lapisan Salah Dan Waktu Pelayanan Cepat - Score Burger = 30 - Time Burger = 2 E. Jika Dua Lapisan Salah Dan Waktu Pelayanan Sedang - Score Burger = 30 - Time Burger = 4 F. Jika Dua Lapisan Salah Dan Waktu Pelayanan Lambat - Score Burger = 30 - Time Burger = 8 G. Jika Tiga Lapisan Salah Dan Waktu Pelayanan Cepat - Score Burger = 20	Burger sesuai dengan kondisi masukan yang didapat.	dan Time Burger sesuai dengan kondisi masukan yang didapat.	

- Time Burger = 2 H. Jika Tiga Lapisan Salah Dan Waktu Pelayanan Sedang - Score Burger = 20 - Time Burger = 6 I. Jika Tiga Lapisan Salah Dan Waktu Pelayanan Lambat - Score Burger = 20 - Time Burger = 8 J. Jika Empat Lapisan Salah Dan Waktu Pelayanan Cepat - Score Burger = 10 - Time Burger = 2 K. Jika Empat Lapisan Salah Dan Waktu Pelayanan Sedang - Score Burger = 10 - Time Burger = 6 L. Jika Empat Lapisan Salah Dan Waktu Pelayanan Lambat - Score Burger = 10 - Time Burger = 8 M. Jika Semua Lapisan Salah Dan Waktu Pelayanan Cepat - Score Burger = 0			

- Time Burger = 2 N. Jika Semua Lapisan Salah Dan Waktu Pelayanan Sedang - Score Burger = 0 - Time Burger = 6 O. Jika Semua Lapisan Salah Dan Waktu Pelayanan Lambat - Score Burger = 0 - Time Burger = 8			

3. Pengujian Penyajian Kopi

Pengujian penyajian kopi dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.4:

Tabel 4.4 Pengujian Penyajian Kopi

Kasus Dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
A. Jika Semua Lapisan Benar Dan Waktu Pelayanan Cepat - Score Kopi = 50 - Time Kopi = 2 B. Jika Semua Lapisan Benar Dan Waktu Pelayanan Sedang - Score Kopi = 50 - Time Kopi = 6	Sistem mencatat dan menyimpan Score Kopi dan Time Kopi sesuai dengan kondisi masukan yang didapat.	Sistem berhasil mencatat dan menyimpan Score Kopi dan Time Kopi sesuai dengan kondisi masukan yang didapat.	[√] Berhasil [ ] Tidak Berhasil

C. Jika Semua Lapisan Benar Dan Waktu Pelayanan Lambat - Score Kopi = 50 - Time Kopi = 8			
Kasus Dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
A. Jika Satu Lapisan Salah Dan Waktu Pelayanan Cepat - Score Kopi = 40 - Time Kopi = 2 B. Jika Satu Lapisan Salah Dan Waktu Pelayanan Sedang - Score Kopi = 40 - Time Kopi = 6 C. Jika Satu Lapisan Salah Dan Waktu Pelayanan Lambat - Score Kopi = 40 - Time Kopi = 8 D. Jika Dua Lapisan Salah Dan Waktu Pelayanan Cepat - Score Kopi = 30 - Time Kopi = 2	Sistem mencatat dan menyimpan Score Kopi dan Time Kopi sesuai dengan kondisi masukan yang didapat.	Sistem berhasil mencatat dan menyimpan Score Kopi dan Time Kopi sesuai dengan kondisi masukan yang didapat.	[√] Berhasil [ ] Tidak Berhasil

E. Jika Dua Lapisan Salah Dan Waktu Pelayanan Sedang - Score Kopi = 30 - Time Kopi = 4 F. Jika Dua Lapisan Salah Dan Waktu Pelayanan Lambat - Score Kopi = 30 - Time Kopi = 8 G. Jika Tiga Lapisan Salah Dan Waktu Pelayanan Cepat - Score Kopi = 20 - Time Kopi = 2 H. Jika Tiga Lapisan Salah Dan Waktu Pelayanan Sedang - Score Kopi = 20 - Time Kopi = 6 I. Jika Tiga Lapisan Salah Dan Waktu Pelayanan Lambat - Score Kopi = 20 - Time Kopi = 8 J. Jika Empat Lapisan Salah Dan Waktu Pelayanan Cepat - Score Kopi = 10 - Time Kopi = 2			

K. Jika Empat Lapisan Salah Dan Waktu Pelayanan Sedang - Score Kopi = 10 - Time Kopi = 6 L. Jika Empat Lapisan Salah Dan Waktu Pelayanan Lambat - Score Kopi = 10 - Time Kopi = 8 M. Jika Semua Lapisan Salah Dan Waktu Pelayanan Cepat - Score Kopi = 0 - Time Kopi = 2 N. Jika Semua Lapisan Salah Dan Waktu Pelayanan Sedang - Score Kopi = 0 - Time Kopi = 6 O. Jika Semua Lapisan Salah Dan Waktu Pelayanan Lambat - Score Kopi = 0 - Time Kopi = 8			

4.2.1.2 Pengujian *White Box*

Pengujian *white box* merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan dengan menggunakan stuktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Tujuan dari pengujian *white box* untuk mengetahui cara kerja suatu

aplikasi atau perangkat lunak secara internal dan untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kontrol dari prosedur yang dirancang. Berikut adalah pseudocode untuk pembentukan himpunan *fuzzy* yang terdapat pada tabel 4.7:

Tabel 4.5 *Pseudocode* Himpunan *Fuzzy*

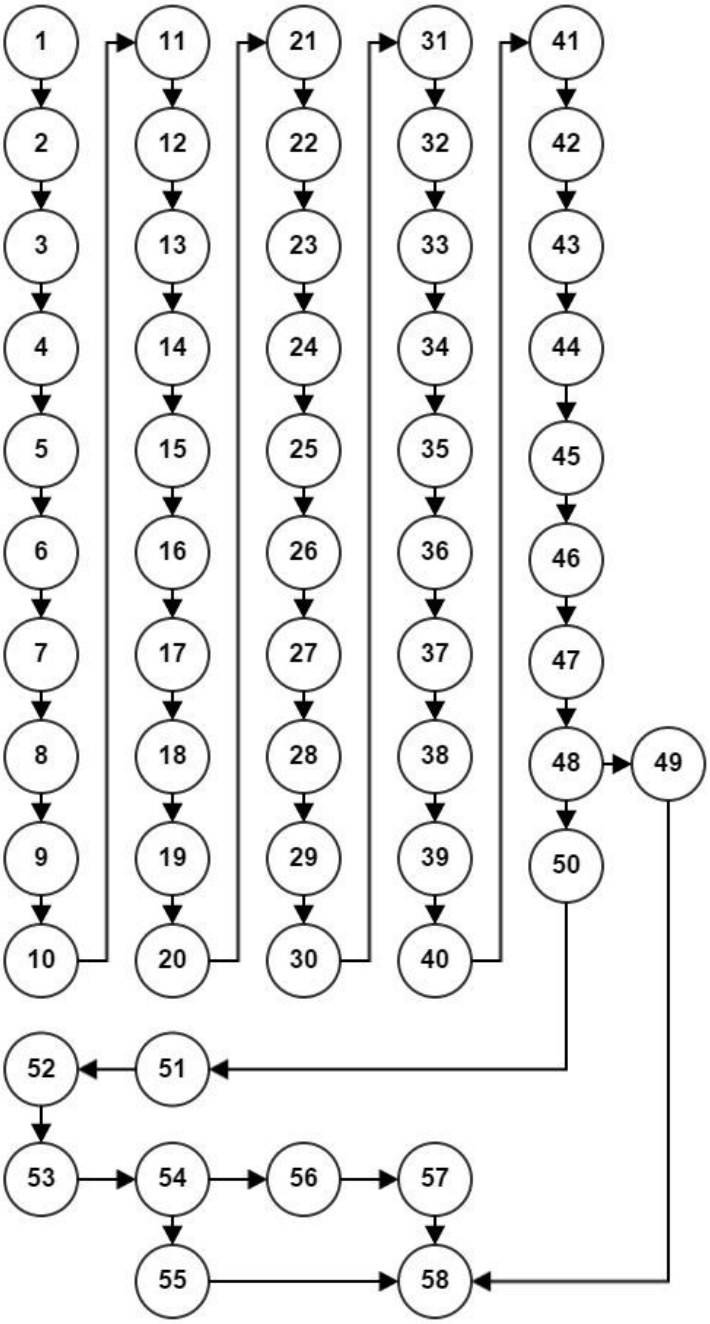
	//Variabel Waktu Burger
1	public float cepat;
2	private float cepat1 = 0;
3	private float cepat2 = 4;
4	private float cepat3 = 6;
5	public float sedang;
6	private float sedang1 = 4;
7	private float sedang2 = 6;
8	private float sedang3 = 7;
9	private float sedang4 = 9;
10	public float lambat;
11	private float lambat1 = 7;
12	private float lambat2 = 9;
13	private float lambat3 = 12;
	//Variabel Kesesuaian Burger
14	public float tidaksesuai;
15	private float tidaksesuai1 = 0;
16	private float tidaksesuai2 = 20;
17	private float tidaksesuai3 = 40;
18	public float sesuai;
19	private float sesuai1 = 20;
20	private float sesuai2 = 40;
21	private float sesuai3 = 50;
	//Variabel Waktu Kopi
22	public float cepaat;

```
23  private float cepaat1 = 0;
24  private float cepaat2 = 4;
25  private float cepaat3 = 6;
26  public float sedaang;
27  private float sedaang1 = 4;
28  private float sedaang2 = 6;
29  private float sedaang3 = 7;
30  private float sedaang4 = 9;
31  public float lambaat;
32  private float lambaat1 = 7;
33  private float lambaat2 = 9;
34  private float lambaat3 = 12;
    //Variabel Kesesuaian Kopi
35  public float tidaksesuaai;
36  private float tidaksesuaai1 = 0;
37  private float tidaksesuaai2 = 20;
38  private float tidaksesuaai3 = 40;
39  public float sesuaai;
40  private float sesuaai1 = 20;
41  private float sesuaai2 = 40;
42  private float sesuaai3 = 50;
43  hasil = 0;
44  hasil_investasi = 0.0f;
45  hasil_tidakinvestasi = 0;
46  for (int i=0; i<investasi.Length; i++) {
47  if (investasi[i]>0.0f) {
48  if (hasil_investasi<=0.0f) {
49  hasil_investasi = investasi[i];
50  } else if (investasi[i]>=hasil_investasi)
    { hasil_investasi = investasi[i]; } } }
51
```

```
52  for (int i=0; i<tidakinvestasi.Length; i++) {
53  if (tidakinvestasi[i]>0.0f) {
54  if (hasil_tidakinvestasi<=0.0f) {
55  hasil_tidakinvestasi = tidakinvestasi[i];
56  } else if (tidakinvestasi[i]>=hasil_tidakinvestasi)
57  { hasil_tidakinvestasi = tidakinvestasi[i]; } } }
58  hasil = ((hasil_tidakinvestasi * 50) + (hasil_investasi * 100)) /
    (hasil_investasi + hasil_tidakinvestasi);
```

a. *Flow Graph Fuzzy Logic*

Berikut ini merupakan *flow graph* dari logika fuzzy yang dapat dilihat pada gambar 4.12:



Gambar 4.1 ***Flow Graph Fuzzy Logic***

Keterangan:

: Menyatakan Kondisi

: Menyatakan Aksi

b. *Cyclomatic Complexity* V(G)

Dari *flow graph* di atas, dapat dihitung *cyclomatic complexity* yang di dapat dari perhitungan dibawah ini:

E = 59, N = 58

V(G) = E – N + 2

V(G) = 59 - 58 + 2

V(G) = 3

Keterangan

E = Jumlah Aksi

N = Jumlah Kondisi

c. *Independent Path*

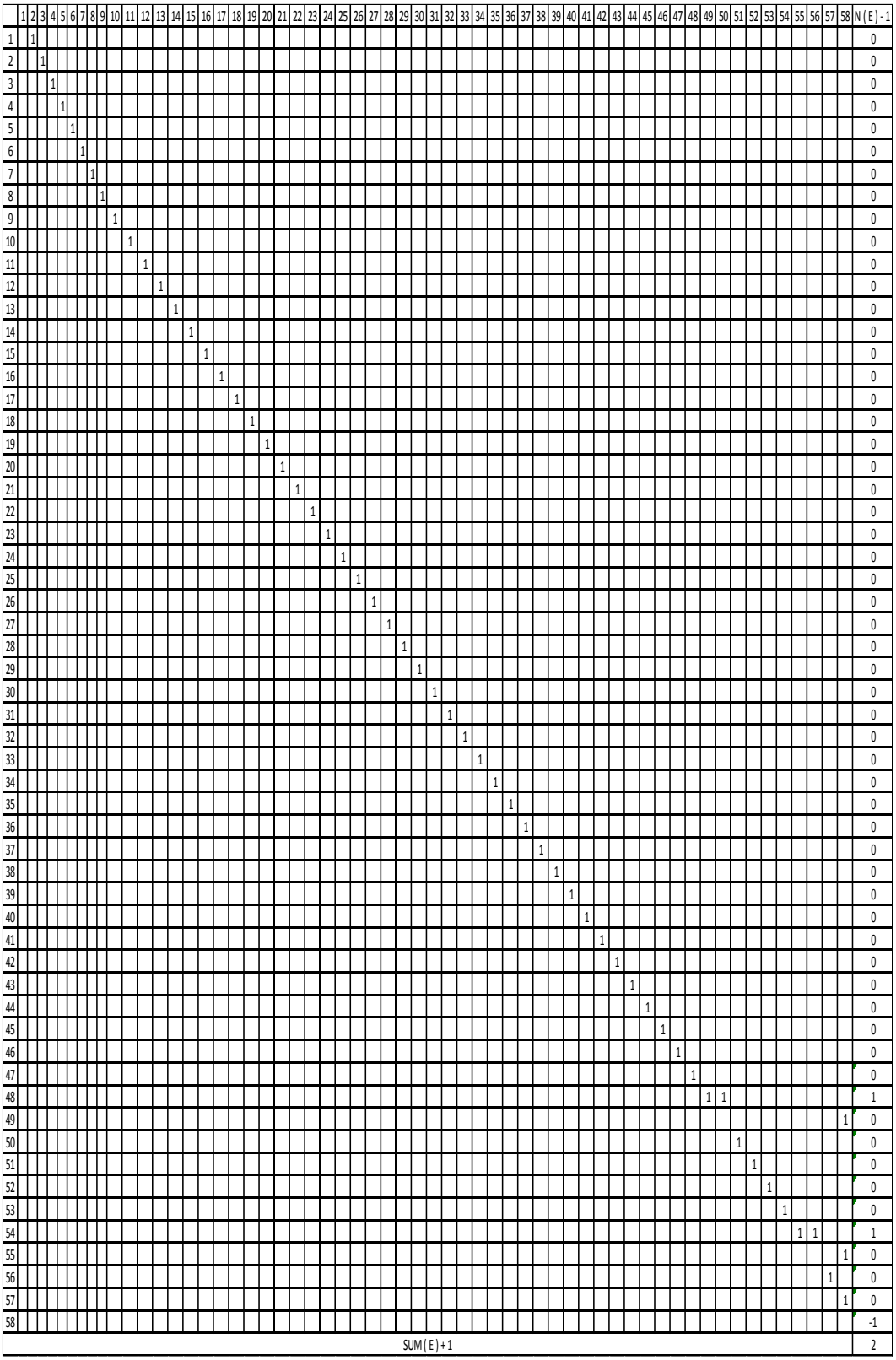
Dari hasil perhitungan *cyclomatic complexity* terdapat 5 independent path yaitu:

Path 1: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-58

Path 2: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-50-51-52-53-54-56-57-58

Path 3: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-58

d. *Graph Matriks Fuzzy Logic*



Gambar 4.2 Grafik Matrik Logika Fuzzy

Keterangan:

1. Baris dan kolom merepresentasikan simpul (node).
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul.

4.2.2 Pengujian Simulasi

Pengujian ditujukan pada tingkat kesesuaian nilai defuzzifikasi dengan perilaku yang sudah ditetapkan. Kondisi parameter yang digunakan sesuai dengan tabel 3.1. Berikut akan diberikan data hasil pengujian terhadap beberapa kondisi.

4.2.3 Pengujian Aturan *Fuzzy*

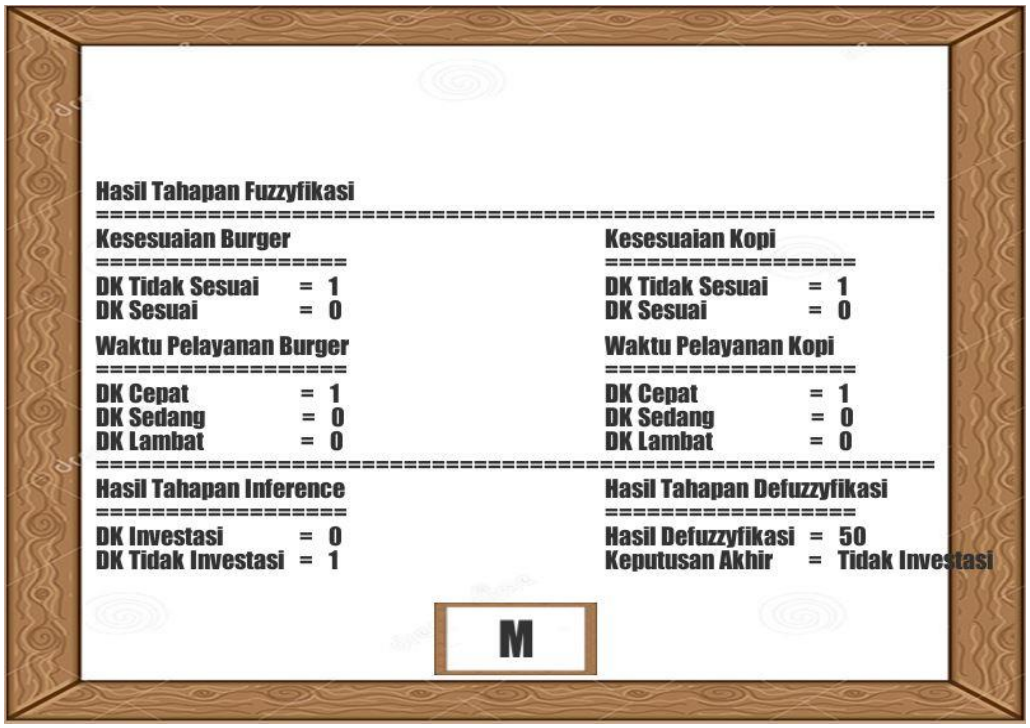
Rule yang akan diuji berjumlah 36 jenis *rule*, dan 2 jenis keputusan berinvestasi. Pengujian dilakukan pada hasil akhir yang ditentukan oleh waktu pelayanan dan kesesuaian pelayanan.

Parameter yang berpengaruh untuk keputusan akhir adalah:

1. Waktu Pelayanan Burger
2. Waktu Pelayanan Kopi
3. Kesesuaian Burger
4. Kesesuaian Kopi

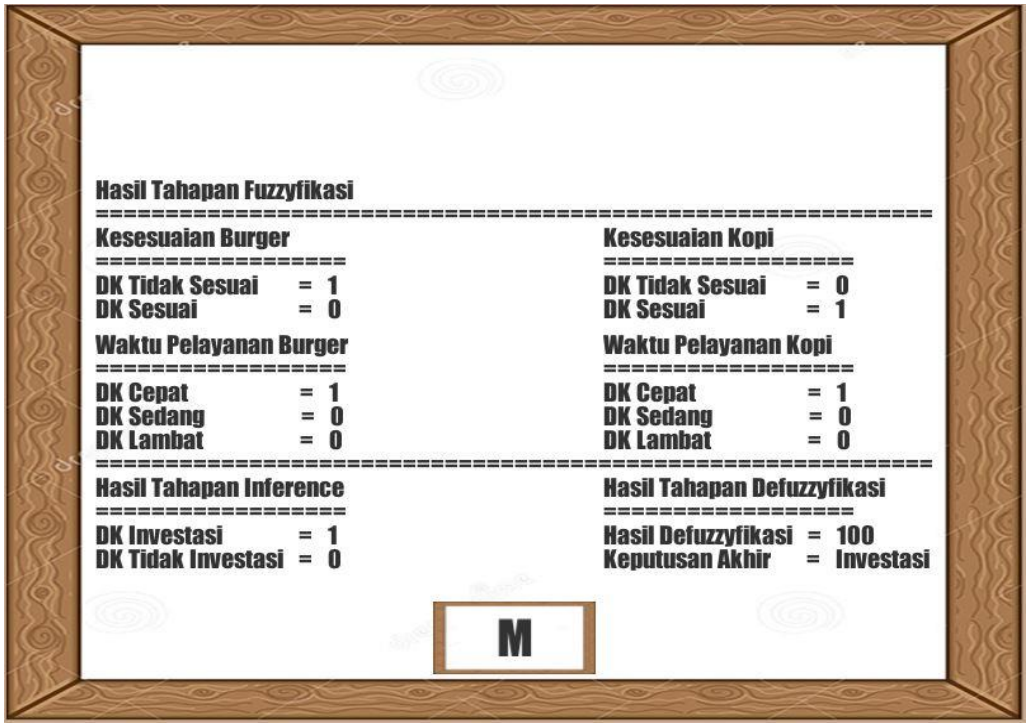
Berikut adalah pengujian *rule* yang dilakukan:

1. *Rule* 1



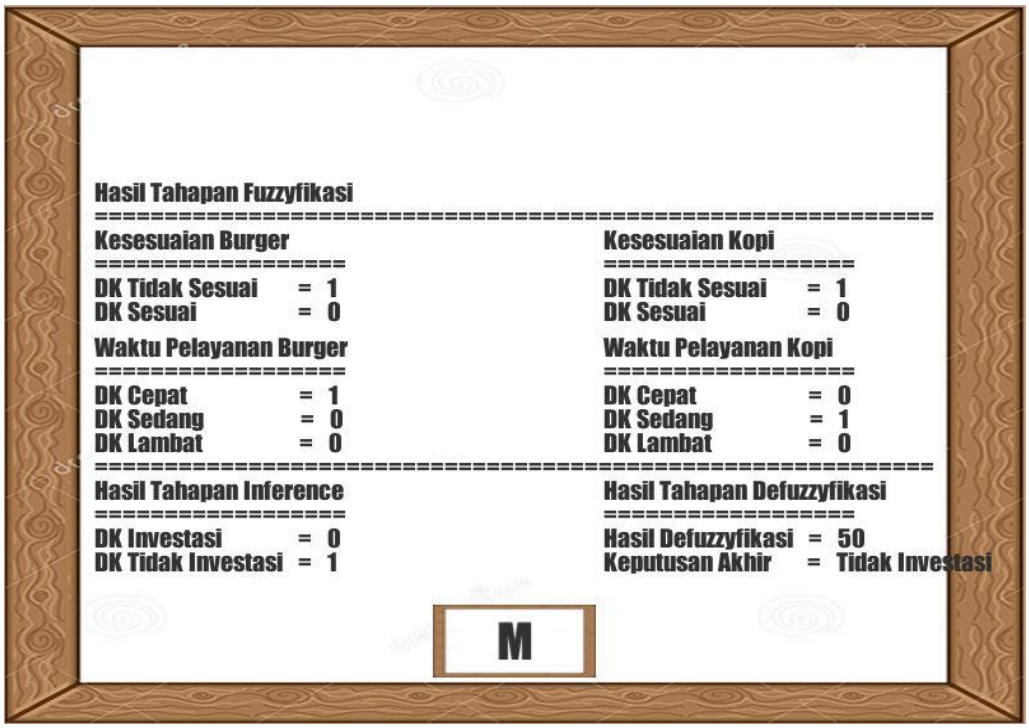
Gambar 4.3 Pengujian *Rule 1*

2. *Rule* 2



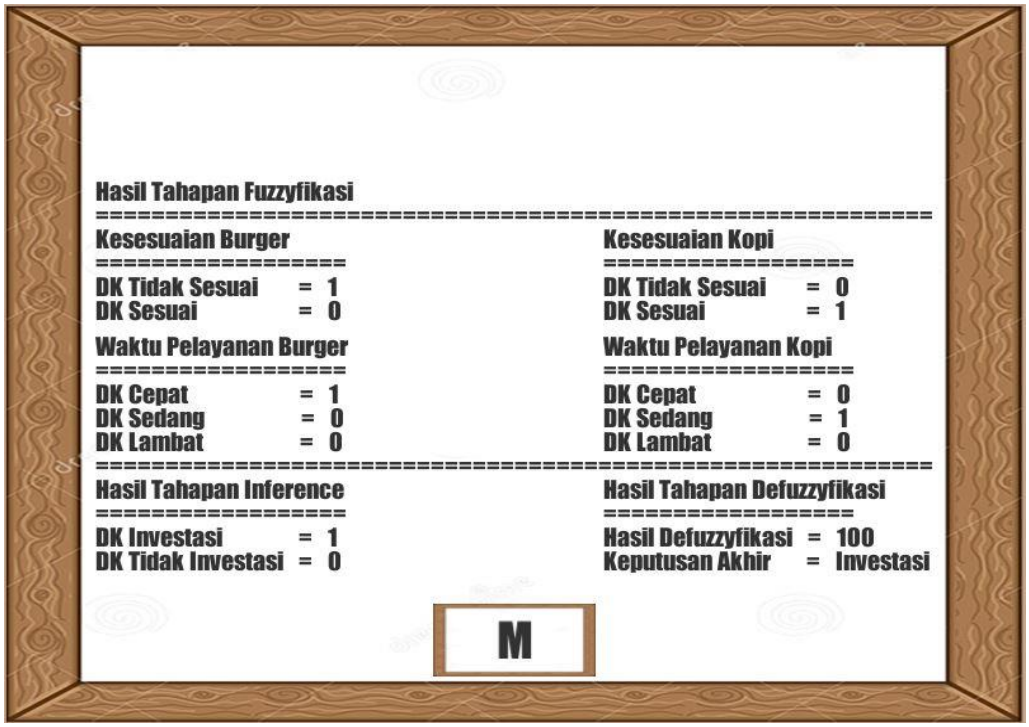
Gambar 4.4 Pengujian *Rule 2*

3. *Rule* 3



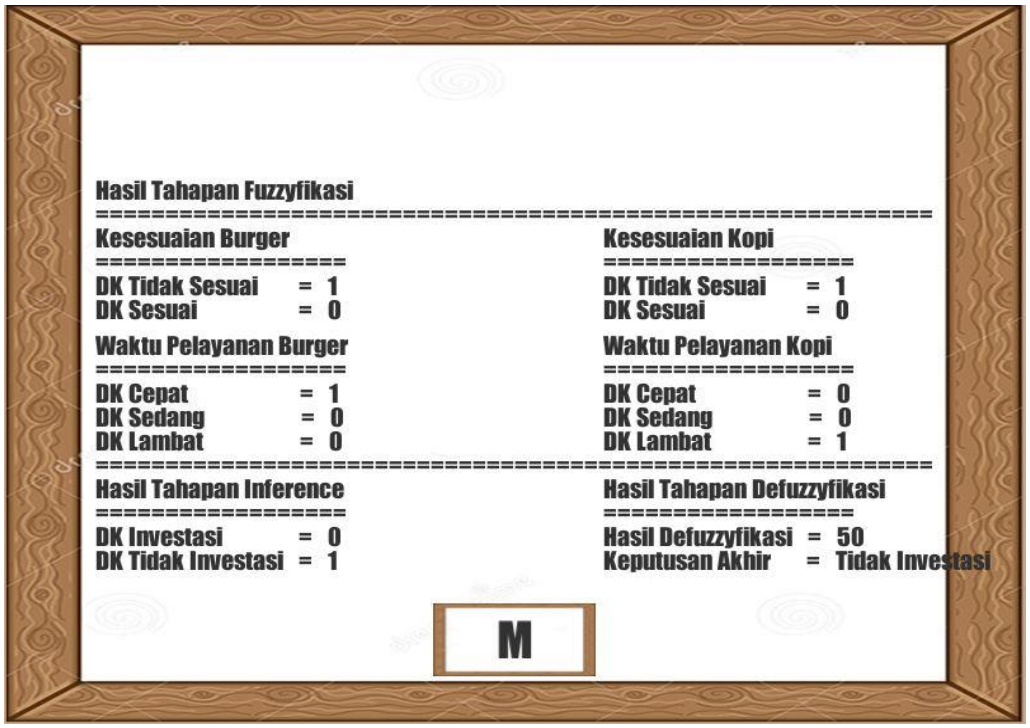
Gambar 4.5 Pengujian *Rule 3*

4. *Rule* 4



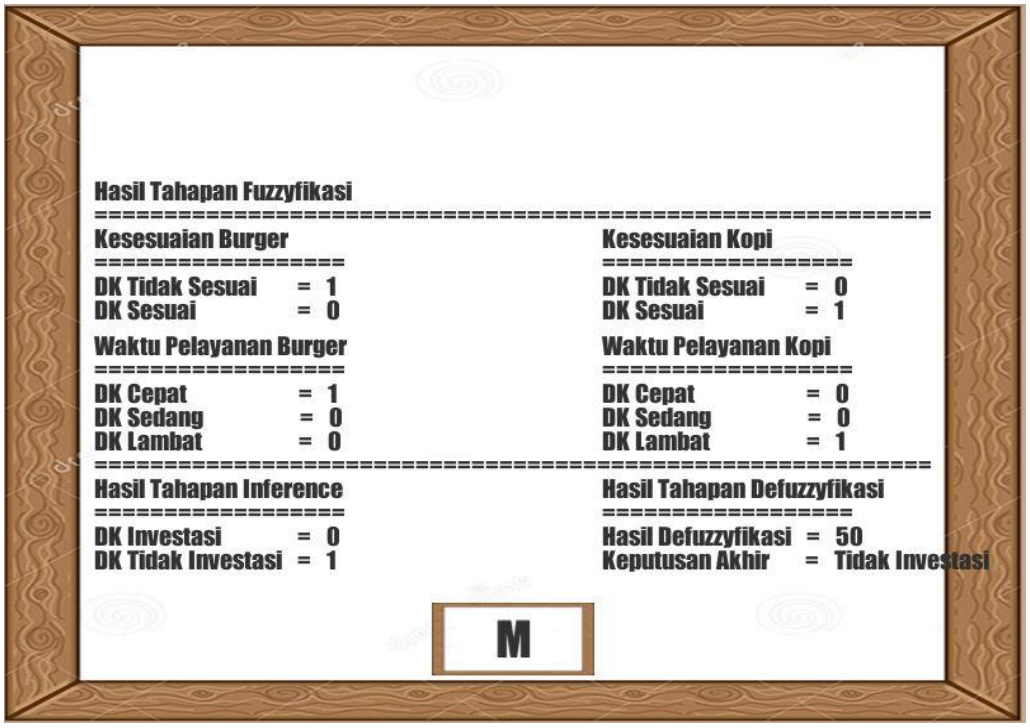
Gambar 4.6 Pengujian *Rule 4*

5. *Rule* 5



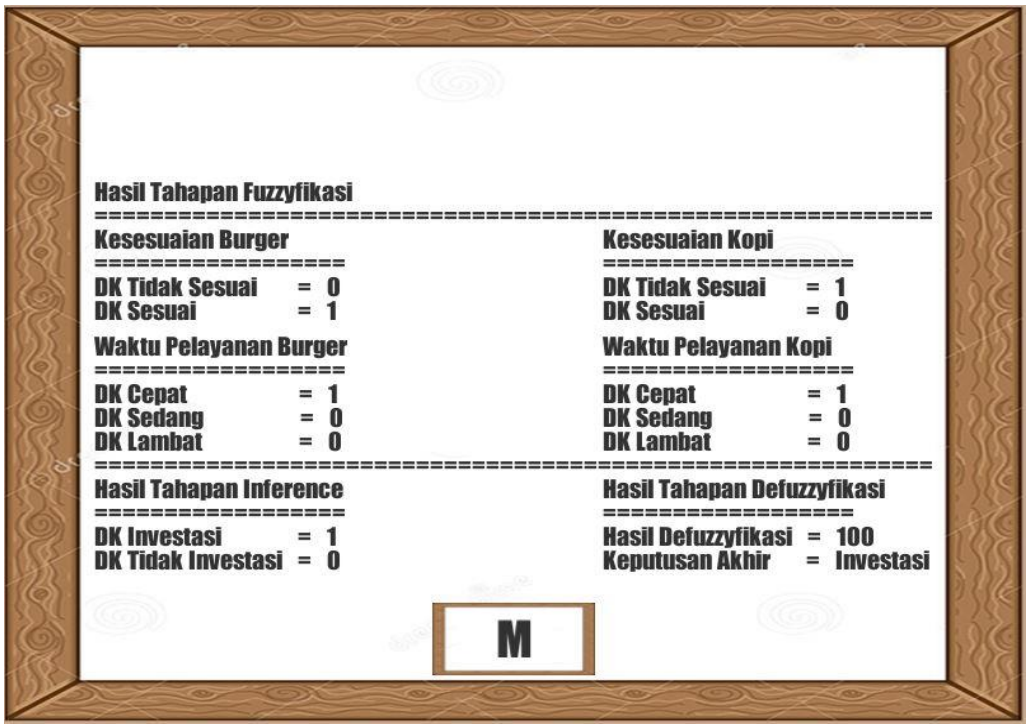
Gambar 4.7 Pengujian *Rule 5*

6. *Rule* 6



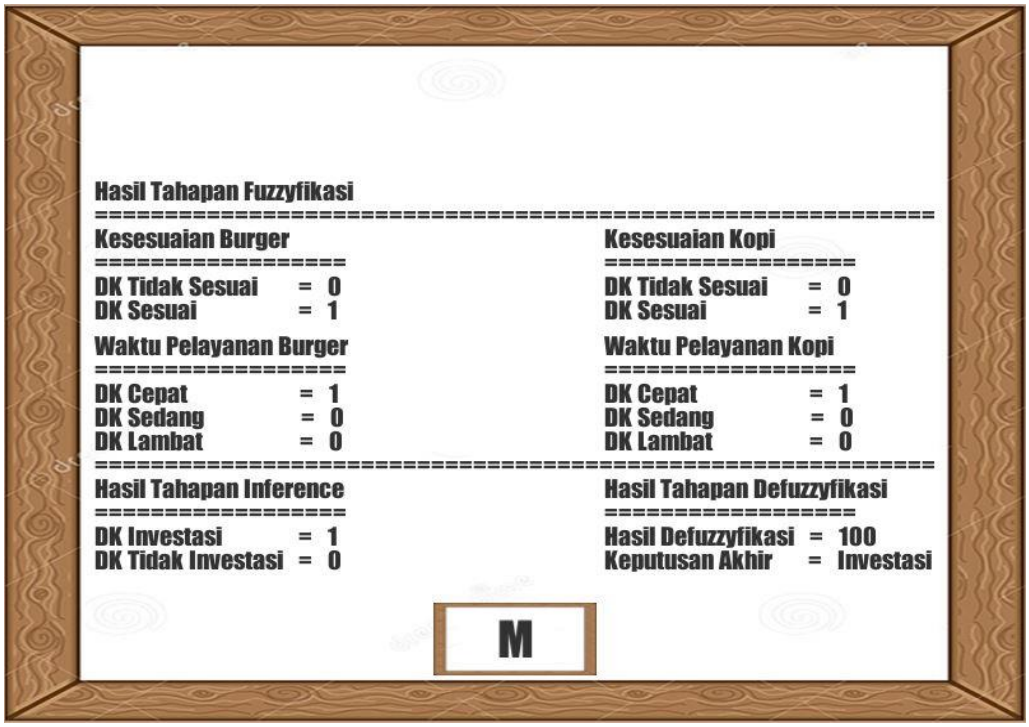
Gambar 4.8 Pengujian *Rule 6*

7. *Rule* 7



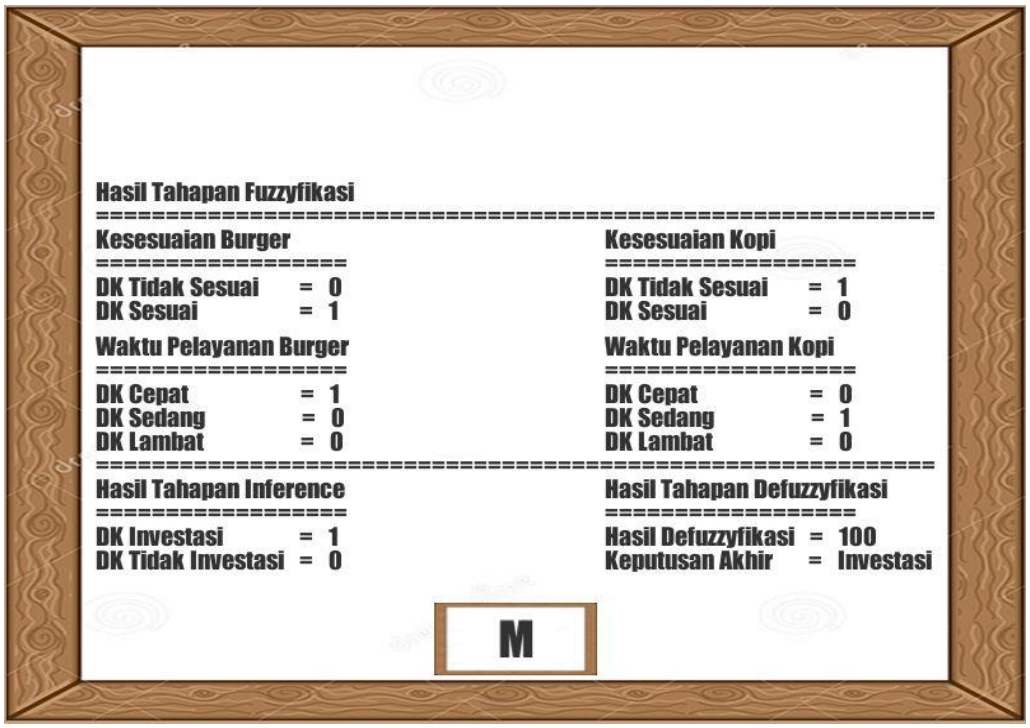
Gambar 4.9 Pengujian *Rule 7*

8. *Rule* 8



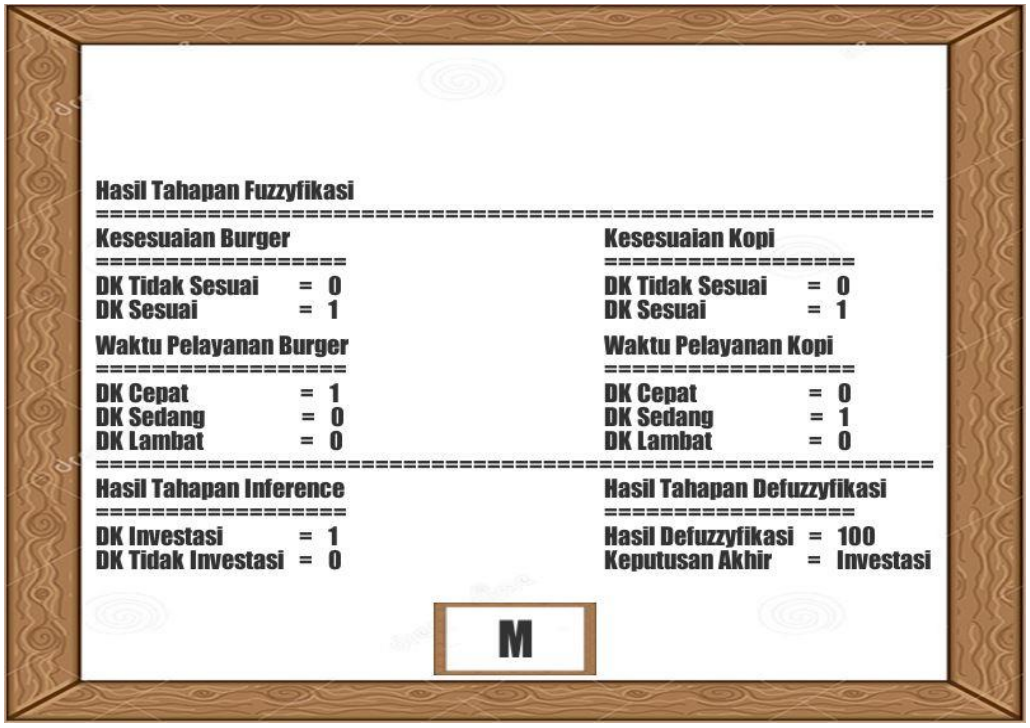
Gambar 4.10 Pengujian *Rule 8*

9. *Rule* 9



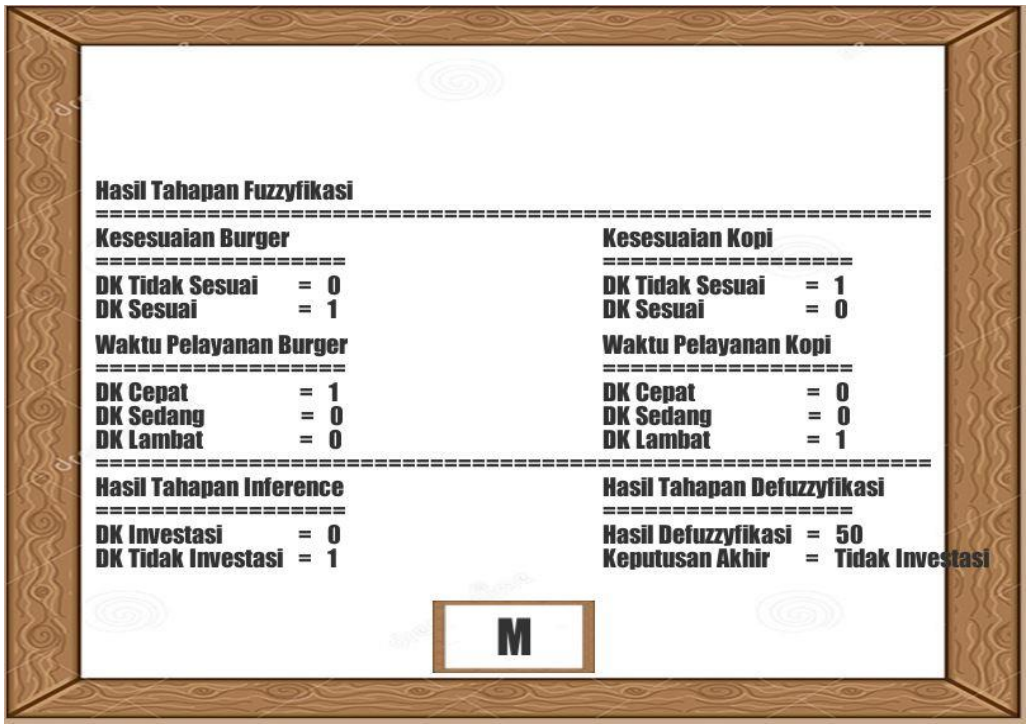
Gambar 4.11 Pengujian *Rule 9*

10. *Rule* 10



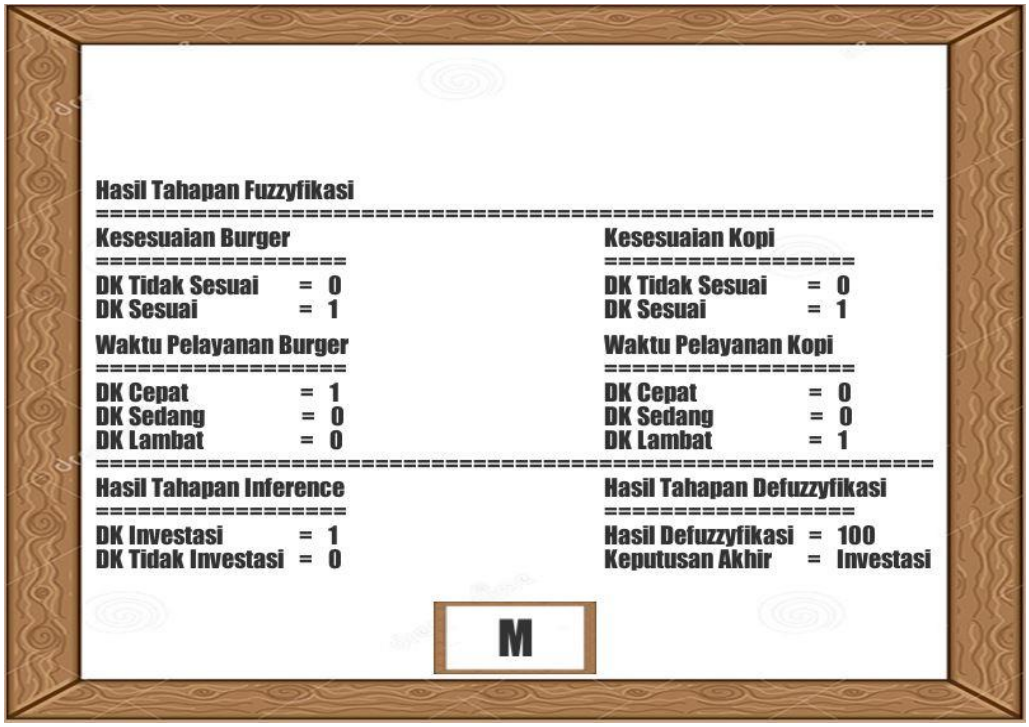
Gambar 4.12 Pengujian *Rule 10*

11. *Rule* 11



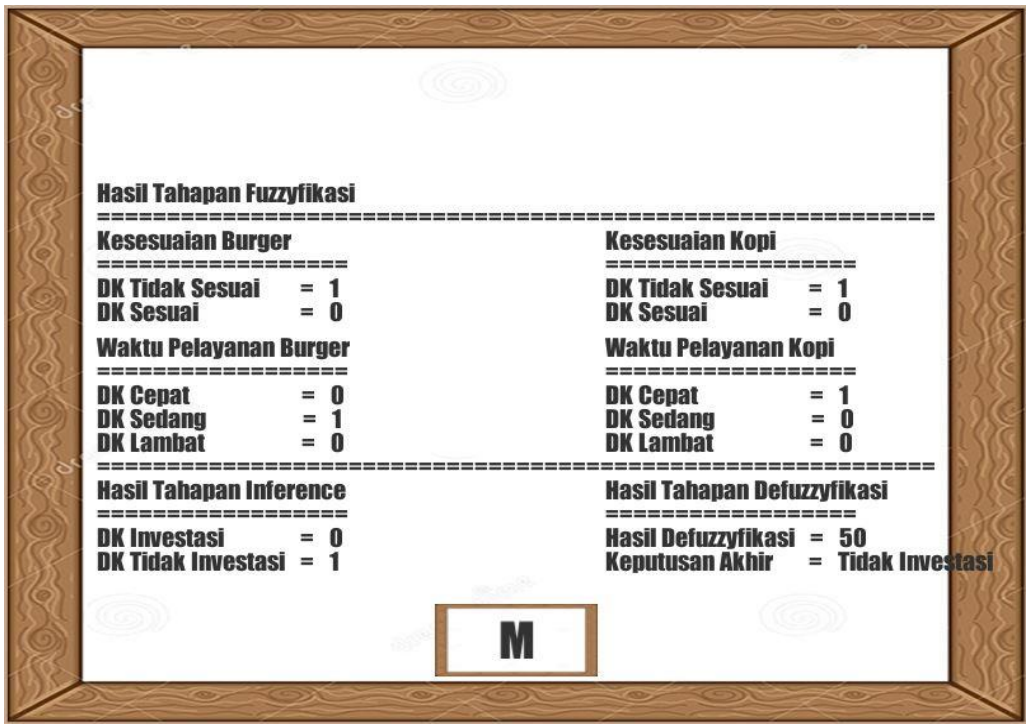
Gambar 4.13 Pengujian *Rule 11*

12. *Rule* 12



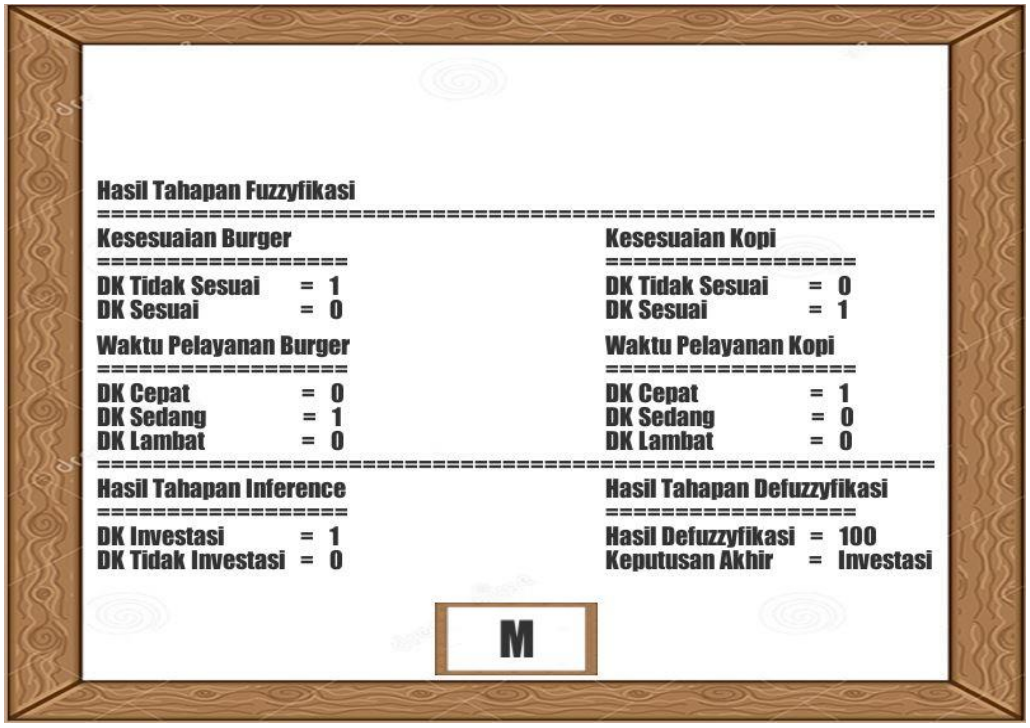
Gambar 4.14 Pengujian *Rule 12*

13. *Rule* 13



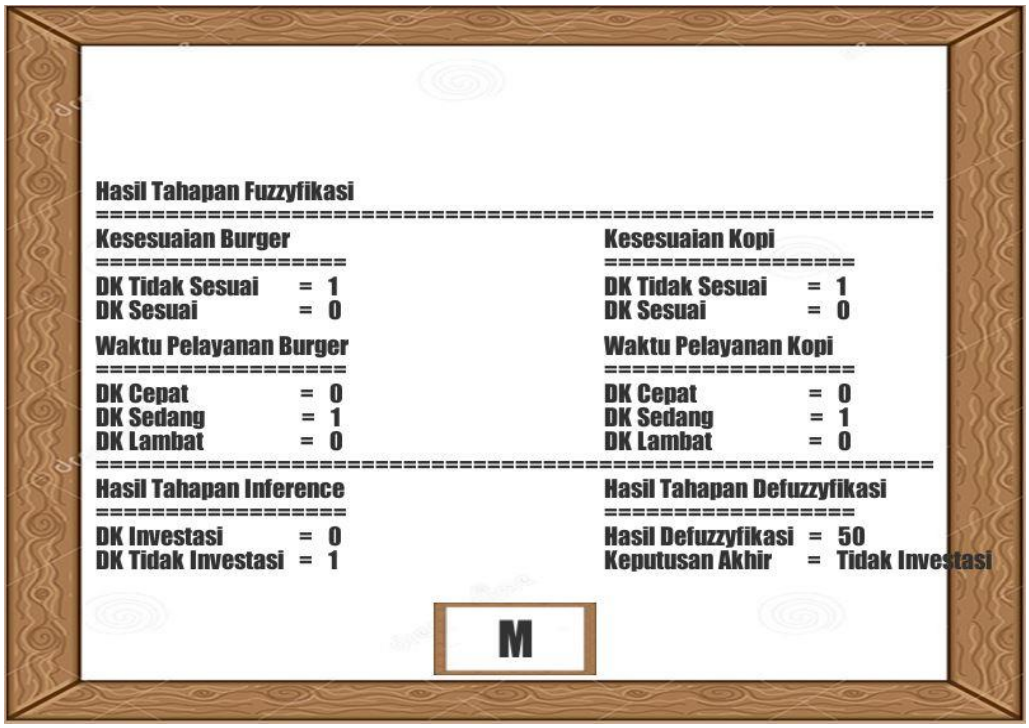
Gambar 4.15 Pengujian *Rule 13*

14. *Rule* 14



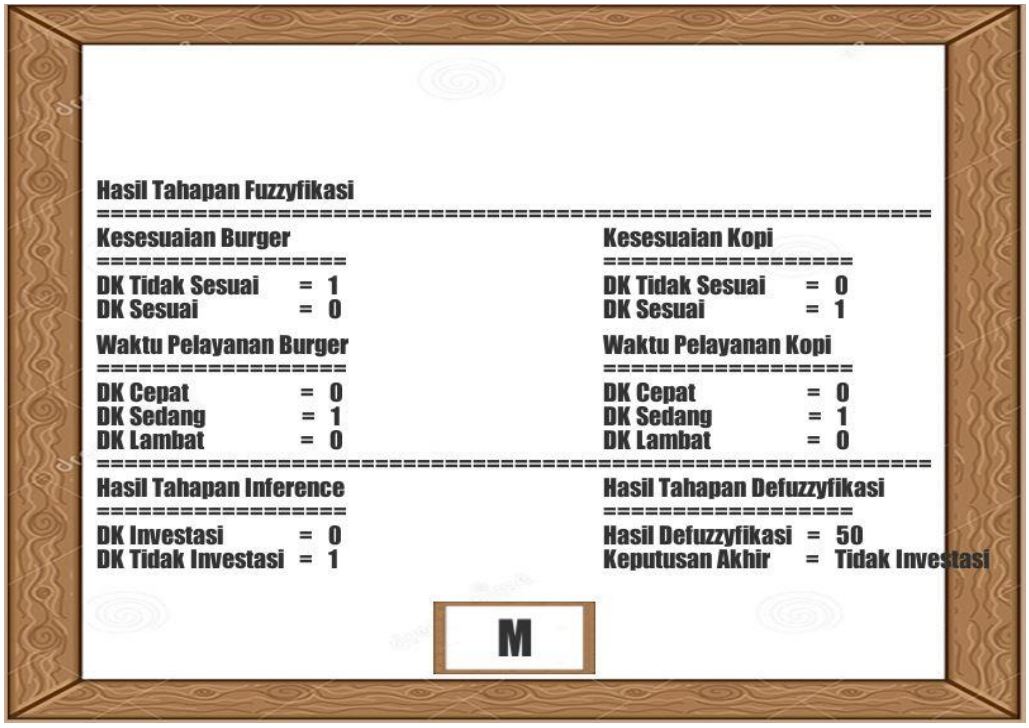
Gambar 4.16 Pengujian *Rule 14*

15. *Rule* 15



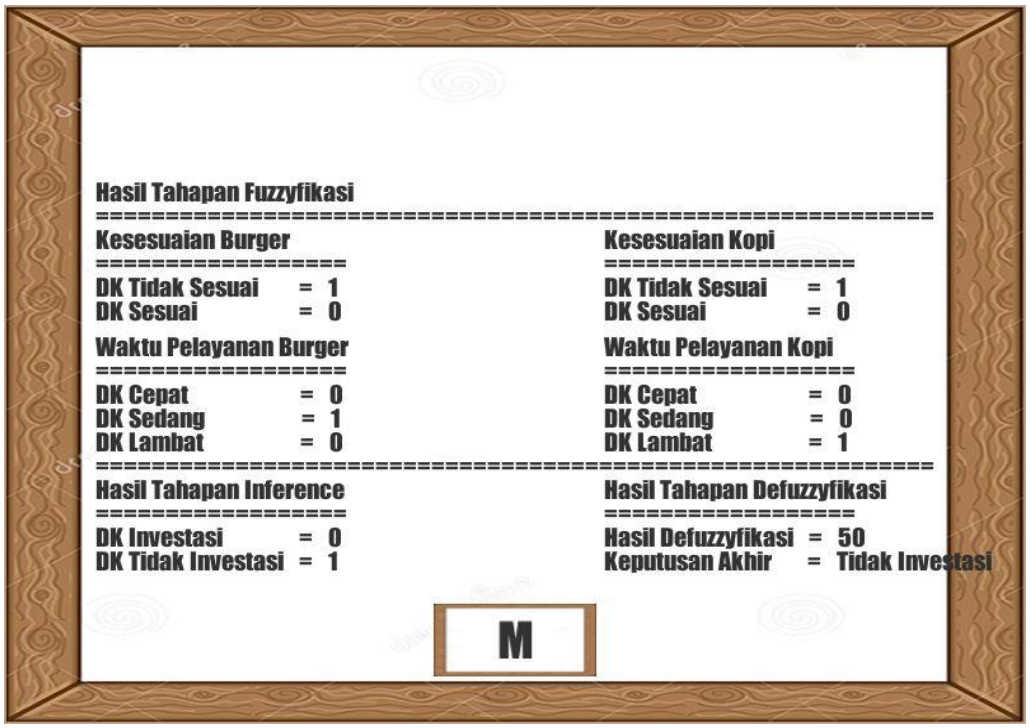
Gambar 4.17 Pengujian *Rule 15*

16. *Rule* 16



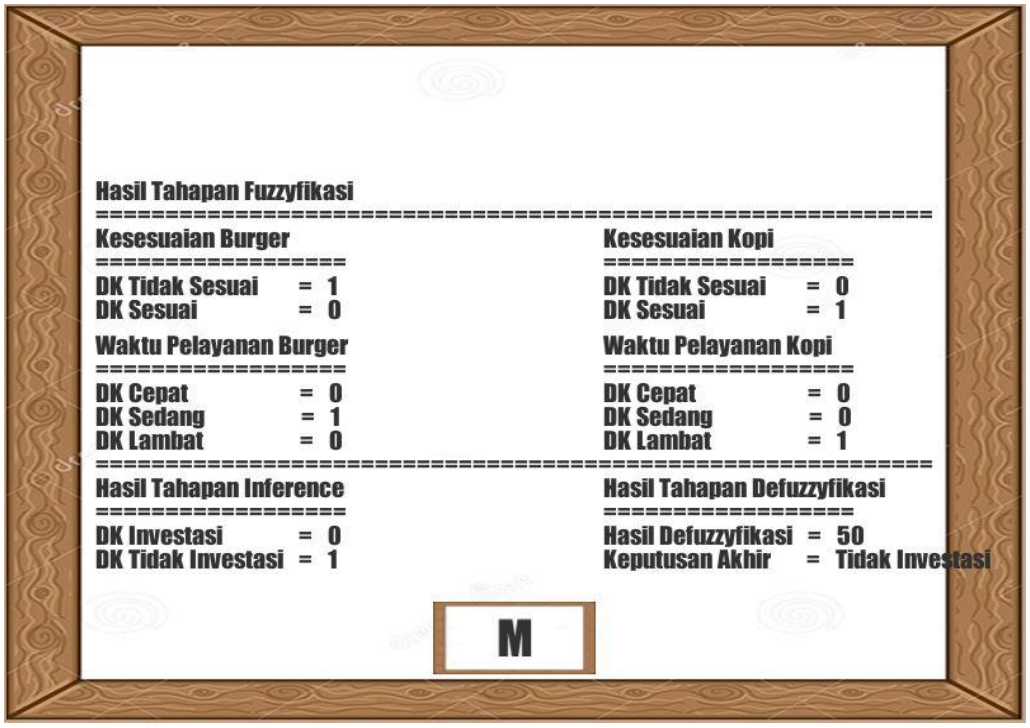
Gambar 4.18 Pengujian *Rule 16*

17. *Rule* 17



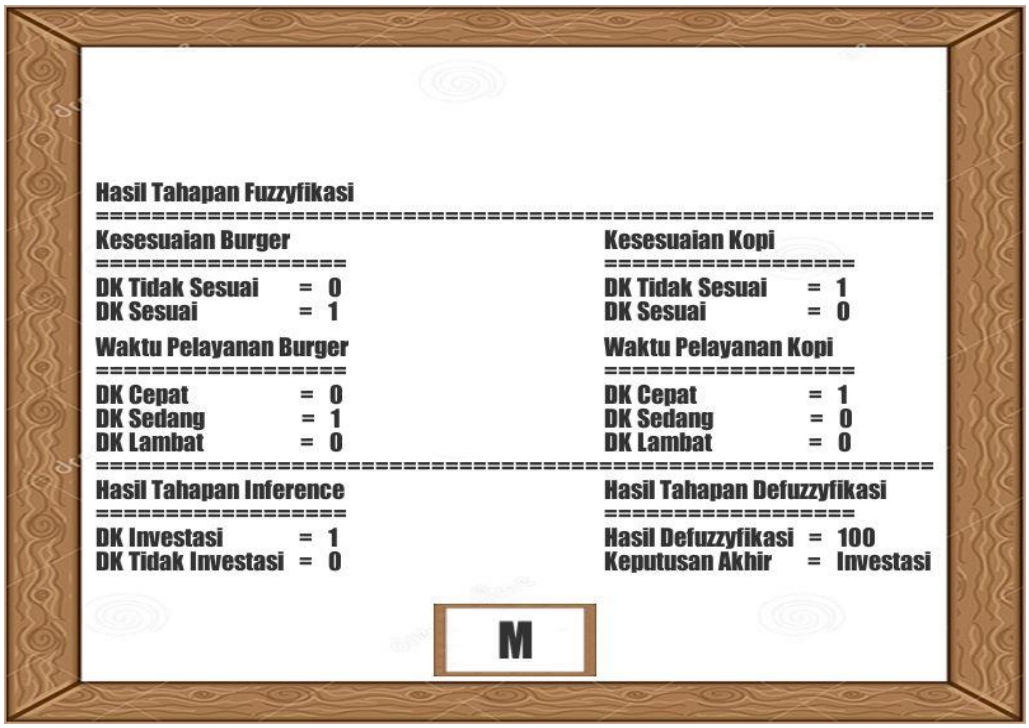
Gambar 4.19 Pengujian *Rule 17*

18. *Rule* 18



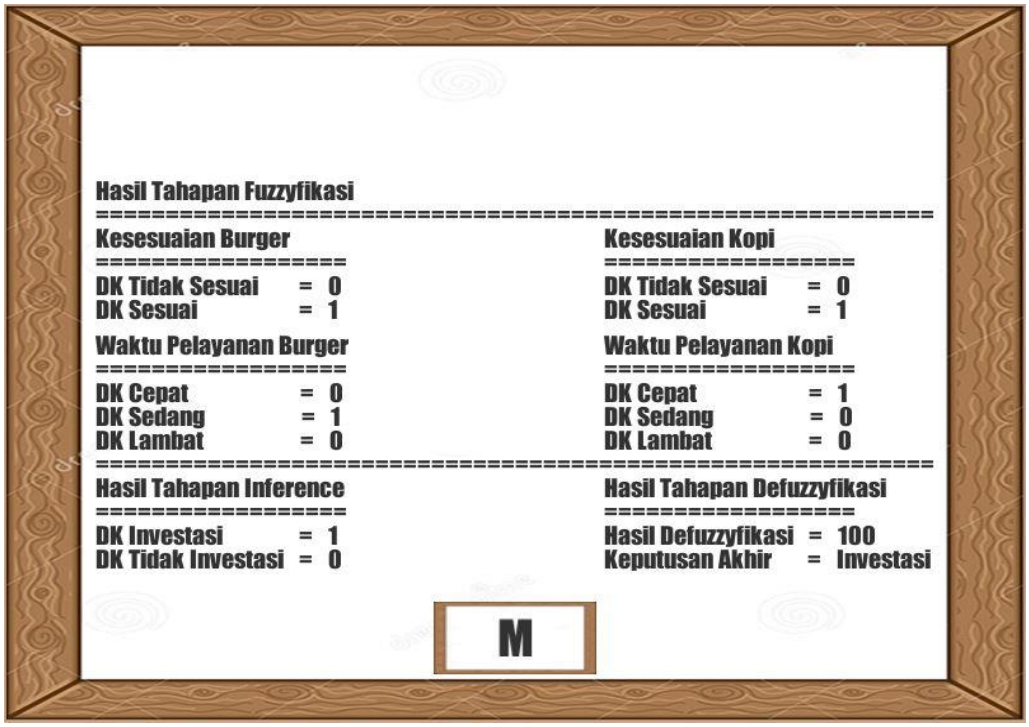
Gambar 4.20 Pengujian *Rule 18*

19. *Rule* 19



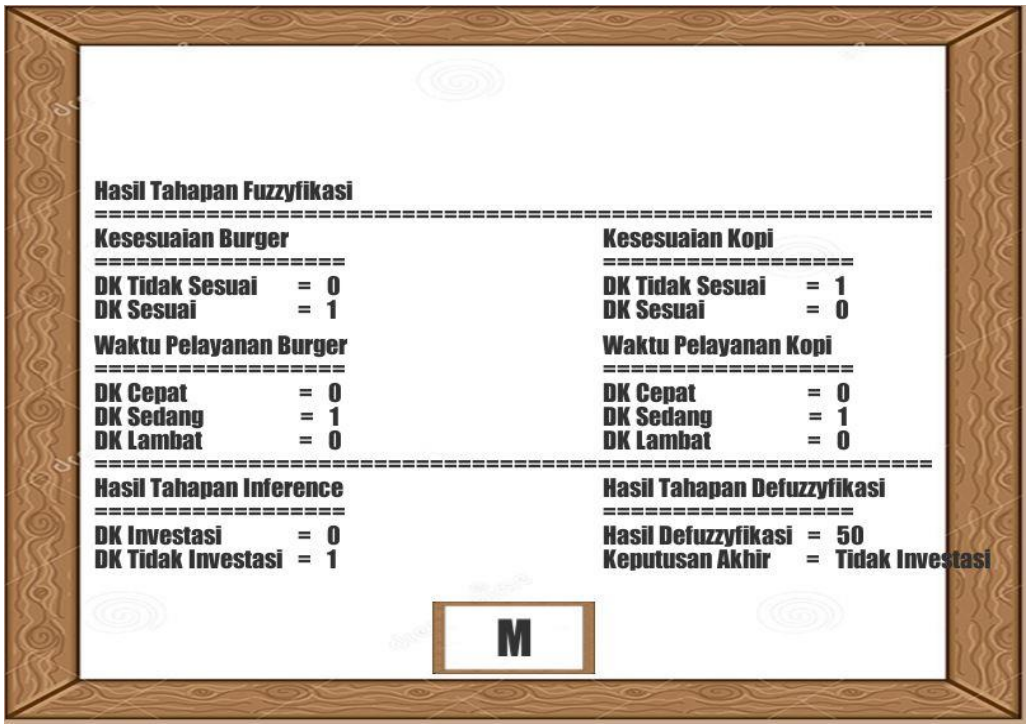
Gambar 4.21 Pengujian *Rule 19*

20. *Rule* 20



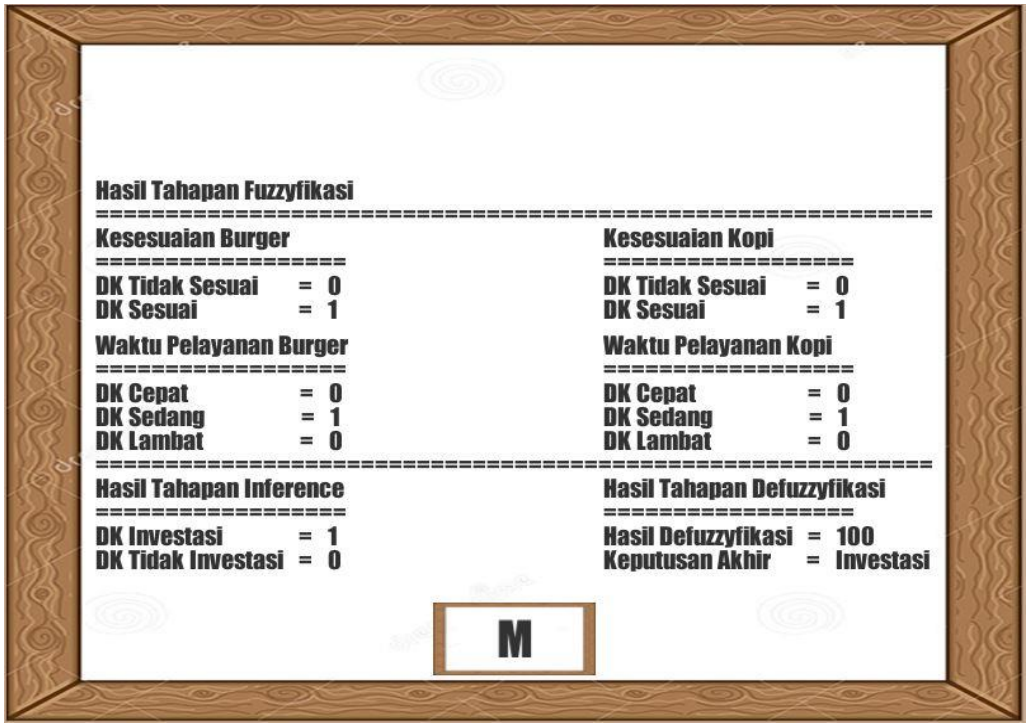
Gambar 4.22 Pengujian *Rule 20*

21. *Rule* 21



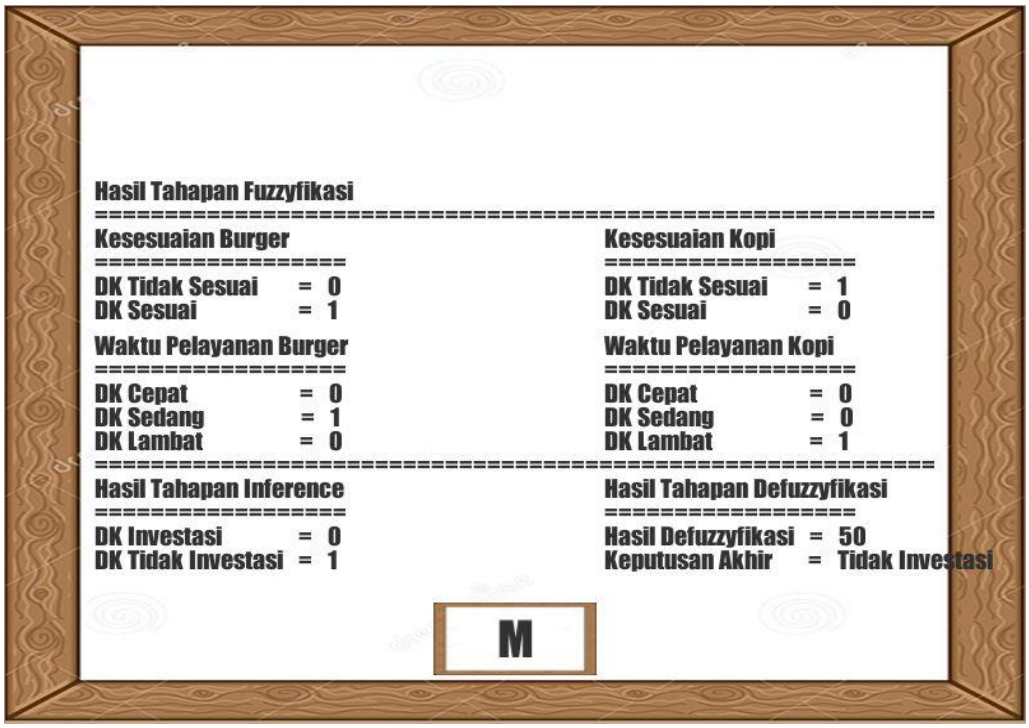
Gambar 4.23 Pengujian *Rule 21*

22. *Rule* 22



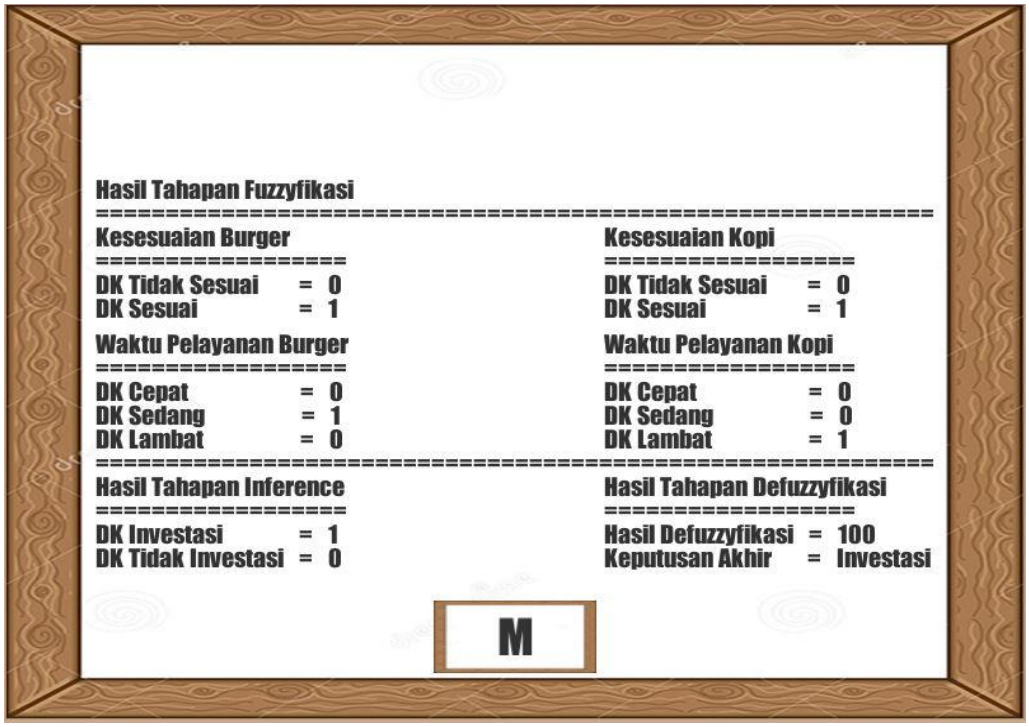
Gambar 4.24 Pengujian *Rule 22*

23. *Rule* 23



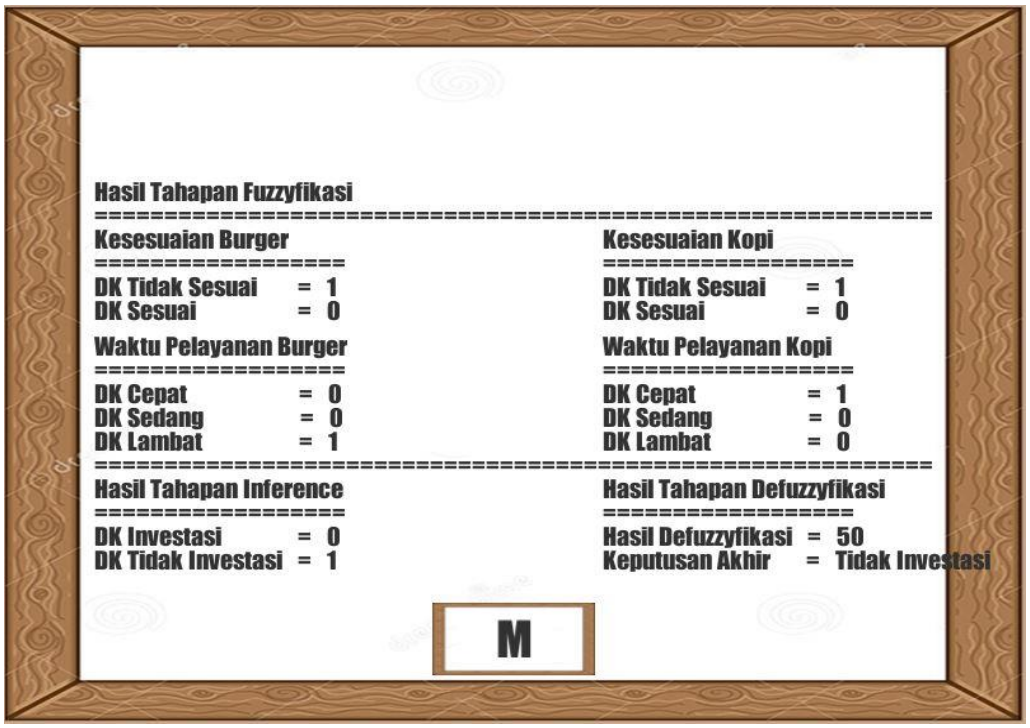
Gambar 4.25 Pengujian *Rule 23*

24. *Rule* 24



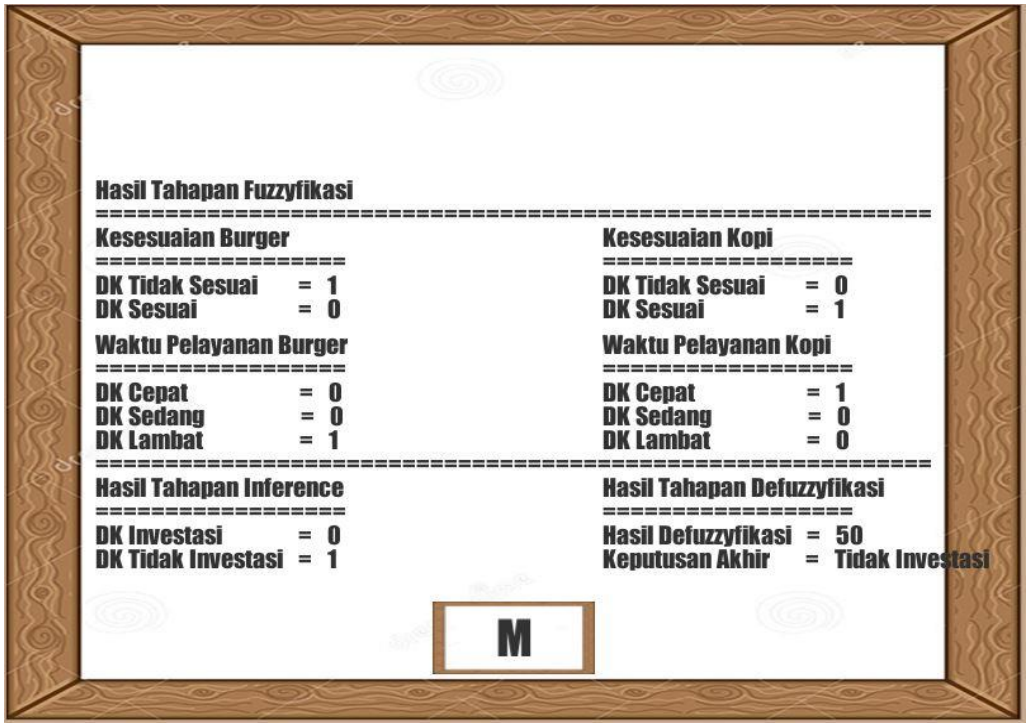
Gambar 4.26 Pengujian *Rule 24*

25. *Rule* 25



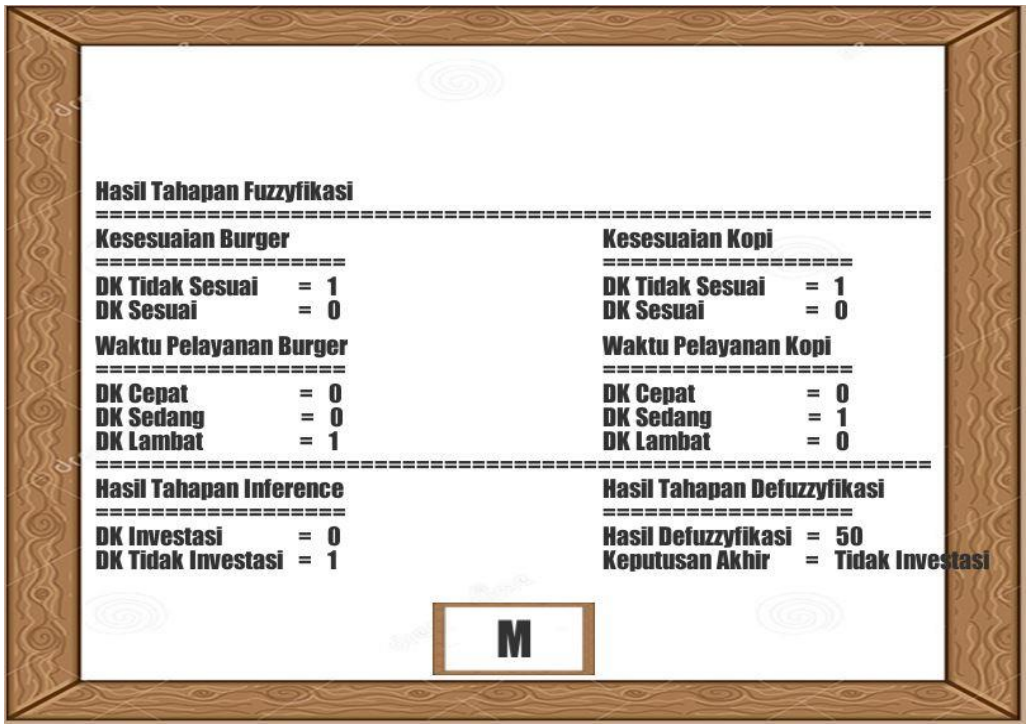
Gambar 4.27 Pengujian *Rule 25*

26. *Rule* 26



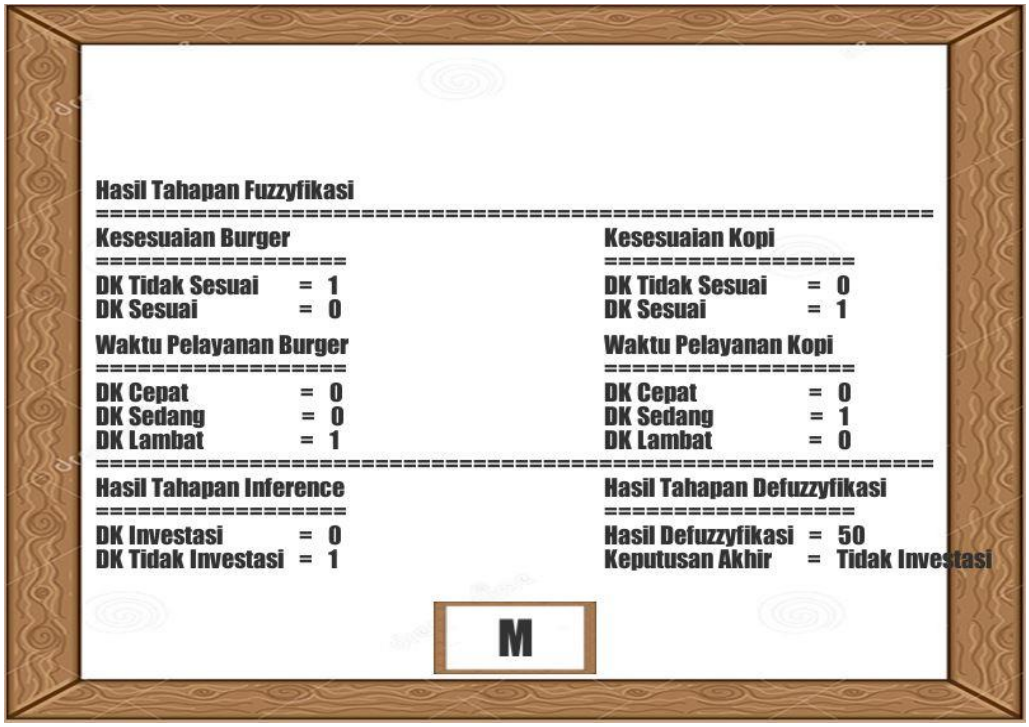
Gambar 4.28 Pengujian *Rule 26*

27. *Rule* 27



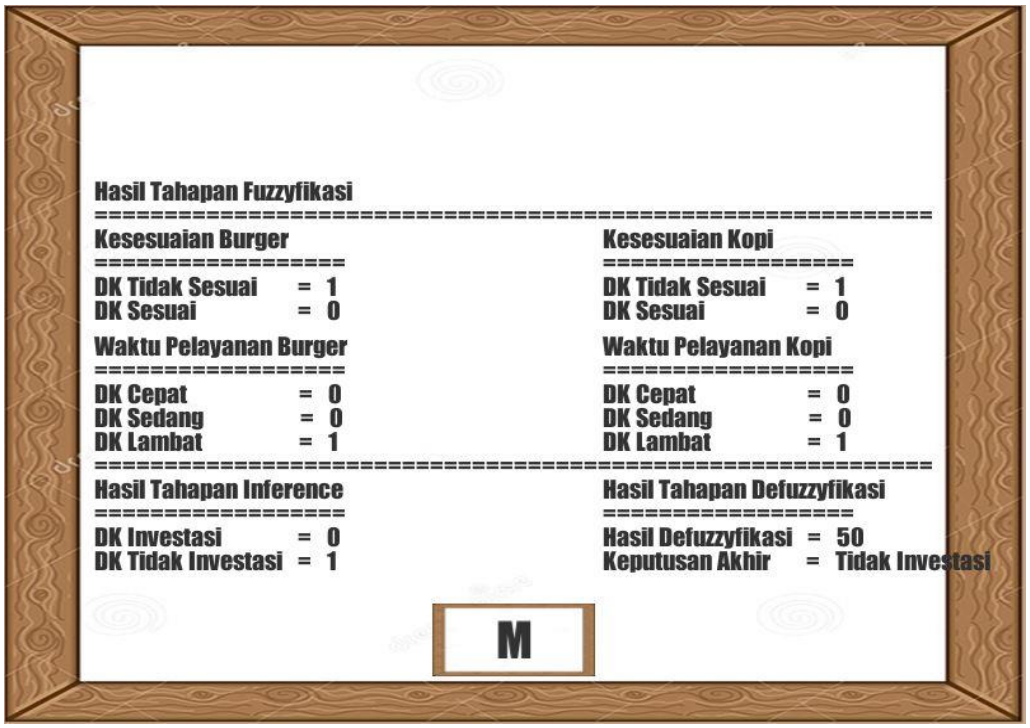
Gambar 4.29 Pengujian *Rule 27*

28. *Rule* 28



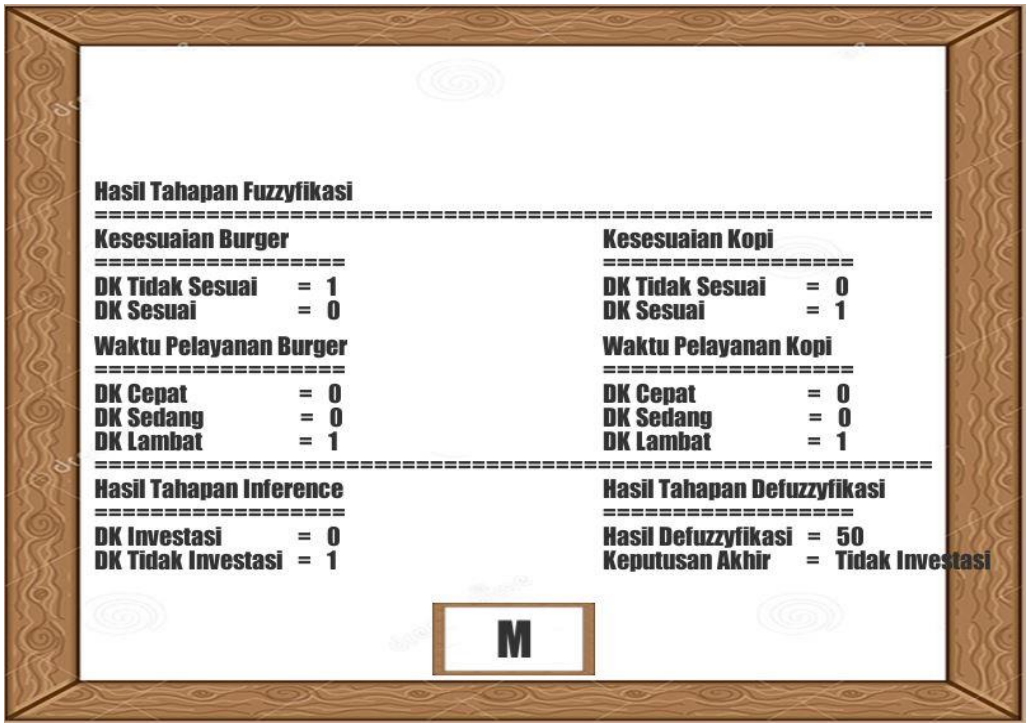
Gambar 4.30 Pengujian *Rule 28*

29. *Rule* 29



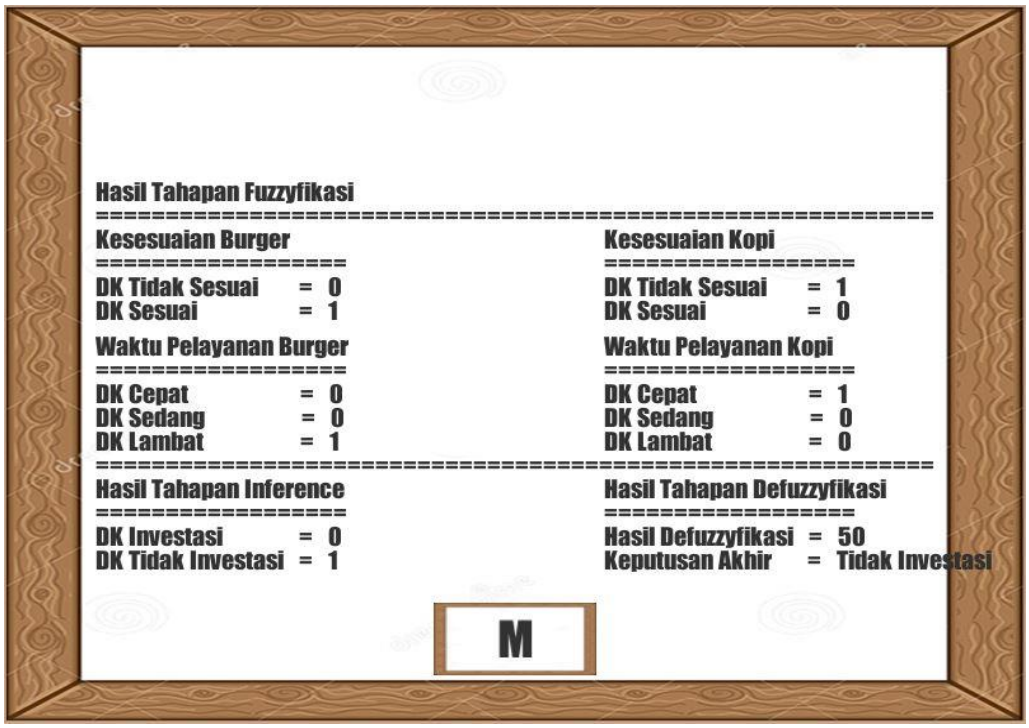
Gambar 4.31 Pengujian *Rule 29*

30. *Rule* 30



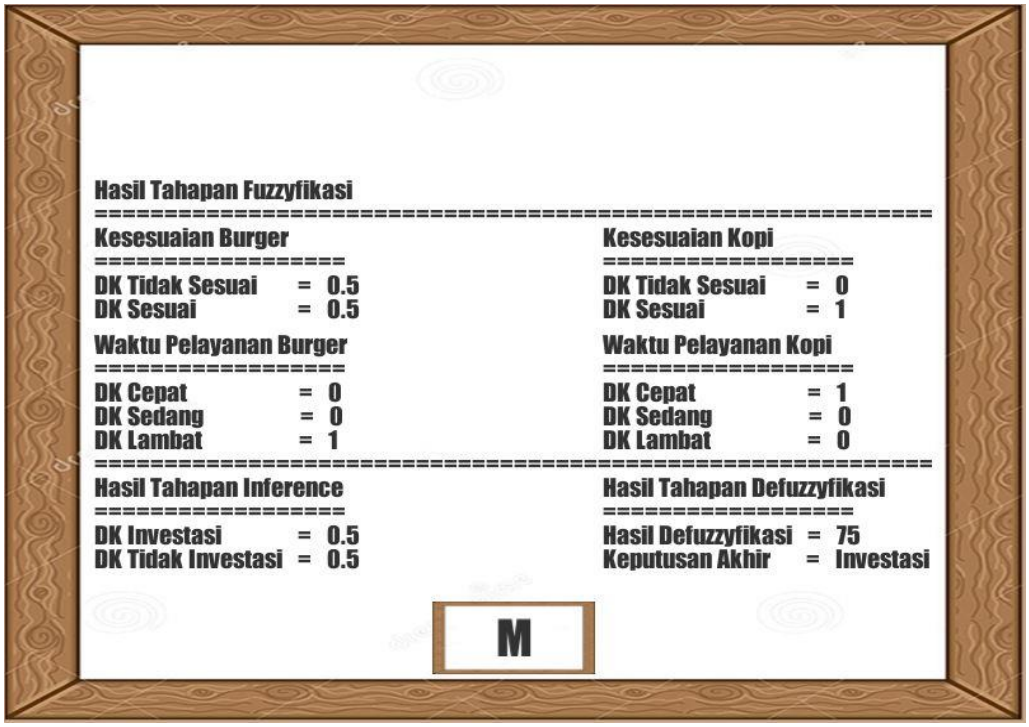
Gambar 4.32 Pengujian *Rule 30*

31. *Rule* 31



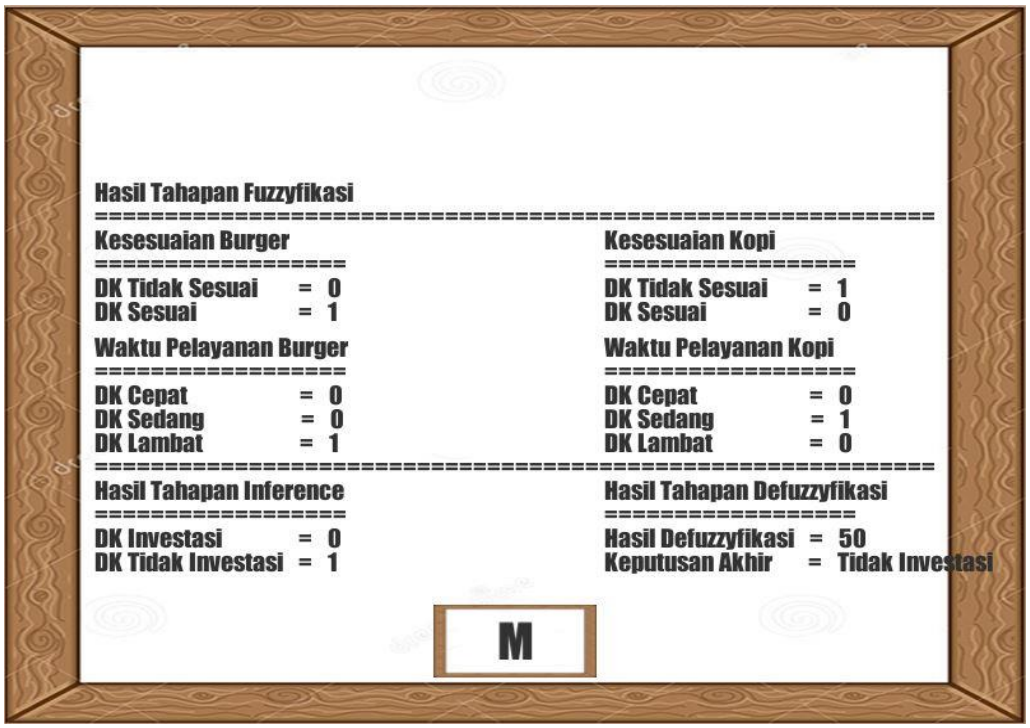
Gambar 4.33 Pengujian *Rule 31*

32. *Rule* 32



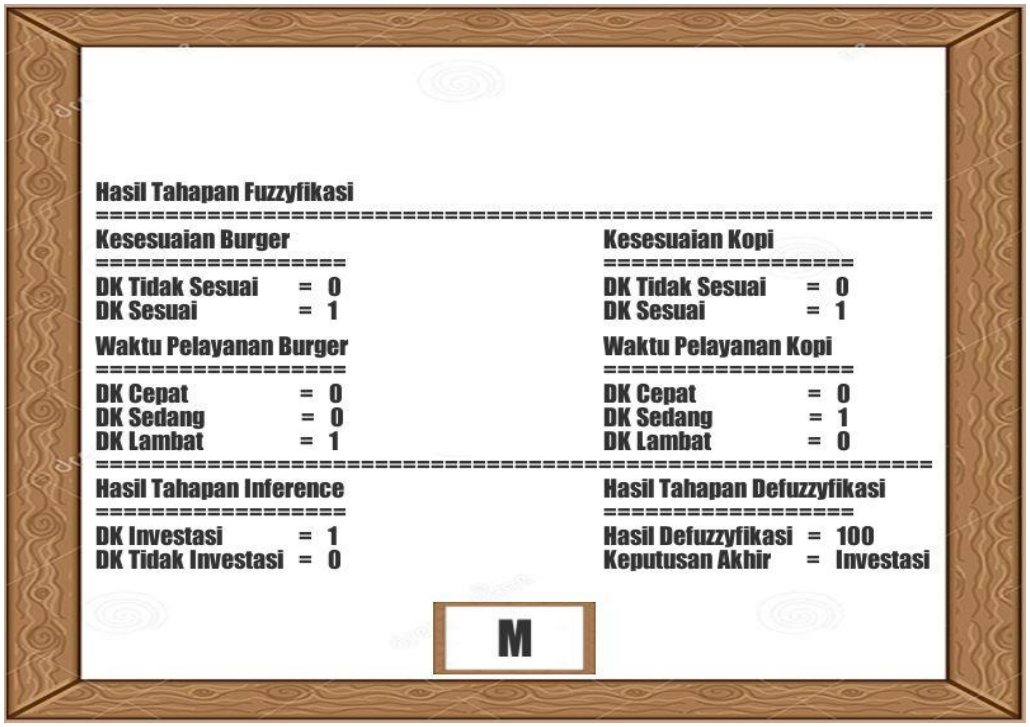
Gambar 4.34 Pengujian *Rule 32*

33. *Rule* 33



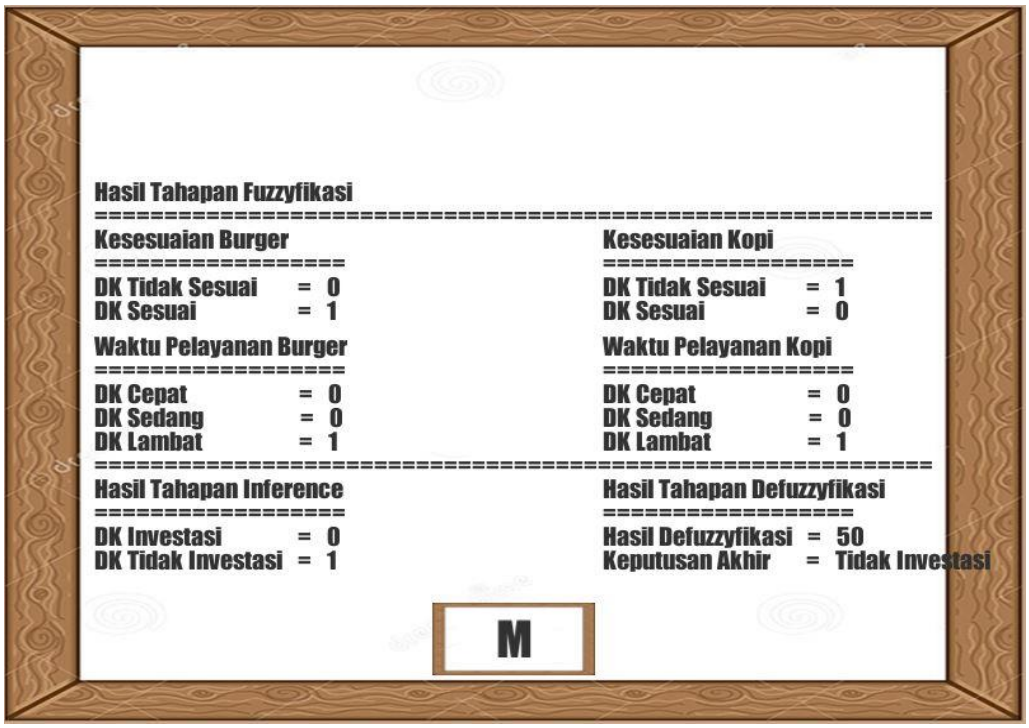
Gambar 4.35 Pengujian *Rule 33*

34. *Rule* 34



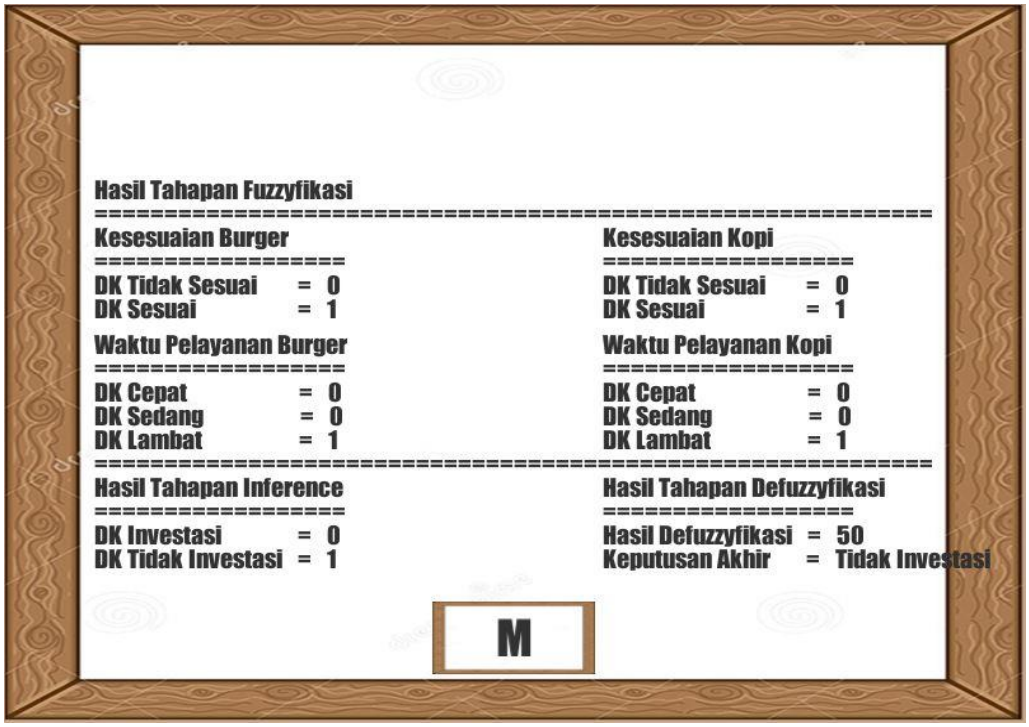
Gambar 4.36 Pengujian *Rule 34*

35. *Rule* 35



Gambar 4.37 Pengujian *Rule 35*

36. *Rule* 36



Gambar 4.38 Pengujian *Rule 36*

4.2.4 Pengujian Keputusan Akhir

Kondisi untuk pengujian hasil akhir yang akan diuji ada 3 yaitu ketika jumlah yang menyatakan investasi lebih banyak daripada jumlah yang menyatakan tidak investasi sehingga hasil akhir yang dikeluarkan akan menyatakan "Investasi". Juga ketika jumlah yang menyatakan tidak investasi lebih banyak dari jumlah yang menyatakan investasi sehingga hasil akhir yang dikeluarkan akan menyatakan "Tidak Investasi". Sedangkan untuk kondisi dimana jumlah yang menyatakan investasi sama dengan jumlah yang menyatakan tidak investasi akan mengeluarkan keputusan "Tidak Berinvestasi".

1. Nol pesanan berhasil menyatakan investasi dan delapan pesanan berhasil menyatakan tidak investasi



Gambar 4.39 Keputusan Akhir (1)

2. Satu pesanan berhasil menyatakan investasi dan tujuh pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.40 Keputusan Akhir (2)

3. Dua pesanan berhasil menyatakan investasi dan enam pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.41 Keputusan Akhir (3)

4. Tiga pesanan berhasil menyatakan investasi dan lima pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.42 Keputusan Akhir (4)

5. Empat pesanan berhasil menyatakan investasi dan empat pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.43 Keputusan Akhir (5)

6. Lima pesanan berhasil menyatakan investasi dan tiga pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.44 Keputusan Akhir (6)

7. Enam pesanan berhasil menyatakan investasi dan dua pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.45 Keputusan Akhir (7)

8. Tujuh pesanan berhasil menyatakan investasi dan satu pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.46 Keputusan Akhir (8)

9. Delapan pesanan berhasil menyatakan investasi dan nol pesanan menyatakan tidak investasi. Sehingga hasilnya sebagai berikut:



Gambar 4.47 Keputusan Akhir (9)

4.2.5 Kesimpulan Pengujian

Dari hasil pengujian Implementasi Logika *Fuzzy* didapatkan kesimpulan bahwa hasil pengujian sesuai dengan *rule* yang ada dalam sistem *fuzzy* sehingga keputusan akhir yang dihasilkan sesuai dengan pengetahuan yang ditanamkan pada *game*.