

**IMPLEMENTASI *DYNAMIC CHALLENGING LEVEL ADAPTER*
(DCA) MENGGUNAKAN *FUZZY SUGENO* PADA LEVEL
GAME ADVENTURE GUNUNG KELUD UNTUK NPC**

SKRIPSI

Oleh :
LISA AULIA UMAMI
NIM. 16650023



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**IMPLEMENTASI *DYNAMIC CHALLENGING LEVEL ADAPTER*
(DCA) MENGGUNAKAN *FUZZY SUGENO* PADA LEVEL
GAME ADVENTURE GUNUNG KELUD UNTUK NPC**

SKRIPSI

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :
LISA AULIA UMAMI
NIM. 16650023**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN

**IMPLEMENTASI *DYNAMIC CHALLENGING LEVEL ADAPTER*
(DCA) MENGGUNAKAN *FUZZY SUGENO* PADA LEVEL
GAME ADVENTURE GUNUNG KELUD UNTUK NPC**

SKRIPSI

Oleh :
LISA AULIA UMAMI
NIM. 16650023

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : 08 Desember 2020

Dosen Pembimbing I

Dosen Pembimbing II

Fresy Nugroho, M.T
NIP. 19710722 201101 1 001

M. Imamuddin, Lc., MA
NIP. 19740602 200901 1 010

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

IMPLEMENTASI *DYNAMIC CHALLENGING LEVEL ADAPTER* (DCA) MENGGUNAKAN *FUZZY SUGENO* PADA LEVEL *GAME ADVENTURE GUNUNG KELUD* UNTUK NPC

SKRIPSI

Oleh:
LISA AULIA UMAMI
NIM. 16650023

Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal 21 Desember 2020

Susunan Penguji

| | | Tanda tangan |
|-----------------------|--|--------------|
| 1. Penguji Utama | <u>Hani Nurhayati, M.T</u> : NIP. 19780625 200801 2 006 | () |
| 2. Ketua Penguji | <u>Fachrul Kurniawan, M.MT</u> : NIP. 19771020 200912 1 001 | () |
| 3. Sekretaris Penguji | <u>Fresy Nugroho, M.T</u> : NIP. 19710722 201101 1 001 | () |
| 4. Anggota Penguji | <u>M. Imamuddin, Lc., MA</u> : NIP. 19740602 200901 1 010 | () |

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Lisa Aulia Umami

NIM : 16650023

Fakultas/jurusan : Sains dan Teknologi/Teknik Informatika

Judul Skripsi : Implementasi *Dynamic Challenging Level Adapter* (DCA)
Menggunakan *Fuzzy Sugeno* Pada Level *Game Adventure*
Gunung Kelud untuk NPC

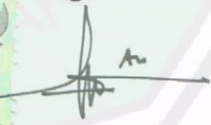
Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 28 Desember 2020

Yang membuat pernyataan,




Lisa Aulia Umami

NIM. 16650023

HALAMAN MOTO



HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Skripsi ini telah selesai meskipun tidak tepat waktu, namun cukup melegakan bagi Penulis. Penulis persembahkan skripsi ini kepada:

Saya sendiri yang sangat berjasa dan selalu berusaha sekuat tenaga memberikan dorongan lahir batin untuk tetap semangat dalam menyelesaikan skripsi sampai dititik pencapaian sekarang ini.

Kedua orang tua tercinta dan tersayang, Bapak Mustofa dan Ibu Mudayaroh, saudara perempuan Tatimul Fajriyah, dan saudara laki-laki M. Ali Aviansyah, maupun keluarga besar lainnya yang selalu memanjatkan do'a dan memberikan motivasi untuk segera menyelesaikan skripsi ini.

Dosen pembimbing, Bapak Fresy Nugroho, M.T dan Bapak M. Imamuddin, Lc., MA, yang telah memberikan bimbingan dan saran sehingga penelitian ini bisa berjalan dengan lancar.

Seluruh dosen Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang telah dengan ikhlas membagi ilmu dan berbagai pengalamannya.

Sahabat-sahabat saya yang sangat banyak dan tidak bisa disebutkan namanya satu per satu, saudara Andromeda Teknik Informatika 2016, keluarga besar Teknik Informatika UIN Maulana Malik Ibrahim Malang, yang tiada hentinya memberikan semangat untuk menyelesaikan skripsi ini.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan yang Maha Esa yang telah memberikan berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi *Dynamic Challenging Level Adapter (DCA)* Menggunakan *Fuzzy Sugeno* Pada *Level Game Adventure* Gunung Kelud untuk NPC”. Tujuan dari penyusunan skripsi ini guna memenuhi salah satu syarat untuk bisa menempuh ujian sarjana komputer pada Fakultas Sains dan Teknologi Program Studi Teknik Informatika di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Dalam penyusunan skripsi penulis banyak mendapatkan bantuan ataupun masukan dari berbagai pihak. Oleh sebab itu, penulis mengungkapkan rasa terima kasih kepada:

1. Bapak Prof. Dr. Abdul Haris, M.Ag., selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Ibu Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi.
3. Bapak Dr. Cahyo Crysdiyan, selaku Ketua jurusan Teknik Informatika, Fakultas Sains dan Teknologi UIN Maliki Malang.
4. Bapak Syahiduz Zaman, M.Kom, selaku Wali Dosen yang telah memberikan saran dan masukan hingga saya bisa menyelesaikan perkuliahan dengan baik.
5. Bapak Fresy Nugroho, M. T, selaku pembimbing I, yang telah memberikan bimbingan dan masukan sehingga penulis dapat menyelesaikan penelitian ini.
6. Bapak M. Imamuddin, Lc., MA, selaku pembimbing II, yang telah memberikan bimbingan dan masukan sehingga penulis dapat menyelesaikan penelitian ini.
7. Mbak Citra Fidya Atmalia, S.H dan para staff laboran, yang telah membantu dalam urusan administrasi.
8. Orang tua dan keluarga penulis yang telah memanjatkan do’a dan memberikan dukungan dan motivasi.
9. Kakak dan adik tercinta juga anggota keluarga dan kerabat yang senantiasa memberikan doa dan dukungan semangat kepada penulis.
10. Sahabat-sahabat seperjuangan yang tiada henti memberi dukungan dan motivasi kepada penulis serta target bersama untuk lulus skripsi dan wisuda bersama

11. Rekan-rekan dan saudara Andromeda TI 16, kakak tingkat, dan adik tingkat Teknik Informatika UIN Maulana Malik Ibrahim Malang.

12. Semua pihak yang terlibat dalam penyusunan skripsi.

Penulis menyadari bahwa skripsi ini masih terdapat banyak kekurangan, namun penulis berharap skripsi ini dapat bermanfaat khususnya bagi penulis secara pribadi.

Malang, 28 Desember 2020

Penulis



DAFTAR ISI

| | |
|--|------------|
| HALAMAN PENGAJUAN | i |
| LEMBAR PERSETUJUAN | ii |
| LEMBAR PENGESAHAN | iii |
| PERNYATAAN KEASLIAN TULISAN | iv |
| HALAMAN MOTO | v |
| HALAMAN PERSEMBAHAN | vi |
| KATA PENGANTAR..... | vii |
| DAFTAR ISI..... | ix |
| DAFTAR GAMBAR..... | xi |
| DAFTAR TABEL | xii |
| ABSTRAK | xiv |
| ABSTRACT | xv |
| المخلص | xvi |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Identifikasi Masalah..... | 6 |
| 1.3 Batasan Masalah | 7 |
| 1.4 Tujuan Penelitian | 7 |
| 1.5 Manfaat Penelitian | 8 |
| BAB II TINJAUAN PUSTAKA..... | 9 |
| 2.1 Penelitian Terkait | 9 |
| 2.2 Landasan Teori..... | 10 |
| 2.2.1 <i>Game adventure</i> | 10 |
| 2.2.2 <i>Non-Player Character</i> | 12 |
| 2.2.3 Fitur Kabut..... | 14 |
| 2.2.4 <i>Fuzzy Sugeno</i> | 16 |
| 2.2.5 <i>Dynamic Challenging Level Adapter (DCA)</i> | 17 |
| BAB III METODOLOGI PENELITIAN | 19 |
| 3.1 Fokus Penelitian | 19 |
| 3.2 Desain Sistem <i>Game</i> | 20 |
| 3.3 Desain Sistem DCA pada <i>Game</i> Menggunakan <i>Fuzzy Sugeno</i> | 22 |
| 3.4 Skenario <i>Game</i> | 23 |

| | | |
|--|--|------------|
| 3.4.1 | Storyline <i>Game</i> | 23 |
| 3.5 | <i>Dynamic Challenging Level Adapter</i> | 24 |
| 3.6 | DCA Menggunakan <i>Fuzzy Sugeno</i> | 25 |
| 3.6.1 | Penentuan Nilai <i>Input Level</i> | 25 |
| 3.6.2 | Perancangan <i>Fuzzy</i> | 26 |
| 3.6.2.1 | <i>Fungsi Keanggotaan</i> | 27 |
| 3.6.2.2 | <i>Fuzzyfikasi</i> | 30 |
| 3.6.2.3 | <i>Fuzzy Rules</i> | 32 |
| 3.6.2.4 | <i>Implikasi dan Defuzzyfikasi</i> | 36 |
| 3.7 | Contoh Penentuan Level DCA Menggunakan <i>Fuzzy Sugeno</i> | 37 |
| 3.8 | Pengujian dan Perhitungan Akurasi, Presisi, <i>Recall</i> , <i>F-Measure</i> | 40 |
| BAB IV HASIL DAN PEMBAHASAN | | 49 |
| 4.1 | Implementasi | 49 |
| 4.2 | Implementasi Tampilan <i>Game</i> | 49 |
| 4.2.1 | Tampilan Main Menu <i>Game</i> | 49 |
| 4.2.2 | Tampilan Awal <i>Game</i> | 50 |
| 4.2.3 | Tampilan Panel Variabel Dalam <i>Game</i> | 51 |
| 4.2.4 | Tampilan Perubahan Level Kabut | 51 |
| 4.2.5 | Tampilan <i>Canvas</i> Pada <i>Button Setting</i> | 52 |
| 4.2.6 | Tampilan <i>Canvas Input</i> Nilai Variabel <i>Game</i> | 53 |
| 4.2.7 | Tampilan <i>Canvas Game Win</i> | 54 |
| 4.2.8 | Tampilan <i>Canvas Game Lose</i> | 55 |
| 4.3 | Tahap Pengujian | 55 |
| 4.3.1 | Pembuatan Sistem Level DCA | 56 |
| 4.3.2 | Objek Pendukung Perubahan Level Pada Sistem <i>Game</i> | 57 |
| 4.4 | Hasil Pengujian | 58 |
| 4.5 | Pembahasan | 111 |
| 4.6 | Integrasi Islam | 113 |
| BAB V KESIMPULAN DAN SARAN | | 116 |
| 5.1 | Kesimpulan | 116 |
| 5.2 | Saran | 117 |
| DAFTAR PUSTAKA | | 118 |
| LAMPIRAN | | 120 |

DAFTAR GAMBAR

| | |
|---|-----|
| Gambar 2.1 Contoh penggambaran NPC | 12 |
| Gambar 2.2 Ilustrasi Navmesh | 13 |
| Gambar 2.3 Blok diagram logika <i>fuzzy</i> | 16 |
| Gambar 2.4 Blok diagram alur DCA menggunakan <i>Fuzzy Sugeno</i> | 18 |
| Gambar 3.1 Blok diagram prosedur penelitian..... | 19 |
| Gambar 3.2 Roadmap Penelitian | 20 |
| Gambar 3.3 Desain sistem <i>game</i> pada penelitian sebelumnya..... | 21 |
| Gambar 3.4 Desain sistem <i>game</i> yang akan dibuat | 21 |
| Gambar 3.5 Desain sistem <i>game</i> DCA menggunakan <i>Fuzzy Sugeno</i> | 22 |
| Gambar 3.6 Diagram alir <i>fuzzy</i> pada <i>game</i> | 27 |
| Gambar 3.7 Fungsi keanggotaan <i>input score</i> | 30 |
| Gambar 3.8 Fungsi keanggotaan <i>input time remain</i> | 31 |
| Gambar 3.9 Fungsi keanggotaan <i>input player_life</i> | 32 |
| Gambar 3.10 Contoh Ilustrasi <i>Confusion Matrix</i> | 44 |
| Gambar 4.1 Tampilan awal <i>main menu</i> pada <i>game</i> | 49 |
| Gambar 4.2 Tampilan awal permainan..... | 50 |
| Gambar 4.3 Panel variabel di dalam <i>game</i> | 51 |
| Gambar 4.4 Tampilan perubahan level kabut..... | 52 |
| Gambar 4.5 Tampilan <i>canvas</i> pada <i>button setting</i> | 53 |
| Gambar 4.6 Tampilan <i>canvas input</i> nilai variabel <i>DCA Fuzzy</i> | 53 |
| Gambar 4.7 Tampilan <i>canvas game win</i> | 54 |
| Gambar 4.8 Tampilan <i>canvas lose game</i> | 55 |
| Gambar 4.9 <i>Panel Script</i> sistem level DCA pada <i>Unity</i> | 56 |
| Gambar 4.10 <i>Objek</i> pendukung perubahan level..... | 57 |
| Gambar 4.11 Grafik hasil pengujian percobaan | 112 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Tingkat visibilitas internasional dan koefisien hamburan..... | 14 |
| Tabel 2.2 Pembagian level kabut di dalam <i>game</i> | 15 |
| Tabel 3.1 Pengujian DCA dengan nilai CR tiap level..... | 25 |
| Tabel 3.2 Pembagian nilai <i>input</i> level | 26 |
| Tabel 3.3 Nilai linguistik <i>input</i> variabel <i>score</i> | 28 |
| Tabel 3.4 Nilai linguistik <i>input</i> variabel <i>time remain</i> | 28 |
| Tabel 3.5 Nilai linguistik <i>input</i> <i>player life</i> | 29 |
| Tabel 3.6 Nilai linguistik <i>output</i> variabel kabut..... | 29 |
| Tabel 3.7 <i>Rules fuzzy</i> penentuan <i>output</i> | 33 |
| Tabel 3.8 Contoh Kombinasi Variabel <i>Input</i> | 40 |
| Tabel 3.9 Hasil Perhitungan DCA..... | 41 |
| Tabel 3.10 Hasil Akhir Level DCA..... | 42 |
| Tabel 3.11 Hasil Level menggunakan DCA dan Tanpa DCA | 43 |
| Tabel 3.12 Hasil Level pada Tabel 3.11 | 45 |
| Tabel 3.13 <i>Confusion Matrix</i> Hasil Level (<i>FP</i>)..... | 46 |
| Tabel 3.14 <i>Confusion Matrix</i> Hasil Level (<i>FN</i>) | 47 |
| Tabel 4.1 Percobaan pengujian 1..... | 58 |
| Tabel 4.2 Percobaan pengujian 2..... | 61 |
| Tabel 4.3 Percobaan pengujian 3..... | 64 |
| Tabel 4.4 Percobaan pengujian 4..... | 66 |
| Tabel 4.5 Percobaan pengujian 5..... | 69 |
| Tabel 4.6 Percobaan pengujian 6..... | 72 |
| Tabel 4.7 Percobaan pengujian 7..... | 74 |
| Tabel 4.8 Percobaan pengujian 8..... | 77 |
| Tabel 4.9 Percobaan pengujian 9..... | 80 |
| Tabel 4.10 Percobaan pengujian 10..... | 82 |
| Tabel 4.11 Percobaan pengujian 11..... | 85 |
| Tabel 4.12 Percobaan pengujian 12..... | 88 |
| Tabel 4.13 Percobaan pengujian 13..... | 90 |
| Tabel 4.14 Percobaan pengujian 14..... | 93 |
| Tabel 4.15 Percobaan pengujian 15..... | 96 |

| | |
|---|-----|
| Tabel 4.16 Percobaan pengujian 16..... | 98 |
| Tabel 4.17 Percobaan pengujian 17..... | 101 |
| Tabel 4.18 Percobaan pengujian 18..... | 103 |
| Tabel 4.19 Percobaan pengujian 19..... | 106 |
| Tabel 4.20 Percobaan pengujian 20..... | 108 |
| Tabel 4.21 Rekapitulasi Hasil Percobaan..... | 111 |



ABSTRAK

Umami, Lisa Aulia. 2020. *Implementasi Dynamic Challenging Level Adapter (DCA) Menggunakan Fuzzy Sugeno Pada Level Game Adventure Gunung Kelud untuk NPC*. Skripsi. Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Fresy Nugroho, M. T. (II) M. Imamuddin, Lc., MA.

Kata Kunci : *Game*, Tingkat kesulitan, *Dynamic Challenging Level Adapter (DCA)*, *Fuzzy Sugeno*

Pengembangan sistem level dalam pembuatan *game* membutuhkan desain dan alur yang menarik untuk mempertahankan minat serta keterlibatan pemain sebagai hiburan hingga mendapatkan kepuasan batin tersendiri. *Difficulty Adjustment* atau tingkat kesulitan level sebuah *game* dibuat semenarik mungkin agar pemain tidak merasa bosan apabila level terlalu mudan ataupun merasa frustrasi apabila level terlalu sulit. Dalam hal ini peneliti membuat sebuah sistem level *game* dengan kabut sebagai objek dengan menerapkan metode DCA sebagai pengelola variabel input diantaranya, variabel *score*, *time remain* dan *player life* yang bertujuan untuk melihat kesesuaian tingkatan level kabut dengan bantuan *Fuzzy Sugeno* sebagai pengambil keputusan level kabut yang akan ditetapkan. Kemudian dilakukan pengujian dan perhitungan menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi, presisi, *recall*, dan *f-measure* dengan menggunakan data hasil pengujian antara sistem yang menggunakan metode DCA dan ataupun sistem yang tanpa metode. Hasil dari 20 percobaan pengujian yang telah dilakukan diperoleh nilai akurasi dan *precision* terbesar berada pada percobaan 4 dengan nilai akurasi sebesar 0,6 dan *precision* sebesar 0,5 serta untuk nilai terkecil berada pada percobaan 20 dengan nilai akurasi sebesar 0,2 dan *precision* sebesar 0,066666667. Sedangkan untuk nilai *recall* dan *f-measure* terbesar berada pada percobaan 6 dengan nilai *recall* sebesar 0,3166667 dan *f-measure* sebesar 0,3506711, serta untuk nilai terkecil berada pada percobaan 16 dengan nilai *recall* sebesar 0,04 dan *f-measure* sebesar 0,0571420039.

ABSTRACT

Umami, Lisa Aulia. 2020. *Implementation of Dynamic Challenging Level Adapter (DCA) Using Fuzzy Sugeno on Mount Kelud Adventure Game Level for NPCs*. Essay. Department of Informatics, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Advisors: (I) Fresy Nugroho, M. T. (II) M. Imamuddin, Lc., MA.

Keyword : Game, Difficulty Adjustment, Dynamic Challenging Level Adapter (DCA), Fuzzy Sugeno

The development of the system level in making games requires a design and flow that is attractive to maintain the interest and involvement of the players as entertainment to get the inner satisfaction of its own. Difficulty Adjustment or the level of difficulty level of a game made as attractive as possible so that players do not feel bored if the level is too mudan or feel frustrated if the level is too difficult. In this case the researchers make a system level the game with the fog as objects with apply methods of DCA as the manager of the input variables, such variables score, time remain and the player's life that aims to see the suitability of the levels of fog with the help of Fuzzy Sugeno as decision-makers the level of fog that will be set. Then do the testing and calculations using a Confusion Matrix to determine the value of accuracy, precision, recall, and f-measure by using the testing result data between the systems using the method of DCA and or system without the method. The results of 20 experiments testing has been done to obtain the value of accuracy and precision is greatest on trial 4 with an accuracy value of 0.6 and a precision of 0.5.as well as to the smallest value is on the trial 20 with the accuracy values of 0.2 and precision of 0,066666667. As for the value of recall and f-measure is the largest are on the experiment 6 with the recall value of 0,3166667 and f-measure of 0,3506711, as well as to the smallest value was at experiment 16 with the recall value of 0.04 and f-measure of 0,0571420039.

المخلص

أومامي, ليسا اوليا. ٢٠٢٠. تنفيذ محول مستوى التحدي الديناميكي باستخدام غامض سوجينو على جبل كيلود مستوى لعبة المغامرة لشخصية غير لاعب. أطروحة. قسم علوم و الطقنية, كلية العلوم و الهندسة، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرف: (١) فراشي نغراها الماجستير (٢) محمد امام الدين ليسانس, الماجستير.

الكلمات الرئيسية: لعبة ، تعديل صعوبة ، DCA, غامض سوجينو

يتطلب تطوير مستوى النظام في صنع الألعاب تصميمًا وتدقيقًا جذابًا للحفاظ على اهتمام وإشراك اللاعبين كترفيه للحصول على الرضا الداخلي الخاص به. تعديل صعوبة أو مستوى مستوى صعوبة لعبة جعلت جذابة قدر الإمكان بحيث لا يشعر اللاعبون بالملل إذا كان المستوى مودان جدا أو يشعرون بالإحباط إذا كان المستوى صعبا للغاية. في هذه الحالة الباحثون جعل مستوى النظام اللعبة مع الضباب ككائنات مع تطبيق أساليب ديناميكية محول مستوى التحدي كمدير للمتغيرات المدخلات ، مثل هذه المتغيرات يسجل ، والوقت لا يزال وحياة اللاعب الذي يهدف إلى رؤية مدى ملاءمة مستويات الضباب مع مساعدة من غامض سوجينو وصناع القرار مستوى الضباب الذي سيتم تعيينه. ثم قم بإجراء الاختبار والحسابات باستخدام مصفوفة الارتباك لتحديد قيمة الدقة والدقة والاستدعاء وقياس الدقة باستخدام بيانات نتيجة الاختبار بين الأنظمة باستخدام طريقة محول مستوى التحدي الديناميكي أو بدون الطريقة. تم إجراء نتائج اختبار 20 تجربة للحصول على قيمة الدقة والدقة هي الأكبر في التجربة 4 بقيمة دقة 0,6 ودقة 0,5. فضلا عن أصغر قيمة على محاكمة 20 مع قيم دقة 0,2 ودقة 0,066666667. أما بالنسبة لقيمة الاستدعاء وقياس الدقة هو الأكبر على التجربة 6 مع قيمة الاستدعاء من 0,3166667 وقياس دقة 0,3506711, وكذلك لأصغر قيمة كانت في التجربة 16 مع قيمة الاستدعاء من 0,04 وقياس دقة 0,0571420039.

BAB I PENDAHULUAN

1.1 Latar Belakang

Game merupakan permainan berbentuk multimedia yang dibuat menarik dan mempunyai sifat sebagai hiburan, karena bagi pengguna yang memainkan *game* akan merasa senang serta mendapatkan kepuasan batin tersendiri. *Game* merupakan salah satu wujud dari pemanfaatan teknologi. Beberapa tahun terakhir industri *game* telah mengalami serangkaian perubahan besar. Meningkatnya keunggulan *game online* persisten, *platform* dan portal distribusi digital, *game mobile* dan sosial, dan kemunculan model bisnis baru semuanya telah mengubah aspek fundamental pembuatan dan bermain *game* (Riedl & Zook, 2013). Di era saat ini, *game* disajikan dengan kualitas visualisasi yang cukup canggih, sehingga pemain lebih interkatif sesuai kemaunnya sendiri. Karena didukung oleh teknologi pemain terasa hidup dalam *game* tersebut. Sehingga dapat dikatakan bahwa *game* dapat berkembang beriringan dengan teknologi (Pratama, 2014).

Game umumnya juga banyak diketahui mempunyai unsur kecerdasan. Kecerdasan dalam *game* saat ini juga semakin berkembang pesat. Termasuk kecerdasan buatan atau *Artificial Intellingence* (AI) yang diterapkan pada karakter bukan pemain atau sering disebut *Non-Player Character* (NPC) yang lebih dikenal dengan sebutan AI NPC. Menurut (Owen, 2009), istilah "*game* AI" kadang-kadang digunakan untuk membedakan AI yang digunakan dalam *game* komputer dari AI akademik.

Menurut Shin. H.C dan Nai. Y.Y (2012), alasan *game-game* menarik banyak pemain *game* tidak hanya presentasi *game* yang mewah tetapi juga *game* AI yang menantang lawan komputer. Untuk mencocokkan tingkat tantangan *game* dengan pemain *game* yang berbeda, *game* ini selalu menyediakan beberapa tingkat kesulitan *default* untuk pemilihan pemain *game*. Namun, pengaturan tingkat kesulitan ini tidak selalu dapat mengakomodasi persyaratan tingkat yang menantang dari pemain permainan tingkat yang berbeda

Tingkat kesulitan *default* atau statis dalam *game* ini mungkin tidak cocok dengan kebutuhan setiap pemain *game*. Dengan kata lain, sulit bagi pemain untuk memilih *level* tantangan yang tepat cocok untuknya. Saat *player* memainkan sebuah permainan berharap dapat menyelesaikan misi dan mengikuti alur permainan sesuai dengan *genre* yang dimainkan untuk mencapai tujuan kemenangan. Dalam penerapan *level* atau tingkat kesulitan di dalam *game*, penggunaan mekanisme *Dynamic Challenging Level Adapter* (DCA) digunakan untuk mengatur tingkat kesulitan tantangan *game*. Kelebihan dari penggunaan mekanisme DCA ini adalah mengatur tantangan di dalam *game* secara dinamis menyesuaikan perilaku lawan komputer. Objek yang digunakan pada penelitian ini yaitu AI NPC dan rintangan di dalam *game* petualangan Gunung Kelud.

Rintangan, tantangan atau hambatan yang dihadapi pemain merupakan bagian dalam perancangan skenario dalam *game*. Rintangan dalam permainan adalah bagian dimana pemain atau *player* akan melewati atau menghadapi untuk dapat meneruskan atau menyelesaikan alur sebuah *game*. Sebagai contoh *game adventure* atau *game* petualangan: Super Mario Bros (*Game Nintendo*, 1985).

Rintangan dalam *game* tersebut seperti melompati jurang, menghindari bola api, dan mendapatkan senjata untuk menyerang musuh yang ada di depannya, serta terdapat beberapa objek yang dapat dihancurkan untuk dapat dilewati oleh *player*. Tujuan utama dari permainan Mario Bros di setiap level adalah untuk sampai ke tingkat akhir (Saurik et al., 2015).

Setiap permainan atau *game* yang telah dibuat memiliki tujuan atau target tertentu. Seperti halnya sebagai hiburan serta untuk melatih atau mengasah kemampuan kecerdasan pemain. Dalam Al-Qur'an surah Al-Baqarah ayat 269 dijelaskan mengenai anugerah yang diberikan oleh Allah sebuah kecerdasan melalui kemampuan akal pikiran, Allah berfirman:

يُؤْتِي الْحِكْمَةَ مَنْ يَشَاءُ وَمَنْ يُؤْتَ الْحِكْمَةَ فَقَدْ أُوتِيَ خَيْرًا كَثِيرًا وَمَا يَذَّكَّرُ إِلَّا أُولُو الْأَلْبَابِ

Artinya: “Allah menganugerahkan *al hikmah* (kefahaman yang dalam tentang *Al Quran dan As Sunnah*) kepada siapa yang dikehendaki-Nya. Dan barangsiapa yang dianugerahi *hikmah*, ia benar-benar telah dianugerahi karunia yang banyak. Dan hanya orang-orang yang berakallah yang dapat mengambil pelajaran.” (QS. Al-Baqarah:269)

Tafsir Al-Qur'an menurut pengawasan Syaikh Prof. Dr. Imad Zuhair Hafidz. Dari kata *يُؤْتِي الْحِكْمَةَ* (Allah menganugerahkan *al hikmah* kepada siapa yang dikehendaki-Nya). Yakni berupa ilmu, dan pendapat lain mengatakan berupa pemahaman berbagai hal, terlebih lagi pemahaman terhadap Al-Qur'an dan as-Sunnah, dan pendapat lainnya mengatakan yakni berupa ketepatan dalam perkataan.

Selain mengasah kecerdasan, *game* atau permainan juga pada awalnya terdapat banyak rintangan yang harus dilalui pemain untuk menyelesaikannya. Meskipun terdapat rintangan tujuan utamanya adalah agar *game* lebih menantang

dan tidak membosankan sehingga pemain dapat merasa terhibur selama bermain. Disebutkan juga dalam hadis *Shahih Al Jami*. Rasulullah *shallallahu 'alaihi wa sallam* bersabda:

لِيَغْزِرَ الْمُسْلِمِينَ فِي مَصَائِبِهِمْ الْمَصِيبَةُ بِي

Artinya: “Musibah yang menimpaku sungguh akan menghibur kaum muslimin.” (HR. *Shahih Al Jami*, 5459, dari *Al Qosim bin Muhammad*. Syaikh *Al Albani* mengatakan bahwa hadis ini shahih)

Dalam lafadz yang lain disebutkan:

مَنْ عَظَمَتْ مُصِيبَتَهُ فَلْيَذْكُرْ مُصِيبَتِي، فَإِنَّهَا سَتَهَوِّنُ عَلَيْهِ مُصِيبَتَهُ

Artinya: “Siapa saja yang terasa berat ketika menghadapi musibah, maka ingatlah musibah yang menimpaku. Ia akan merasa ringan menghadapi musibah tersebut.” (Disebutkan dalam *Bahjatul Majalis wa Ansul Majalis*, Ibnu ‘Abdil Barr, hal. 249, *Mawqi’ Al Waroq*)

Dari hadis tersebut dapat diketahui ternyata, musibah orang yang lebih sholih dari kita memang lebih berat dari yang kita alami. Ingatlah bahwa Nabi kita *shallallahu 'alaihi wa sallam* sering mendapatkan cobaan sampai dicaci, dicemooh dan disiksa oleh orang-orang musyrik dengan berbagai cara. Kalau kita mengingat musibah yang menimpa beliau, maka tentu kita akan merasa ringan menghadapi musibah kita sendiri karena musibah kita dibanding beliau tidaklah seberapa.

Berbagai model metode tingkat kesulitan pada *game* telah dikembangkan. Salah satunya adalah *Dynamic Challenging Level Adapter* (DCA) yang diimplementasikan pada permainan *game adventure*. DCA merupakan mekanisme komputer untuk menentukan tindakan secara otomatis terhadap tingkah laku pemain yang beragam. Karena DCA adalah hasil perkembangan dari model DDA, ada perbedaan diantara keduanya. Pada dasarnya penggunaan DCA akan menyesuaikan tingkah laku komputer dengan memperhatikan tingkah laku pemain

di dalam permainan yang menggunakan tingkat kesulitan standar AI NPC sebagai *player*.

Kelebihan *Fuzzy Sugeno* adalah dapat memberikan keputusan yang adil. Keputusan yang adil disini dikatakan karena fuzzy dapat diterapkan dalam permasalahan yang sering ditemui di dunia nyata yang sulit di ekspesikan. *Fuzzy Sugeno* lebih cocok diterapkan pada pemilihan keputusan untuk permasalahan yang menangani control seperti, memutuskan untuk memilih ukuran besar atau kecil, banyak atau sedikit, rendah atau tinggi, dan lain-lain (Suyanto, 2014).

Penelitian oleh (Rahman, 2019) yang membuat sebuah *game* berkategori simulasi. Simulasi pada *game* yang dibuat digunakan untuk mempersiapkan barang yang akan di bawa saat terjadi bencana. Pada penelitian yang telah dilakukan, peneliti menggunakan metode *Simple Additive Weighting* sebagai penentu rekomendasi barang yang akan diambil. *Game* yang dibuat oleh Rahman. A (2019), terkesan lebih nyata karena menggunakan media yang berbasis *virtual reality* dan *leap motion*.

Penelitian sebelumnya oleh (Cahyani, 2020) yang menambahkan fitur kabut realistis pada *game*, dengan acuan kabut akan bertambah atau berkurang sesuai dengan waktu setempat di Gunung Kelud dengan pengambilan data melalui CCTV. Objek kabut tersebut dalam penelitiannya dibuat dalam bentuk simulasi adegan kabut ralistic pada *unity 3D* dengan perapan metode *Perlin Noise*. Sehingga kabut realistis yang dibuat pada *game 3D* menjadi daya tarik dalam adegan *game* simulasi yang dibuat serta dapat meningkatkan realitas virtual pada permainan.

Berdasarkan uraian dua paragraf diatas, maka dalam penelitian ini penulis ingin mengajukan mekanisme kontrol permainan AI, yang disebut *Dynamic Challenging Level Adapter* (DCA), untuk secara otomatis menghasilkan AI permainan yang baik dengan menggunakan *Fuzzy Sugeno* sebagai pengatur keputusan tingkat kabut yang dihasilkan dalam *game*. Kabut berperan sebagai rintangan atau halangan untuk *player* selama bermain, semakin tebal kabut yang dihasilkan semakin tinggi *level* yang dicapai *player* dan begitu sebaliknya, proses perubahan kabut terus berganti sesuai kondisi sampai *player* menyelesaikan permainan. Serta dalam penelitian ini NPC digunakan sebagai *player* dalam *game*.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, permasalahan yang dapat dirumuskan sebagai berikut:

1. Bagaimana membuat sistem level pada *game adventure* Gunung Kelud yang mengatur tingkat kesulitan *game* dengan menerapkan metode *Dynamic Challenging Level Adapter* berbasis *Fuzzy Sugeno* dengan tanpa metode di dalam *game* yang mempengaruhi perubahan level *game* secara dinamis sesuai perilaku *player*?
2. Berapa akurasi, presisi, *recall* dan *f-measure* dari sistem level dengan menggunakan data hasil level pada *game* yang menerapkan metode *Dynamic Challenging Level Adapter* berbasis *Fuzzy Sugeno* dengan hasil level pada *game* tanpa metode?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut:

1. *Game* dibuat menggunakan *Unity 3D* sebagai *Game Engine*
2. *Game* dengan kategori *adventure* atau petualangan mengambil barang dan melewati rintangan untuk menyelesaikan misi dengan *gameplay* yang mengacu pada penelitian sebelumnya oleh Abdul Rahman di tahun 2019
3. *Dynamic Challenging Level Adapter* digunakan untuk mengatur tingkatan kabut di dalam *game* dengan bantuan *Fuzzy Sugeno* sebagai AI sehingga level kabut dapat berubah secara otomatis, dimana data objek kabut diperoleh dari penelitian sebelumnya oleh Berlian Gita Cahyani di tahun 2020

1.4 Tujuan Penelitian

Berdasarkan latar belakang yang sudah dijelaskan maka penulis mempunyai tujuan seperti berikut:

1. Mengetahui implementasi *Dynamic Challenging Level Adapter* menggunakan *Fuzzy Sugeno* dengan membangun sistem level pada *game adventure Gunung Kelud*
2. Mengetahui tingkat akurasi, presisi, *recall* dan *f-measure* pada sistem level dengan menggunakan hasil level penerapan tanpa metode dan hasil level menggunakan metode

1.5 Manfaat Penelitian

1. Memperoleh sebuah sistem permainan serta tingkat kesulitan sebuah permainan yang diberi kecerdasan buatan agar dinamis
2. Memberikan pengalaman membuat sebuah *game* dengan sistem yang dinamis menyesuaikan tingkat kesulitan AI NPC sebagai *player*
3. Meningkatkan daya tarik di dalam *game* dengan penyesuaian sistem level sehingga dapat mempertahankan minat serta keterlibatan pemain bermain untuk waktu yang lama



BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan beberapa studi pustaka yang digunakan sebagai dasar teori dalam penelitian. Selain itu, bab ini juga membahas tentang penelitian terkait dengan penelitian yang akan dilakukan.

2.1 Penelitian Terkait

Terdapat beberapa penelitian terdahulu di bawah ini yang mendukung penelitian tentang menggunakan metode *Dynamic Challenging Level Adapter* (DCA) dengan menggunakan *Fuzzy Sugeno*, antara lain sebagai berikut:

Penelitian oleh (Chang & Yang, 2012) menggunakan mekanisme *Dynamic Challenging Level Adapter* (DCA) untuk secara otomatis menyesuaikan perilaku lawan komputer untuk pemain *game* yang berbeda. Setiap pemain *game* tidak perlu memilih tingkat kesulitan di muka dan AI permainan lawan komputer yang dikendalikan oleh mekanisme DCA dapat secara dinamis beradaptasi untuk memenuhi tingkat tantangan dari setiap pemain *game*. Dari hasil percobaan, perbedaan nilai tingkat tantangan (CR) lawan komputer ditingkatkan oleh mekanisme DCA dapat dikurangi lebih dari 80%. Selain itu, durasi pertandingan biasanya dua kali lipat serta pemenang selalu ditentukan di belakang setiap pertandingan.

Menurut penelitian oleh (Attoriq et al., 2016) yang menerapkan algoritma *fuzzy logic sugeno* pada permainan *battle city* untuk membuat perilaku komputer menjadi susah ditebak pada saat bermain, dan menerapkan

algoritma A* untuk mencari jalur terpendek menuju base. *Game* ini dibangun dan dirancang menggunakan unity 3D dan menggunakan metodologi prototype. Hasil uji dari 30 data sampel uji algoritma *fuzzy logic sugeno* menunjukkan bahwa tingkat keberhasilan menentukan perilaku NPC sebesar 100%. Hasil uji dari 30 data sampel uji algoritma A* menunjukkan bahwa tingkat keberhasilan mencari jalur terpendek sebesar 100% pada permainan *Battle City*.

Penelitian selanjutnya oleh (Oktavia & Maulidi, 2019) mengenai logika *Fuzzy Sugeno* untuk penentuan *reward*. Pengembangan permainan edukasi dengan menerapkan logika *fuzzy*. Permainan ini membutuhkan ketangkasan dan kecepatan respon dari pemain. Permainan edukasi ini juga sebagai sarana penyampaian untuk memberikan edukasi kepada anak usia dini untuk waspada terhadap orang yang tidak dikenal. Penentuan *reward* berdasarkan kondisi nyawa, waktu dan skor yang didapatkan pemain. Logika *fuzzy* yang digunakan pada penelitian ini adalah *Fuzzy Sugeno*. Hasil yang didapatkan pada penelitian ini adalah permainan edukasi dapat diterapkan menggunakan logika *Fuzzy Sugeno* pada perangkat mobile dan pengujian *rule fuzzy* dengan *reward* pada *game* menunjukkan hasil yang sesuai.

2.2 Landasan Teori

2.2.1 *Game adventure*

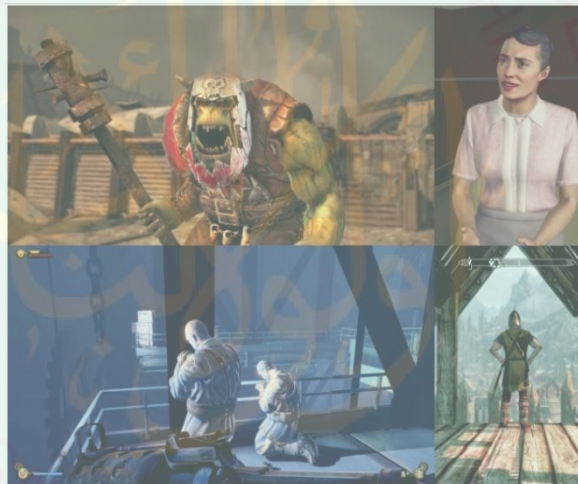
Kemunculan *game* pertama kalinya, *game* masih disajikan secara sederhana dan di prakarsai oleh Steven Russel dan proyek yang bernama *Computer Games* pada tahun 1962 dengan produk andalannya bernama *Star*

Wars. Beberapa puluh tahun kemudian, banyak *game* bermunculan dari *game* 2 dimensi dan *game* 3 dimensi. Serta yang bersifat hiburan saja ataupun sebagai media pembelajaran atau edukatif. *Game* pada dasarnya bersifat hiburan karena jika pengguna memainkan *game* maka akan terasa senang. Dalam era saat ini, *game* disajikan dengan kualitas visualisasi yang cukup canggih karena didukung oleh teknologi sehingga pemain lebih interkatif sesuai kemaunnya sendiri dan pemain terasa hidup dalam *game* tersebut. Maka bisa disebutkan bahwa *game* berkembang beriringan dengan teknologi (Pratama, 2014).

Salah satu kategori *game* disini adalah *game adventure*. *Game* ini merupakan *game* petualangan dimana dalam perjalanannya, pemain akan menemukan banyak hal dan peralatan yang akan disimpan. Peralatan itu akan digunakan selama dalam perjalanan, baik untuk membantu dan menjadi petunjuk. Permainan *genre* ini merupakan permainan yang melakukan penjelajahan seperti memanjat, menelusuri hutan, meloncati tebing yang terpisah jurang, berayun dari pohon ke pohon lainnya, bergulat melawan tanaman atau pun hewan liar demi mencari *clue* atau petunjuk menuju rintangan berikutnya. Adapun yang bertualang diantara jalan jalan perkotaan sekedar mencari tongkat kayu ataupun sabuk untuk membuat alat untuk misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam permainan jenis ini. Contoh: *Kings Quest*, dan *Space Quest*.

2.2.2 *Non-Player Character*

Karakter bukan pemain atau *Non-Player Character* (NPC), adalah karakter di dalam permainan komputer yang dikendalikan oleh komputer, bukan pemain. Istilah ini ada sebelum *game* digital. Dalam *game* digital, NPC datang dalam berbagai bentuk dan ukuran yang berbeda, dan seperti yang diilustrasikan dalam gambar 2.1, mereka berkisar dari yang biasa, yang fantastis, hingga yang tidak manusiawi. NPC adalah entitas yang terlihat seperti pemain dan yang akan berpikir mereka sama jika diberi AI (Warpefelt, 2016).



Gambar 2.1 Contoh penggambaran NPC

Sumber : (Warpefelt, 2016)

Menurut Warpefelt (2016), pada dasarnya NPC secara aktif terlibat dalam penggambaran perannya, dan akan bertindak dengan cara yang kondusif untuk meyakinkan pemain bahwa ia memang ada dalam peran itu. Ini pada dasarnya berarti bahwa NPC harus dapat memberlakukan bagian dari apa yang disebut kepribadian.

2.2.2.1 Navigation Meshes (Navmesh)

Navigation meshes adalah kumpulan poligon cembung, yang disebut *node* yang mewakili rute atau jalur yang dapat dilalui dalam permainan, seperti yang diilustrasikan dalam gambar 2.2. Setiap *node* mewakili area lingkungan permainan. Seorang NPC dapat berjalan dalam garis lurus antara dua titik. *Navmesh* dapat berbentuk segitiga atau bentuk tidak beraturan. Hubungan antar *node* dapat ditentukan dan grafik yang cocok untuk algoritma pencarian *path* dapat dihasilkan. Setelah grafik dihasilkan algoritma pencarian, seperti A* dapat digunakan untuk menemukan jalur dari satu node ke yang lain (Hartley & Mehdi, 2012).



Gambar 2.2 Ilustrasi Navmesh

Sumber: (Hartley & Mehdi, 2012)

Menurut (Heckel et al., 2009) yang menggunakan *navmesh* untuk adaptasi NPC dalam *game* biasanya berfokus pada mengadaptasi rute *pathfinding* A* melalui teknik seperti analisis lokasi taktis dan evaluasi posisi. NPC dapat mencari rute untuk bergerak dari posisi sekarang menuju

posisi tujuannya dengan *pathfinding*. Perilaku *pathfinding* untuk NPC menerapkan metode *navmesh* yang menggunakan algoritma A*. NPC dalam *game* yang dibuat berperan sebagai *player*, dimana NPC harus bergerak mencari barang yang harus diambil dengan cara mencari titik-titik target terdekat. NPC menelusuri setiap node ke node yang lain, kemudian menentukan rute atau jalur terdekat yang harus dilalui untuk melakukan pencarian menuju target. NPC bekerja melakukan pencarian target begitu seterusnya sampai titik target telah habis.

2.2.3 Fitur Kabut

Sistem yang digunakan dalam media penelitian yaitu kabut (*fog*). Kabut adalah awan lembab yang melayang di dekat tanah (KBBI) yang mana disebabkan adanya proses kondensasi yang terjadi pada tanah akibat peristiwa pemanasan atau pendinginan udara. Kabut mirip dengan awan, perbedaan diantara keduanya adalah awan tidak menyentuh permukaan bumi, sedangkan kabut menyentuh permukaan bumi. Penerapan DCA dengan *Fuzzy Sugeno*, fitur kabut atau objek kabut diambil dari penelitian Berlian Gita Cahyani (2019) yang digunakan dalam penelitiannya kabut telah dibagi dalam beberapa tingkat visibilitas pada tabel 2.1.

Tabel 2.1 Tingkat visibilitas internasional dan koefisien hamburan

Sumber: (Guo et al., 2014)

| Tingkat | Kondisi Cuaca | Jarak Penglihatan | Koefisien Hamburan |
|---------|---------------|-------------------|--------------------|
| 0 | Dense fog | <50 m | >78.2 |
| 1 | Thick fog | 50 - 200 m | 78.2 – 19.6 |
| 2 | Moderate fog | 200 – 500 m | 19.6 – 7.82 |

| Tingkat | Kondisi Cuaca | Jarak Penglihatan | Koefisien Hamburan |
|---------|-----------------|-------------------|--------------------|
| 3 | Light fog | 500 m – 1 km | 7.82 – 3.91 |
| 4 | Thin fog | 1 km – 2 km | 3.91 – 1.96 |
| 5 | Haze | 2 km – 4 km | 1.960 – 0.954 |
| 6 | Light haze | 4 km – 10 km | 0.954 – 0.391 |
| 7 | Jernih | 10 km – 20 km | 0.391 – 0.196 |
| 8 | Sangat jernih | 20 km – 20 km | 0.196 – 0.078 |
| 9 | Extremely clear | >50 km | 0.0141 |

Pembagian tingkatan kabut pada tabel 2.1 dibagi menjadi 10 tingkatan, mulai dari tingkat 0 sampai tingkat 9 yang menghasilkan kabut berupa *dense fog* sampai *extremely clear*. Pembagian perkiraan tingkatan kabut dari penelitian sebelumnya akan diterapkan di dalam sistem *game* sebagai penentu *level* di dalam *game*. Berikut pembagiannya terlihat pada tabel 2.2:

Tabel 2.2 Pembagian level kabut di dalam *game*

| No | Level Game | Jenis Kabut |
|----|------------|-----------------|
| 1 | Level 1 | Extremely clear |
| 2 | Level 2 | Sangat jernih |
| 3 | Level 3 | Jernih |
| 4 | Level 4 | Light haze |
| 5 | Level 5 | Haze |
| 6 | Level 6 | Thin fog |
| 7 | Level 7 | Light fog |
| 8 | Level 8 | Moderate fog |
| 9 | Level 9 | Thick fog |
| 10 | Level 10 | Dense fog |

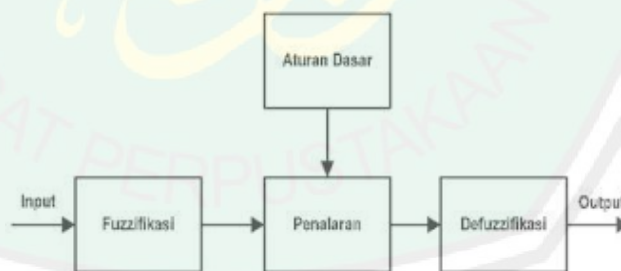
Pembagian tingkatan kabut pada tabel 2.2, digunakan untuk menentukan batasan *output* ketika *level* berubah. Setiap jenis kabut mempunyai ketebalan

berbeda gunanya untuk membedakan setiap perubahan *level game* yang bisa naik turun antara *level 1* sampai *level 10*.

2.2.4 Fuzzy Sugeno

Model *Fuzzy Sugeno* memiliki pengertian yang hampir sama dengan model *Fuzzy Mamdani*. Perbedaannya terletak pada *output* yang dihasilkan *Fuzzy Mamdani* berupa himpunan *Fuzzy*, sedangkan dalam model *Fuzzy Sugeno output* berupa konstanta atau persamaan linier (Mahmud et al., 2016).

Logika *fuzzy* merupakan suatu proses dalam pengambilan keputusan yang berdasarkan pada aturan. Tujuan dari logika *fuzzy* adalah memecahkan masalah suatu pernyataan yang kurang jelas menjadi sebuah pengertian yang bersifat logis. Proses pengambilan keputusan yang merupakan bagian dari sistem kecerdasan buatan yang dapat meniru cara berfikir manusia yang selanjutnya dijalankan oleh mesin (Kusumadewi & Purnomo, 2010). Berikut adalah blok diagram logika *fuzzy* seperti pada gambar 2.3.



Gambar 2.3 Blok diagram logika *fuzzy*

Sumber : (Nugroho, 2010)

Tahapan pertama logika *fuzzy* mulai dari *input* variabel kemudian di proses *fuzzyfikasi* untuk memperoleh nilai derajat keanggotaan, kemudian proses implikasi akan dibandingkan tiap variabel sesuai dengan *rule fuzzy*.

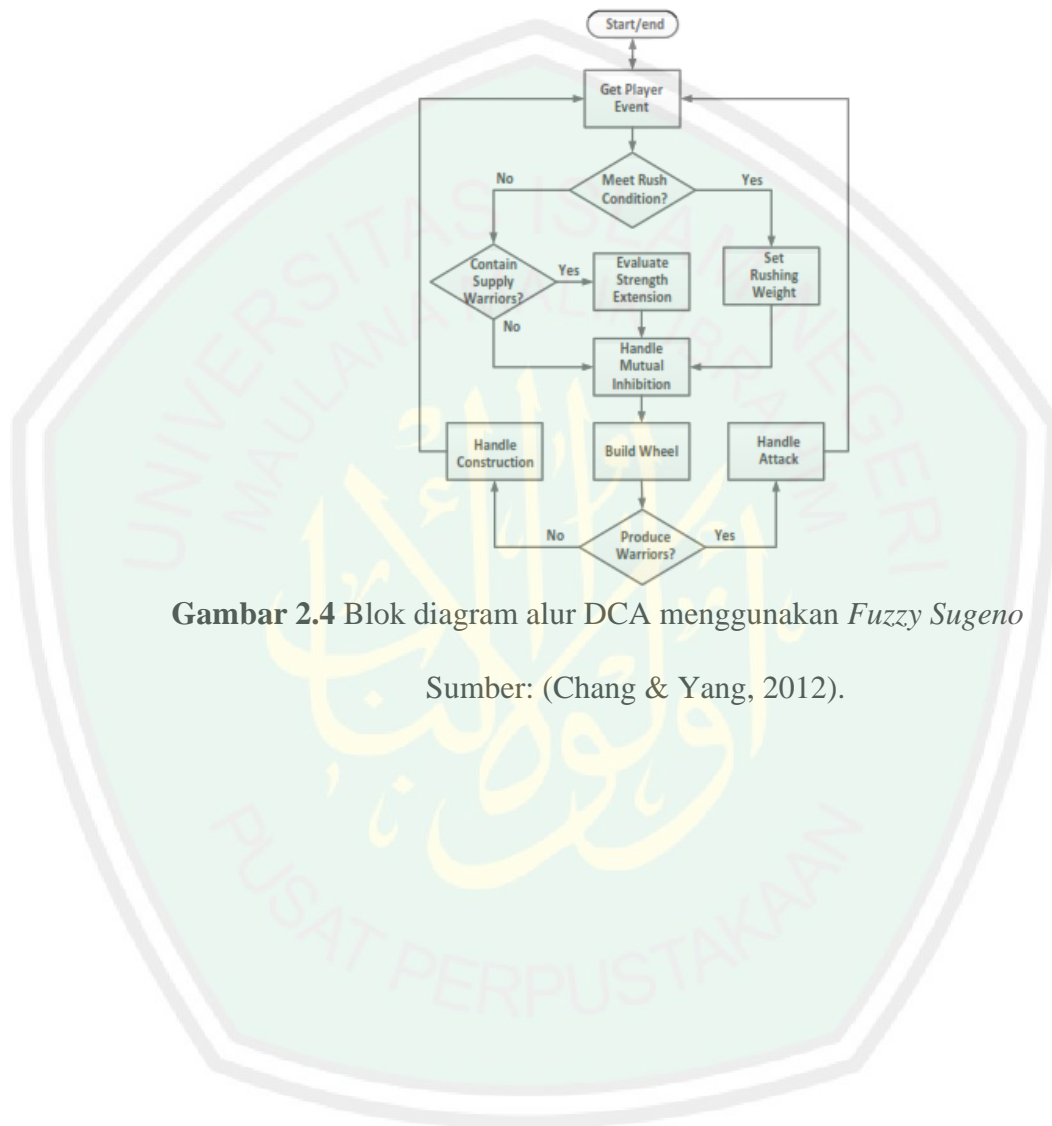
Selanjutnya tahap defuzzyfikasi yang merupakan tahap akhir dengan menghitung nilai minimum sehingga akan menghasilkan nilai *output* yang dikeluarkan.

2.2.5 Dynamic Challenging Level Adapter (DCA)

Hal utama yang mendasari DCA yang diusulkan menurut (Chang & Yang, 2012) adalah untuk menganalisis elemen-elemen dasar, termasuk barang, sumber daya, dan hal umum lainnya dalam *game*. DCA dapat menjaga kekuatan kompetitif dari setiap pertandingan hingga akhir pertandingan dan mempersempit perbedaan *level* antara lawan komputer dan pemain *game*. Selain itu, untuk mengatasi masalah lawan komputer yang kaku ini, DCA meningkatkan AI *game* dengan strategi putaran roda yang tidak ditentukan. Kecuali secara dinamis menyesuaikan tingkat tantangan untuk pemain yang berbeda, mekanisme DCA dapat menjalankan strategi yang berbeda untuk meningkatkan kesenangan *game*, dan hasil pertandingan bisa berbeda bahkan jika menghadapi pemain *game* yang sama. DCA yang diusulkan oleh penulis secara otomatis menghasilkan strategi permainan lawan komputer.

Proses DCA diawali dengan mengekstraksi *event* yang pemain lakukan, lalu dilakukan pengolahan dan mengevaluasi apa tindakan yang seharusnya dilakukan. Dengan proses pertimbangan maka mekanisme DCA akan memilih keputusan terbaik yang akan dilakukan dalam kurun waktu tersebut. Pada umumnya dalam permainan terdapat dua unsur umum yang dapat dilakukan pertama adalah mengambil strategi dan kedua adalah menjalankan

mi. Kedua unsur ini yang akan menjadi roda putar dinamis dalam pengaplikasian DCA. Berikut blok diagram alur DCA yang menggunakan *Fuzzy Sugeno* pada gambar 2.4.



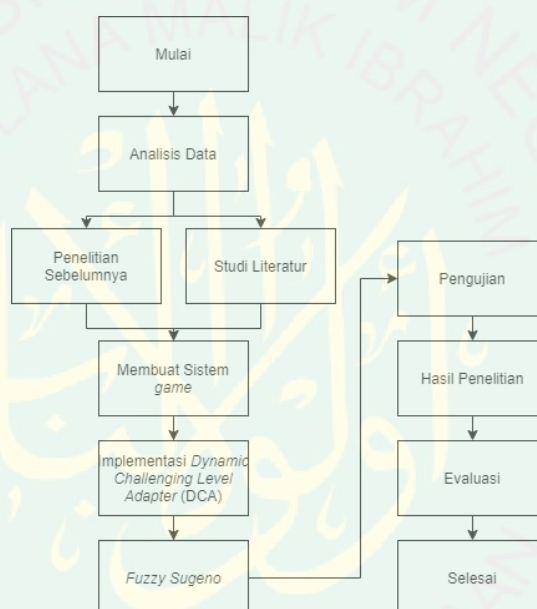
Gambar 2.4 Blok diagram alur DCA menggunakan *Fuzzy Sugeno*

Sumber: (Chang & Yang, 2012).

BAB III

METODOLOGI PENELITIAN

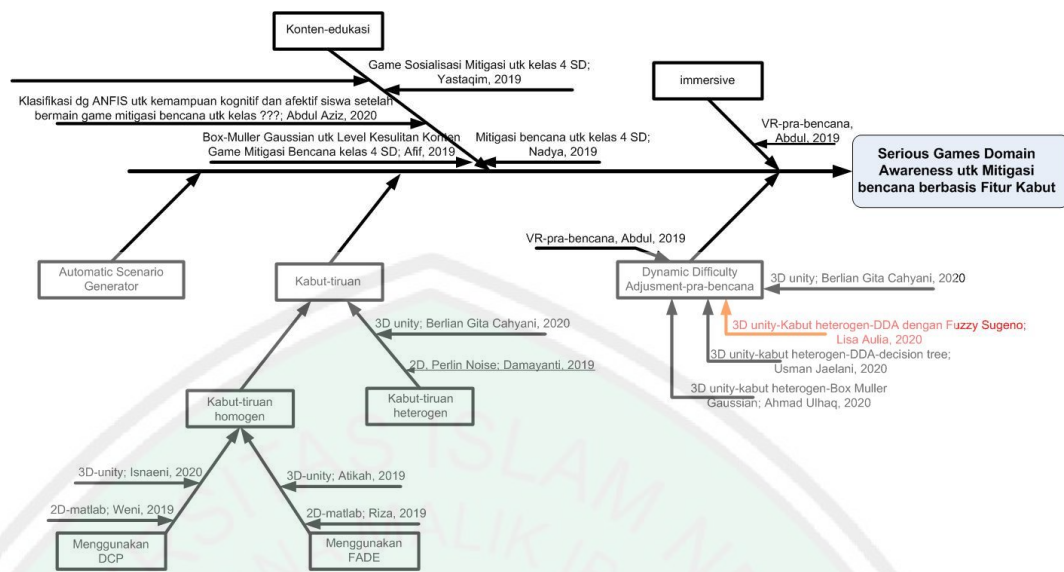
Pada bab ini menjelaskan tentang metodologi penelitian dan tahapan bagaimana penelitian ini akan dilakukan. Penelitian ini akan membuat sebuah *game* yang berkategori *adventure* dengan implementasi *Dynamic Challenging Level Adapter* (DCA) untuk penentu tingkat kesulitan yang diberi kecerdasan buatan menggunakan *Fuzzy Sugeno* pada AI NPC yang berperan sebagai *player game*.



Gambar 3.1 Blok diagram prosedur penelitian

3.1 Fokus Penelitian

Penelitian yang sedang dikerjakan akan meneruskan penelitian sebelumnya. *Road map* secara garis besar dari penelitian yang telah dikerjakan serta dari penelitian yang sedang dikerjakan ditunjukkan pada gambar 3.2. Berikut gambarannya:



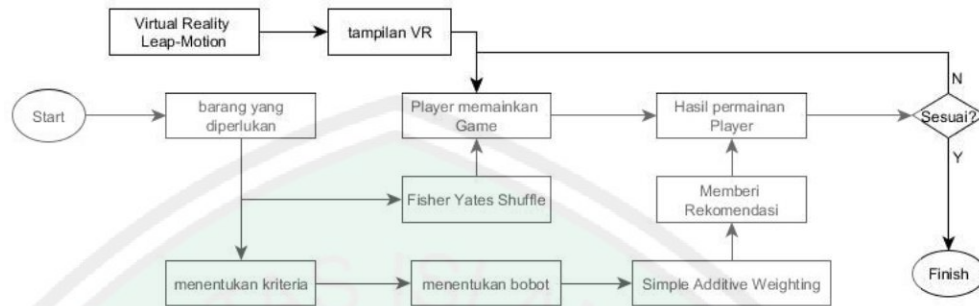
Gambar 3.2 Roadmap Penelitian

Roadmap pada gambar 3.2 menunjukkan bahwa fokus utama penelitian yang dilakukan pada pengerjaan *Dynamic Difficulty Adjustment* pra-bencana yang mengacu pada *gameplay* penelitian Abdul Rahman yang berbasis VR. Namun pada penelitian ini tema *game* yang digunakan adalah *game adventure* dengan menggunakan objek Gunung Kelud dan fitur kabut yang digunakan diambil dari penelitian Berlian Gita Cahyani sebagai tingkat kesulitan di dalam *game*.

3.2 Desain Sistem Game

Pembuatan *game* akan dibuat dengan *gameplay* mengacu pada penelitian Abdul Rahman (2019), pada penelitiannya peneliti membuat *game* simulasi untuk menentukan rekomendasi barang yang diperlukan ketika terjadinya bencana, dimana *gameplay game* adalah menjalankan misi dengan mengumpulkan barang-barang yang sangat penting ketika terjadinya bencana

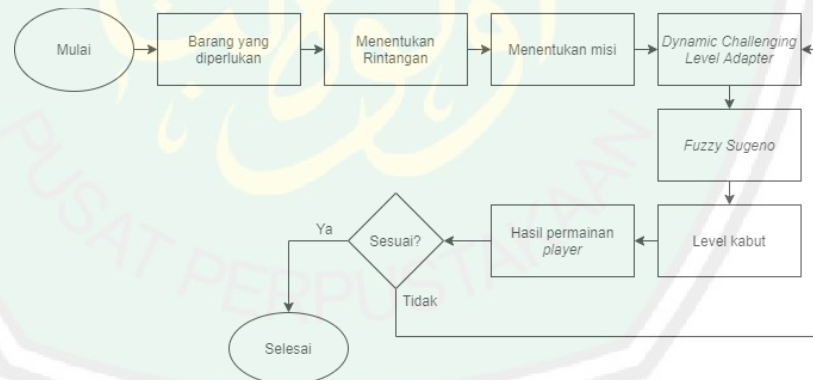
alam yang menggunakan basic *Virtual Reality*. Metode yang digunakan dalam penelitiannya menerapkan metode *Simple Additive Weighting*.



Gambar 3.3 Desain sistem *game* pada penelitian sebelumnya

Sumber: (Rahman, 2019)

Desain sistem *game* yang dibuat penelitian sebelumnya oleh Rahman (2019) akan diambil beberapa blok untuk sistem *game* yang akan dibuat. Desain sistem *game* yang akan dibuat dengan genre *adventure*, dapat dilihat pada gambar 3.4 sebagai berikut:



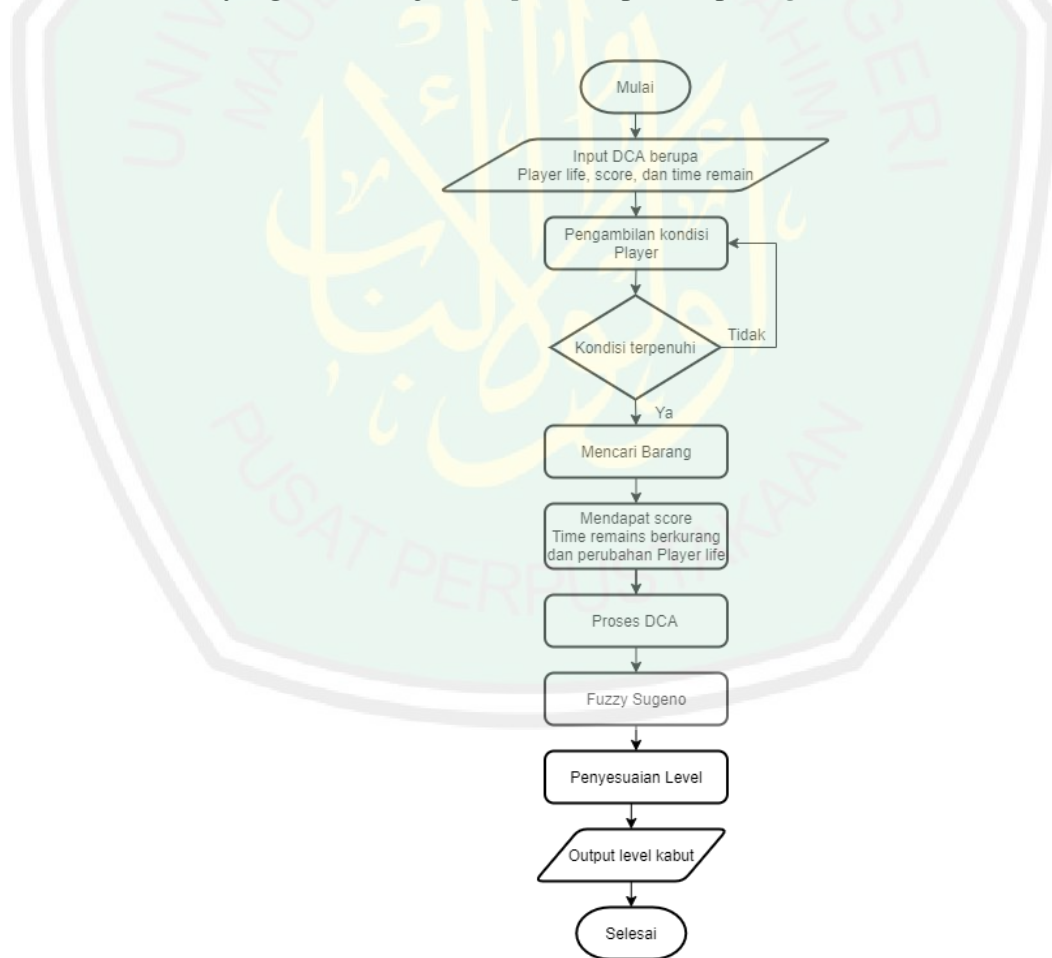
Gambar 3.4 Desain sistem *game* yang akan dibuat

Untuk fitur kabut atau objek kabut yang dihasilkan oleh Berlian Gita Cahyani (2019) yang digunakan sebagai penentu *level* atau tingkat kesulitan di dalam *game* yang akan dibuat. Sehingga setiap level perubahan kabut

dibatasi kemunculannya untuk memberikan batasan agar kabut yang dihasilkan berbeda seiring tingkat perilaku *player*.

3.3 Desain Sistem DCA pada *Game* Menggunakan *Fuzzy Sugeno*

Penerapan metode *Dynamic Challenging Level Adapter* di dalam *game* akan ditambahkan kecerdasan buatan sebagai penentu kesulitan serta menggunakan fitur kabut yang telah dibuat pada penelitian sebelumnya. AI yang ditambahkan pada penelitian ini menggunakan metode *Fuzzy Sugeno*, dimana metode *Fuzzy Sugeno* digunakan sebagai penentu keputusan fitur kabut yang akan menjadi *output* setiap level pada *game*.



Gambar 3.5 Desain sistem *game* DCA menggunakan *Fuzzy Sugeno*

Penambahan variable seperti *Score*, *Time remain*, *Player life*, dan Rintangan di dalam *game* yang akan dibuat dalam penelitian ini digunakan untuk menyesuaikan atau mempengaruhi perubahan tingkat kesulitan kabut sesuai *level* pada *game*. Dari ketiga variable *input* yang ditentukan, kemudian metode AI menggunakan *Fuzzy Sugeno* akan bekerja memproses keputusan *output* selanjutnya sampai *game* selesai. DCA akan memproses secara dinamis dengan menggunakan *Fuzzy Sugeno* yang berperan untuk mengatur *rule* yang akan ditentukan terlebih dahulu.

3.4 Skenario *Game*

3.4.1 Storyline *Game*

Game ini menceritakan seseorang yang akan bersiap-siap mendaki di Gunung Kelud. Pemain harus mencari barang-barang yang perlu dibawa saat akan mendaki ke Gunung Kelud. Disini *player* yang digunakan adalah AI NPC. Berikut adalah skenario *game* yang dibuat:

- Pertama *player* bergerak mencari barang-barang yang diperlukan untuk mendaki di Gunung Kelud
- Setelah itu *player* mengumpulkan barang-barang sesuai apa yang dibutuhkan.
- Selanjutnya *player* akan mendapatkan poin *score* sesuai apa yang didapatkan di dalam *game*.
- *Level* kabut di dalam *game* berubah-ubah seiring waktu sesuai dengan kemampuan *player* mendapatkan *score* saat mengambil barang.

- Setelah mendapatkan barang-barang yang dibutuhkan permainan selesai dengan *score* yang mencapai target.
- Permainan dianggap *win* ketika *score* telah mencapai target yang ditentukan dan waktu tidak habis.
- Permainan dianggap *lose* ketika waktu untuk mengumpulkan barang telah habis dan tidak mencapai *score* target yang ditentukan.

3.5 *Dynamic Challenging Level Adapter*

Untuk menggambarkan dengan jelas sistem DCA perlu adanya pengujian untuk menentukan seberapa dinamis level permainan yang dihasilkan. Rumus tingkat tantangan atau *Challenging Rate* (CR) untuk mewakili tingkat kesulitan permainan secara keseluruhan ditunjukkan sebagai persamaan berikut (3.1).

$$CR = \frac{T}{a} + \frac{H}{b} + \frac{P}{c} \quad (3.1)$$

Penelitian sebelumnya data rumus CR digunakan pada *game* RTS atau *Real Time Strategy* dengan parameter serta konstanta yang berbeda sesuai dengan *input* yang berhubungan. Dalam rumus dari persamaan (3.1) tersebut, parameter T (*Time*) mewakili sisa waktu yang ditempuh dalam permainan, parameter H (*Health*) mewakili jumlah titik kesehatan *player* selama bermain, dan parameter P (*Point*) mewakili jumlah seluruh *point* atau *score* yang diperoleh. Sedangkan a, b, dan c adalah konstanta untuk normalisasi dalam rumus tersebut. Masing masing konstanta tersebut dikonfigurasi dengan

nilai 300, 100, dan 2. Berikut adalah pengujian DCA dengan nilai CR tiap level, mulai dari level 1 sampai 10.

Tabel 3.1 Pengujian DCA dengan nilai CR tiap level

| Level | CR | | | Hasil |
|----------|------|--------------|------|--------------|
| Level 1 | 0 | 1 | 1 | 2 |
| Level 2 | 2,5 | 0,9333333333 | 0,9 | 4,333333333 |
| Level 3 | 5 | 0,8666666667 | 0,8 | 6,666666667 |
| Level 4 | 7,5 | 0,8 | 0,7 | 9 |
| Level 5 | 10 | 0,7333333333 | 0,6 | 11,333333333 |
| Level 6 | 12,5 | 0,6666666667 | 0,5 | 13,666666667 |
| Level 7 | 15 | 0,6 | 0,4 | 16 |
| Level 8 | 17,5 | 0,5333333333 | 0,3 | 18,333333333 |
| Level 9 | 20 | 0,4666666667 | 0,2 | 20,666666667 |
| Level 10 | 22,5 | 0,4 | 0,15 | 23,05 |

Nilai hasil perhitungan menggunakan rumus CR pada tabel 3.1 kemudian akan diuji kembali dengan menggunakan *Fuzzy Sugeno* untuk memilih keputusan level yang dihasilkan selama *player* bermain *game*.

3.6 DCA Menggunakan *Fuzzy Sugeno*

3.6.1 Penentuan Nilai *Input* Level

Keberadaan kabut sebagai penentu level dalam *game* ini dibatasi. Gunanya untuk mengetahui setiap perubahan yang dicapai oleh *player* selama bermain *game*. Sehingga setiap perubahan *input* variabel yang dihasilkan *player* akan menentukan kabut yang dihasilkan. Berikut adalah tabel pembagian tingkat level kabut yang akan dihasilkan dengan *input* tiga variabel:

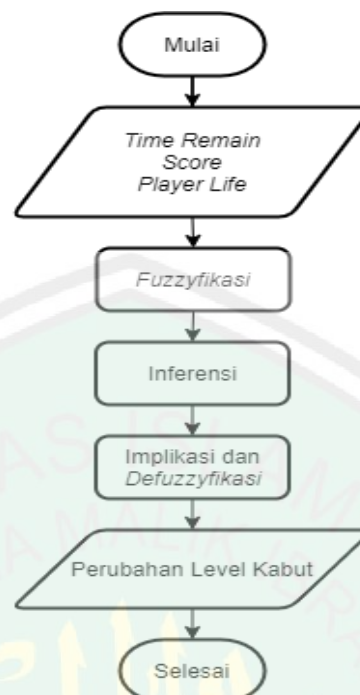
Tabel 3.2 Pembagian nilai *input* level

| <i>Output</i> | <i>Input</i> | | |
|---------------|--------------|--------------------|--------------------|
| | <i>Score</i> | <i>Time Remain</i> | <i>Player Life</i> |
| Level 1 | 0 | 300 | 100 |
| Level 2 | 5 | 280 | 90 |
| Level 3 | 10 | 260 | 80 |
| Level 4 | 15 | 240 | 70 |
| Level 5 | 20 | 220 | 60 |
| Level 6 | 25 | 200 | 50 |
| Level 7 | 30 | 180 | 40 |
| Level 8 | 35 | 160 | 30 |
| Level 9 | 40 | 140 | 20 |
| Level 10 | 45 | 120 | 15 |

Dari tabel 3.2 nilai yang ditentukan pada ketiga *input* tersebut akan menentukan level kabut yang dihasilkan, kemudian akan di proses dengan *Fuzzy Sugeno*.

3.6.2 Perancangan Fuzzy

Penerapan algoritma diimplementasikan pada penentuan kabut. Algoritma yang digunakan adalah algoritma *Fuzzy Sugeno* dengan orde nol. Pada tahap ini akan menentukan *output difficulty* dari permainan petualangan Gunung Kelud yang berupa kabut dengan menggunakan *Fuzzy Sugeno*. Diagram alur perancangan *fuzzy* ditunjukkan pada Gambar 3.6.



Gambar 3.6 Diagram alir fuzzy pada game

Berdasarkan gambar 3.6 proses perhitungan *Fuzzy Sugeno* dengan masukan variabel *Time Remaining*, *Score*, dan *NPC's Life*. Selanjutnya dimulailah proses *fuzzyfikasi*, *inferensi*, dan *defuzzyfikasi* yang kemudian menghasilkan *output* nilai dari *level kabut*.

3.6.2.1 Fungsi Keanggotaan

A. Nilai Linguistik Input

1. *Score*

Variabel *input Score* dibagi ke dalam tiga notasi yaitu rendah, sedang, dan tinggi. Pembagian fungsi keanggotaan diantaranya trapesium turun dengan *range* rendah, segitiga dengan *range* sedang, dan trapesium naik dengan *range* tinggi. Untuk tabel nilai linguistik dari himpunan *fuzzy input* variabel *score* ditunjukkan pada tabel 3.3.

Tabel 3.3 Nilai linguistik *input* variabel *score*

| <i>Input Score</i> | | |
|--------------------|--------|-------|
| Variabel | Notasi | Range |
| RD | Rendah | 0-15 |
| SD | Sedang | 15-35 |
| TG | Tinggi | 35-45 |

2. *Time Remain*

Time Remain atau waktu sisa permainan dibagi menjadi tiga notasi yaitu sedikit, sedang, dan banyak dengan satuan detik, dimana pembagian fungsi keanggotaan trapesium turun dengan *range* sedikit, segitiga dengan *range* sedang, dan trapesium naik dengan *range* banyak. Tabel nilai linguistik dari himpunan *fuzzy input* variabel *time remain* ditunjukkan pada tabel 3.4.

Tabel 3.4 Nilai linguistik *input* variabel *time remain*

| <i>Input Time Remain</i> | | |
|--------------------------|---------|---------|
| Variabel | Notasi | Range |
| SK | Sedikit | 0-160 |
| SD | Sedang | 160-240 |
| BY | Banyak | 240-300 |

3. *Player life*

Player life merupakan nyawa *player* dalam *game* untuk dijadikan *input* yang dibagi menjadi tiga notasi yaitu sedikit, sedang, dan penuh. Pembagian fungsi keanggotaan dari *input player life* diantaranya, trapesium turun dengan *range* sedikit, segitiga dengan *range* sedang,

dan trapesium naik dengan *range* banyak. Himpunan *fuzzy* dari *input* variabel *player life* ditunjukkan pada tabel 3.5.

Tabel 3.5 Nilai linguistik *input player life*

| <i>Input Player life</i> | | |
|--------------------------|---------|--------|
| Variabel | Notasi | Range |
| SK | Sedikit | 0-30 |
| SD | Sedang | 30-70 |
| BY | Banyak | 70-100 |

B. Nilai Linguistik *Output*

1. Kabut

Kabut disini disebut sebagai variable *output* yaitu berupa tingkat perubahan kabut dengan pembagian menjadi 10 notasi. Untuk menentukan *output* yang tegas tiap masing-masing rule berdasarkan *fuzzy* yang disusun bernilai konstan. Berikut pembagian nilai konstan pada tabel 3.6:

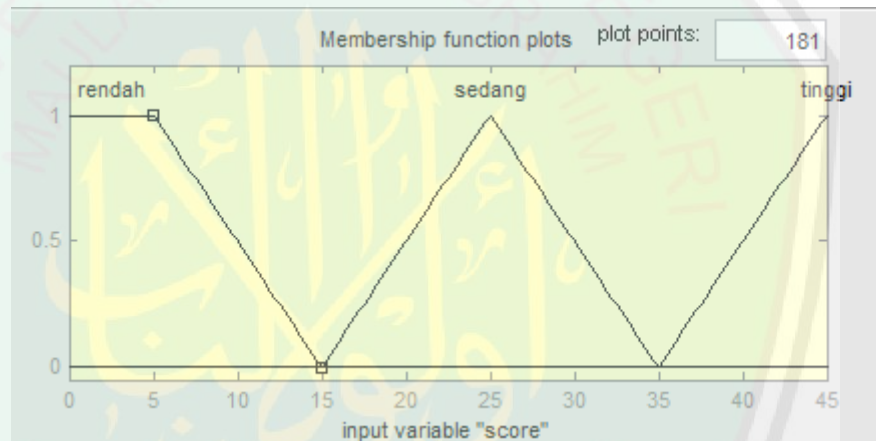
Tabel 3.6 Nilai linguistik *output* variabel kabut

| Konstanta | Notasi |
|-----------|------------------------|
| 1 | <i>Extremely clear</i> |
| 2 | Sangat jernih |
| 3 | Jernih |
| 4 | <i>Light haze</i> |
| 5 | <i>Haze</i> |
| 6 | <i>Thin fog</i> |
| 7 | <i>Light fog</i> |
| 8 | <i>Moderate fog</i> |
| 9 | <i>Thick fog</i> |
| 10 | <i>Dense fog</i> |

3.6.2.2 Fuzzyfikasi

Fuzzyfikasi adalah proses memetakan nilai *crisp* (numerik) kedalam himpunan *fuzzy* dan menentukan derajat keanggotaannya. Perhitungan nilai *fuzzyfikasi* didapatkan dari beberapa fungsi persamaan, fungsi persamaan yang digunakan pada tiap variabel ada 3, yaitu fungsi trapesium turun, segitiga, dan trapesium naik dari tiap variabel nilai linguistik. Berikut perhitungan manual dari ketiga fungsi tiap variabel:

1. Score



Gambar 3.7 Fungsi keanggotaan *input score*

Trapesium Turun : Rendah

$$\mu_{rendah}[x] = \begin{cases} 0; & x \geq 15 \\ 1; & x \leq 5 \\ \frac{15-x}{15-5}; & 5 < x < 15 \end{cases} \quad (3.1)$$

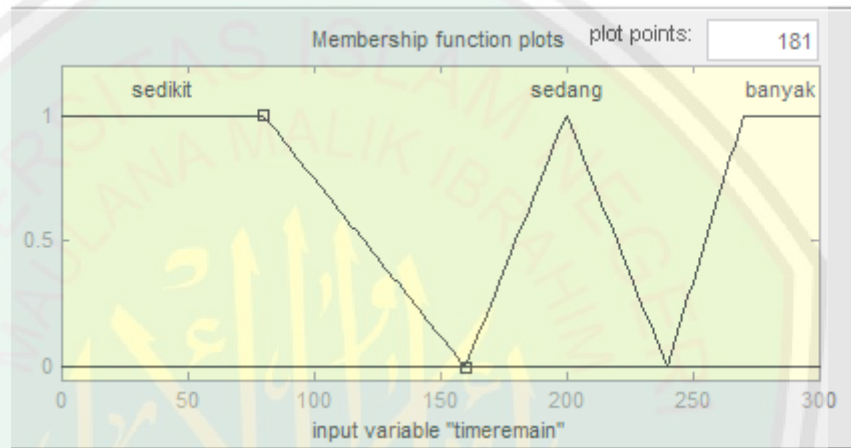
Segitiga : Sedang

$$\mu_{sedang}[x] = \begin{cases} 0; & x \leq 15 \text{ atau } x \geq 35 \\ \frac{x-15}{25-15}; & 15 < x \leq 25 \\ \frac{35-x}{35-25}; & 25 \leq x < 35 \end{cases} \quad (3.2)$$

Trapesium Naik : Tinggi

$$\mu_{tinggi}[x] = \begin{cases} 0; & x \leq 35 \\ \frac{x-35}{40-35}; & 35 < x \leq 45 \\ 1; & x \geq 45 \end{cases} \quad (3.3)$$

2. Time Remain



Gambar 3.8 Fungsi keanggotaan *input time remain*

Trapesium Turun : Sedikit

$$\mu_{sedikit}[x] = \begin{cases} 0; & x \geq 160 \\ 1; & x \leq 80 \\ \frac{160-x}{160-80}; & 80 < x < 160 \end{cases} \quad (3.4)$$

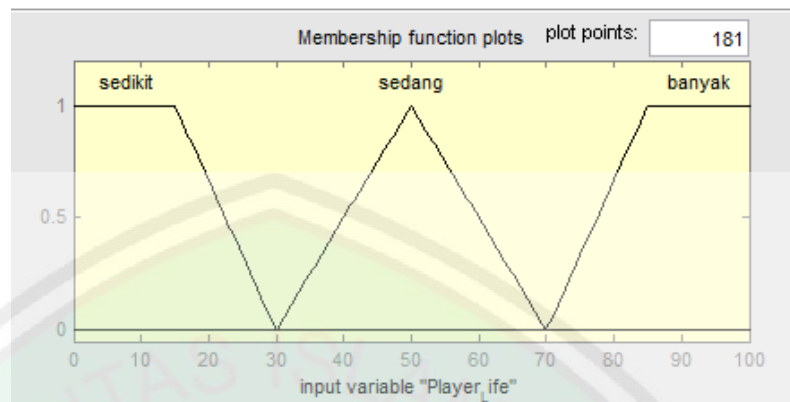
Segitiga : Sedang

$$\mu_{sedang}[x] = \begin{cases} 0; & x \leq 160 \text{ atau } x \geq 240 \\ \frac{x-160}{200-160}; & 160 < x \leq 200 \\ \frac{240-x}{240-200}; & 200 \leq x < 240 \end{cases} \quad (3.5)$$

Trapesium Naik : Banyak

$$\mu_{banyak}[x] = \begin{cases} 0; & x \leq 240 \\ \frac{x-240}{270-240}; & 240 < x \leq 270 \\ 1; & x \geq 300 \end{cases} \quad (3.6)$$

3. *Player life*



Gambar 3.9 Fungsi keanggotaan *input player_life*

Trapesium Turun : Sedikit

$$\mu_{sedikit}[x] = \begin{cases} 0; & x \geq 30 \\ 1; & x \leq 15 \\ \frac{30-x}{30-15}; & 15 < x < 30 \end{cases} \quad (3.7)$$

Segitiga : Sedang

$$\mu_{sedang}[x] = \begin{cases} 0; & x \leq 30 \text{ atau } x \geq 70 \\ \frac{x-30}{50-30}; & 30 < x \leq 50 \\ \frac{70-x}{70-50}; & 50 \leq x < 70 \end{cases} \quad (3.8)$$

Trapesium Naik : Banyak

$$\mu_{penuh}[x] = \begin{cases} 0; & x \leq 70 \\ \frac{x-70}{85-70}; & 70 < x \leq 85 \\ 1; & x \geq 100 \end{cases} \quad (3.9)$$

3.6.2.3 *Fuzzy Rules*

Aturan-aturan atau *rules* yang digunakan diperoleh dari pengetahuan penulis yang kemudian diterapkan dalam sebuah *rule base*. *Rules* ini yang nantinya digunakan sebagai aturan dasar dari logika *fuzzy* (Ahmadi et al.,

2018). Kaidah *fuzzy (rules)* yang diterapkan dalam penentuan kabut berjumlah 27 *rules*. Tabel 3.7 berikut menjelaskan *fuzzy rules*:

Tabel 3.7 *Rules fuzzy* penentuan *output*

| <i>Rules</i> | <i>Input</i> | | | <i>Output</i> |
|--------------|--------------|--------------------|--------------------|------------------------|
| | <i>Score</i> | <i>Time Remain</i> | <i>Player Life</i> | <i>Kabut</i> |
| 1 | Rendah | Banyak | Banyak | <i>Extremely clear</i> |
| 2 | Rendah | Banyak | Sedang | Sangat jernih |
| 3 | Rendah | Banyak | Sedikit | Jernih |
| 4 | Rendah | Sedang | Banyak | Jernih |
| 5 | Rendah | Sedang | Sedang | <i>Extremely clear</i> |
| 6 | Rendah | Sedang | Sedikit | Sangat jernih |
| 7 | Rendah | Sedikit | Banyak | Sangat jernih |
| 8 | Rendah | Sedikit | Sedang | Jernih |
| 9 | Rendah | Sedikit | Sedikit | <i>Extremely clear</i> |
| 10 | Sedang | Banyak | Banyak | <i>Light haze</i> |
| 11 | Sedang | Banyak | Sedang | <i>Haze</i> |
| 12 | Sedang | Banyak | Sedikit | <i>Thin fog</i> |
| 13 | Sedang | Sedang | Banyak | <i>Light fog</i> |
| 14 | Sedang | Sedang | Sedang | <i>Thin fog</i> |
| 15 | Sedang | Sedang | Sedikit | <i>Haze</i> |
| 16 | Sedang | Sedikit | Banyak | <i>Light fog</i> |
| 17 | Sedang | Sedikit | Sedang | <i>Thin fog</i> |
| 18 | Sedang | Sedikit | Sedikit | <i>Light haze</i> |
| 19 | Tinggi | Banyak | Banyak | <i>Dense fog</i> |
| 20 | Tinggi | Banyak | Sedang | <i>Moderate fog</i> |
| 21 | Tinggi | Banyak | Sedikit | <i>Thick fog</i> |
| 22 | Tinggi | Sedang | Banyak | <i>Thick fog</i> |
| 23 | Tinggi | Sedang | Sedang | <i>Light fog</i> |
| 24 | Tinggi | Sedang | Sedikit | <i>Moderate fog</i> |
| 25 | Tinggi | Sedikit | Banyak | <i>Moderate fog</i> |
| 26 | Tinggi | Sedikit | Sedang | <i>Thick fog</i> |
| 27 | Tinggi | Sedikit | Sedikit | <i>Dense fog</i> |

Rules fuzzy pada tabel 3.7 diatas, dapat diperoleh hubungan antara *input* dan *output* dari variabel linguistik, kemudian dijadikan *rule fuzzy*

$$\mathbf{IF\ } x \mathbf{ is\ } a \mathbf{ AND\ } y \mathbf{ is\ } b \mathbf{ THEN\ } z \mathbf{ is\ } c \quad (3.10)$$

Yang tersusun sebagai berikut:

- 1) *If (Score is Rendah) and (Time Remain is Banyak) and (Player Life is Banyak) then (Kabut is Extremely clear)*
- 2) *If (Score is Rendah) and (Time Remain is Banyak) and (Player Life is Sedang) then (Kabut is Sangat jernih)*
- 3) *If (Score is Rendah) and (Time Remain is Banyak) and (Player Life is Sedikit) then (Kabut is Jernih)*
- 4) *If (Score is Rendah) and (Time Remain is Sedang) and (Player Life is Banyak) then (Kabut is Jernih)*
- 5) *If (Score is Rendah) and (Time Remain is Sedang) and (Player Life is Sedang) then (Kabut is Extremely clear)*
- 6) *If (Score is Rendah) and (Time Remain is Sedang) and (Player Life is Sedikit) then (Kabut is Sangat jernih)*
- 7) *If (Score is Rendah) and (Time Remain is Sedikit) and (Player Life is Banyak) then (Kabut is Sangat jernih)*
- 8) *If (Score is Rendah) and (Time Remain is Sedikit) and (Player Life is Sedang) then (Kabut is Jernih)*
- 9) *If (Score is Rendah) and (Time Remain is Sedikit) and (Player Life is Sedikit) then (Kabut is Extremely clear)*
- 10) *If (Score is Sedang) and (Time Remain is Banyak) and (Player Life is Banyak) then (Kabut is Light haze)*
- 11) *If (Score is Sedang) and (Time Remain is Banyak) and (Player Life is Sedang) then (Kabut is Haze)*

- 12) *If (Score is Sedang) and (Time Remain is Banyak) and (Player Life is Sedikit) then (Kabut is Thin fog)*
- 13) *If (Score is Sedang) and (Time Remain is Sedang) and (Player Life is Banyak) then (Kabut is Light fog)*
- 14) *If (Score is Sedang) and (Time Remain is Sedang) and (Player Life is Sedang) then (Kabut is Thint fog)*
- 15) *If (Score is Sedang) and (Time Remain is Sedang) and (Player Life is Sedikit) then (Kabut is Haze)*
- 16) *If (Score is Sedang) and (Time Remain is Sedikit) and (Player Life is Banyak) then (Kabut is Light fog)*
- 17) *If (Score is Sedang) and (Time Remain is Sedikit) and (Player Life is Sedang) then (Kabut is Thin fog)*
- 18) *If (Score is Sedang) and (Time Remain is Sedikit) and (Player Life is Sedikit) then (Kabut is Light fog)*
- 19) *If (Score is Tinggi) and (Time Remain is Banyak) and (Player Life is Banyak) then (Kabut is Dense fog)*
- 20) *If (Score is Tinggi) and (Time Remain is Banyak) and (Player Life is Sedang) then (Kabut is Moderate fog)*
- 21) *If (Score is Tinggi) and (Time Remain is Banyak) and (Player Life is Sedikit) then (Level Kabut is Thick fog)*
- 22) *If (Score is Tinggi) and (Time Remain is Sedang) and (Player Life is Banyak) then (Level Kabut is Thick fog)*

23) *If (Score is Tinggi) and (Time Remain is Sedang) and (Player Life is Sedang) then (Level Kabut is Light fog)*

24) *If (Score is Tinggi) and (Time Remain is Sedang) and (Player Life is Sedikit) then (Level Kabut is Moderate fog)*

25) *If (Score is Tinggi) and (Time Remain is Sedikit) and (Player Life is Banyak) then (Kabut is Moderate fog)*

26) *If (Score is Tinggi) and (Time Remain is Sedikit) and (Player Life is Sedang) then (Kabut is Thick fog)*

27) *If (Score is Tinggi) and (Time Remain is Sedikit) and (Player Life is Sedikit) then (Kabut is Dense fog)*

Sehingga untuk menentukan *output* yang akan dikeluarkan untuk memperoleh keputusan, dapat dilakukan dengan menghitung dengan tahap selanjutnya, yaitu implikasi dan *defuzzyfikasi*.

3.6.2.4 Implikasi dan Defuzzyfikasi

Defuzzyfikasi adalah langkah terakhir dalam suatu sistem logika *fuzzy* yang bertujuan mengkonversi setiap hasil yang kemudian diekspresikan dalam bentuk *fuzzy* kesuatu bilangan *real*. Fungsi implikasi yang digunakan yaitu fungsi implikasi MIN dengan rumus:

$$a_n = \mu_{predikatRn} = \min(\mu_{[x]}; \mu_{[y]}) \quad (3.11)$$

Selanjutnya dilakukan proses *defuzzyfikasi* dengan menggunakan rumus di bawah ini:

$$Z = \frac{\sum a_i * z_i}{\sum a_i} \quad (3.12)$$

3.7 Contoh Penentuan Level DCA Menggunakan *Fuzzy Sugeno*

Berikut adalah studi kasus DCA menggunakan *Fuzzy Sugeno* untuk keputusan level kabut: Contoh kasus, misal *player* bermain *game* dengan sisa darah 75, *time remain* yang tersisa 60 detik dengan perolehan *score* 25. Dengan *Player life* bernilai 75, *time remain* bernilai 60, dan *score* bernilai 25. Dilakukan perhitungan nilai CR terlebih dahulu untuk selanjutnya diproses menggunakan *fuzzy*. Berikut perhitungan nilai CR menggunakan rumus pada persamaan (3.1).

$$T = 60$$

$$H = 75$$

$$P = 25$$

$$CR = \frac{60}{300} + \frac{75}{100} + \frac{25}{2} = 13,45$$

Nilai CR yang dihasilkan adalah 13,45 dengan *range* level yang dihasilkan antara level 5 dan level 7 sesuai perhitungan nilai CR pada tabel 3.1 sebelumnya. Untuk mengetahui *output* level kabut yang sesuai, maka selanjutnya dilakukan perhitungan *Fuzzy Sugeno*, perhitungannya adalah sebagai berikut:

1. *Fuzzyfikasi*

Berdasarkan perhitungan *rule fuzzy* sebelumnya, maka diperoleh nilai dari proses *fuzzyfikasi* tiap variabel *input* sebagai berikut:

- *Score*

Perhitungan *fuzzifikasi input* variabel *score* dengan nilai 25:

$$\mu_{sedang} [25] = \frac{25-15}{25-15}; 15 \leq score \leq 25 = 1$$

- *Time Remain*

Perhitungan *fuzzifikasi input* variabel *time remain* dengan nilai 60:

$$\mu_{sedikit} [60] = \frac{160-60}{160-80}; 80 < timeremain \leq 160 = 1,25$$

- *Player Life*

Perhitungan *fuzzifikasi input* variabel *player life* dengan nilai 70:

$$\mu_{banyak} [75] = \frac{75-70}{85-70}; 70 < pleyerlife \leq 85 = 0,33$$

2. Implikasi

Berdasarkan *rule fuzzy*, dengan menggunakan rumus implikasi min maka sesuai kondisi tersebut diperoleh:

1. $\alpha_1 = \mu_{R1} = \min(0; 0; 0,33) = 0$
2. $\alpha_2 = \mu_{R2} = \min(0; 0; 0) = 0$
3. $\alpha_3 = \mu_{R3} = \min(0; 0; 0) = 0$
4. $\alpha_4 = \mu_{R4} = \min(0; 0; 0,33) = 0$
5. $\alpha_5 = \mu_{R5} = \min(0; 0; 0) = 0$
6. $\alpha_6 = \mu_{R6} = \min(0; 0; 0) = 0$
7. $\alpha_7 = \mu_{R7} = \min(0; 1,25; 0,33) = 0$
8. $\alpha_8 = \mu_{R8} = \min(0; 1,25; 0) = 0$
9. $\alpha_9 = \mu_{R9} = \min(0; 1,25; 0) = 0$
10. $\alpha_{10} = \mu_{R10} = \min(1; 0; 0,33) = 0$
11. $\alpha_{11} = \mu_{R11} = \min(1; 0; 0) = 0$
12. $\alpha_{12} = \mu_{R12} = \min(1; 0; 0) = 0$
13. $\alpha_{13} = \mu_{R13} = \min(1; 0; 0,33) = 0$

$$14. \alpha_{14} = \mu_{R14} = \min(1; 0; 0) = 0$$

$$15. \alpha_{15} = \mu_{R15} = \min(1; 0; 0) = 0$$

$$16. \alpha_{16} = \mu_{R16} = \min(1; 1,25; 0,33) = 0,33$$

$$17. \alpha_{17} = \mu_{R17} = \min(1; 1,25; 0) = 0$$

$$18. \alpha_{18} = \mu_{R18} = \min(1; 1,25; 0) = 0$$

$$19. \alpha_{19} = \mu_{R19} = \min(0; 0; 0,33) = 0$$

$$20. \alpha_{20} = \mu_{R20} = \min(0; 0; 0) = 0$$

$$21. \alpha_{21} = \mu_{R21} = \min(0; 0; 0) = 0$$

$$22. \alpha_{22} = \mu_{R22} = \min(0; 0; 0,33) = 0$$

$$23. \alpha_{23} = \mu_{R23} = \min(0; 0; 0) = 0$$

$$24. \alpha_{24} = \mu_{R24} = \min(0; 0; 0) = 0$$

$$25. \alpha_{25} = \mu_{R25} = \min(0; 1,25; 0,33) = 0$$

$$26. \alpha_{26} = \mu_{R26} = \min(0; 1,25; 0) = 0$$

$$27. \alpha_{27} = \mu_{R27} = \min(0; 1,25; 0) = 0$$

3. Defuzzifikasi

Selanjutnya untuk menentukan nilai *output* maka dilakukan perhitungan dengan menggunakan rumus *weighted average* seperti pada persamaan (3.12). Maka dari hasil tersebut, dikarenakan nilai lainnya sama dengan 0, kemudian mengambil nilai pada rule fuzzy ke 16 dengan nilai 0,33. Sehingga menghasilkan *output* kabut *Light fog* dengan nilai konstanta 7, maka penyelesaiannya adalah pada persamaan (3.13) berikut:

$$Z = \frac{0,33 \times 7}{0,33} = 7 \quad (3.13)$$

Diperoleh respon perubahan kabut pada saat *player life* sebesar 75, *time remain* 60 dan *score* 25 adalah menghasilkan *output Z* sama dengan 7 berupa perubahan kabut menjadi “*Thin fog*”, dimana kabut tersebut berada pada level 7.

3.8 Pengujian dan Perhitungan Akurasi, Presisi, *Recall*, *F-Measure*

Dari perancangan pengujian metode DCA menggunakan *Fuzzy Sugeno* kemudian dibandingkan dengan tanpa menggunakan metode DCA. Berikut adalah contoh perhitungan akurasi sebanyak 20 kali percobaan. Untuk menentukan nilai akurasi, presisi, recal, dan f-measure, peneliti membuat tabel percobaan dengan ketentuan terdapat level 1-10 dengan menggunakan 3 variabel *input* yang menghasilkan nilai kombinasi dengan contoh sebagai berikut:

Tabel 3.8 Contoh Kombinasi Variabel *Input*

| No | Variabel <i>Input</i> | | |
|----|-----------------------|--------------------|--------------------|
| | <i>Score</i> | <i>Time Remain</i> | <i>Player Life</i> |
| 1 | 9 | 269 | 87 |
| 2 | 14 | 256 | 84 |
| 3 | 15 | 248 | 81 |
| 4 | 16 | 229 | 81 |
| 5 | 16 | 221 | 80 |
| 6 | 20 | 219 | 79 |
| 7 | 24 | 184 | 74 |
| 8 | 24 | 173 | 70 |
| 9 | 26 | 150 | 63 |
| 10 | 28 | 147 | 62 |
| 11 | 34 | 141 | 59 |
| 12 | 37 | 140 | 55 |
| 13 | 37 | 136 | 52 |
| 14 | 39 | 131 | 49 |
| 15 | 46 | 127 | 44 |

| No | Variabel <i>Input</i> | | |
|----|-----------------------|--------------------|--------------------|
| | <i>Score</i> | <i>Time Remain</i> | <i>Player Life</i> |
| 16 | 46 | 124 | 43 |
| 17 | 46 | 123 | 37 |
| 18 | 47 | 116 | 32 |
| 19 | 48 | 115 | 30 |
| 20 | 48 | 114 | 29 |

Setelah kombinasi variabel *input* terpenuhi seperti pada tabel 3.8, selanjutnya dilakukan proses perhitungan DCA dengan mengambil nilai CR sebagai berikut:

Tabel 3.9 Hasil Perhitungan DCA

| No | Variabel <i>Input</i> | | | Total |
|----|-----------------------|--------------------|--------------------|--------------|
| | <i>Score</i> | <i>Time Remain</i> | <i>Player Life</i> | |
| 1 | 4,5 | 0,896666667 | 0,87 | 6,266666667 |
| 2 | 7 | 0,853333333 | 0,84 | 8,693333333 |
| 3 | 7,5 | 0,826666667 | 0,81 | 9,136666667 |
| 4 | 8 | 0,763333333 | 0,81 | 9,573333333 |
| 5 | 8 | 0,736666667 | 0,8 | 9,536666667 |
| 6 | 10 | 0,73 | 0,79 | 11,52 |
| 7 | 12 | 0,613333333 | 0,74 | 13,353333333 |
| 8 | 12 | 0,576666667 | 0,7 | 13,276666667 |
| 9 | 13 | 0,5 | 0,63 | 14,13 |
| 10 | 14 | 0,49 | 0,62 | 15,11 |
| 11 | 17 | 0,47 | 0,59 | 18,06 |
| 12 | 18,5 | 0,466666667 | 0,55 | 19,516666667 |
| 13 | 18,5 | 0,453333333 | 0,52 | 19,473333333 |
| 14 | 19,5 | 0,436666667 | 0,49 | 20,426666667 |
| 15 | 23 | 0,423333333 | 0,44 | 23,863333333 |
| 16 | 23 | 0,413333333 | 0,43 | 23,843333333 |
| 17 | 23 | 0,41 | 0,37 | 23,78 |
| 18 | 23,5 | 0,386666667 | 0,32 | 24,206666667 |
| 19 | 24 | 0,383333333 | 0,3 | 24,683333333 |
| 20 | 24 | 0,38 | 0,29 | 24,67 |

Dari hasil perhitungan sesuai tabel 3.9 tersebut akan menghasilkan level setiap nilai dari variabel *input*, kemudian akan dilakukan proses pengambilan keputusan menggunakan *rule fuzzy sugeno* dengan tetap mempertimbangkan nilai CR dari perhitungan DCA sebagai berikut:

Tabel 3.10 Hasil Akhir Level DCA

| No | Variabel Input | | | Total Perhitungan | Hasil Level | |
|----|----------------|-------------|-------------|-------------------|-------------------------|--|
| | Score | Time Remain | Player Life | | DCA Setelah Perhitungan | Hasil Akhir Level DCA Menggunakan Fuzzy Sugeno |
| 1 | 9 | 269 | 87 | 6,266666667 | 2 | 1 |
| 2 | 14 | 256 | 84 | 8,693333333 | 2 | 1 |
| 3 | 15 | 248 | 81 | 9,136666667 | 2 | 4 |
| 4 | 16 | 229 | 81 | 9,573333333 | 2 | 7 |
| 5 | 16 | 221 | 80 | 9,536666667 | 3 | 7 |
| 6 | 20 | 219 | 79 | 11,52 | 3 | 7 |
| 7 | 24 | 184 | 74 | 13,353333333 | 3 | 7 |
| 8 | 24 | 173 | 70 | 13,276666667 | 4 | 6 |
| 9 | 26 | 150 | 63 | 14,13 | 4 | 6 |
| 10 | 28 | 147 | 62 | 15,11 | 4 | 6 |
| 11 | 34 | 141 | 59 | 18,06 | 5 | 6 |
| 12 | 37 | 140 | 55 | 19,516666667 | 5 | 9 |
| 13 | 37 | 136 | 52 | 19,473333333 | 5 | 9 |
| 14 | 39 | 131 | 49 | 20,426666667 | 6 | 9 |
| 15 | 46 | 127 | 44 | 23,863333333 | 6 | 9 |
| 16 | 46 | 124 | 43 | 23,843333333 | 6 | 9 |
| 17 | 46 | 123 | 37 | 23,78 | 7 | 9 |
| 18 | 47 | 116 | 32 | 24,206666667 | 7 | 9 |
| 19 | 48 | 115 | 30 | 24,683333333 | 8 | 10 |
| 20 | 48 | 114 | 29 | 24,67 | 8 | 10 |

Hasil akhir level kabut telah diperoleh seperti pada tabel 3.9 dengan penerapan perhitungan CR dari DCA dan diolah kembali dengan bantuan *rule* dari *Fuzzy Sugeno*. Dari hasil akhir tersebut kemudian akan dibandingkan dengan sistem level tanpa menggunakan metode DCA sebagai berikut:

Tabel 3.11 Hasil Level menggunakan DCA dan Tanpa DCA

| No | Variabel Input | | | Hasil Level | |
|----|----------------|-------------|-------------|-------------|-----------|
| | Score | Time Remain | Player Life | DCA | Tanpa DCA |
| 1 | 9 | 269 | 87 | 1 | 2 |
| 2 | 14 | 256 | 84 | 1 | 3 |
| 3 | 15 | 248 | 81 | 4 | 4 |
| 4 | 16 | 229 | 81 | 7 | 4 |
| 5 | 16 | 221 | 80 | 7 | 4 |
| 6 | 20 | 219 | 79 | 7 | 5 |
| 7 | 24 | 184 | 74 | 7 | 5 |
| 8 | 24 | 173 | 70 | 6 | 5 |
| 9 | 26 | 150 | 63 | 6 | 6 |
| 10 | 28 | 147 | 62 | 6 | 6 |
| 11 | 34 | 141 | 59 | 6 | 7 |
| 12 | 37 | 140 | 55 | 9 | 8 |
| 13 | 37 | 136 | 52 | 9 | 8 |
| 14 | 39 | 131 | 49 | 9 | 8 |
| 15 | 46 | 127 | 44 | 9 | 10 |
| 16 | 46 | 124 | 43 | 9 | 10 |
| 17 | 46 | 123 | 37 | 9 | 10 |
| 18 | 47 | 116 | 32 | 9 | 10 |
| 19 | 48 | 115 | 30 | 10 | 10 |
| 20 | 48 | 114 | 29 | 10 | 10 |

Setelah hasil kedua sistem level diperoleh, kemudian dilakukan pengujian menggunakan *Confusion Matrix* yang digunakan untuk mengukur atau menghitung akurasi, presisi, *recall*, dan *f-measure* pada hasil sistem level di dalam *game* yang menggunakan metode DCA dengan hasil level tanpa metode. *Confusion Matrix* adalah mekanisme singkatan yang sangat kuat atau yang biasa disebut “*error matrix*”. *Confusion Matrix* menggambarkan bagaimana sampel yang termasuk dalam satu topik, cluster, atau kelas (baris dalam matriks) ditetapkan ke pluralitas topik, kelompok, atau kelas yang mungkin (Simske, 2019). Bentuk dari *confusion*

matrix sendiri berupa tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui. Berikut ini adalah gambar *confusion matrix* dengan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda.

| | | True/Actual Class | |
|-----------------|-----------|---------------------|---------------------|
| | | Positive (P) | Negative (N) |
| Predicted Class | True (T) | True Positive (TP) | False Positive (FP) |
| | False (F) | False Negative (FN) | True Negative (TN) |
| | | P=TP+FN | N=FP+TN |

Gambar 3.10 Contoh Ilustrasi *Confusion Matrix*

Sumber : Tharwat (2018)

Dari gambar 3.10 tersebut terdapat 4 istilah sebagai representasi hasil klasifikasi pada *confusion matrix*, diantaranya *True Positive* (TP) merupakan data positif yang diprediksi benar, *True Negative* (TN) merupakan data negatif yang diprediksi benar, *False Positive* (FP) merupakan data negatif namun diprediksi sebagai data positif, dan *False Negative* (FN) merupakan data positif namun diprediksi sebagai data negatif. *Confusion matrix* digunakan untuk menghitung berbagai *performance metrics* diantaranya yang sering digunakan yaitu, akurasi, presisi, *recall*, dan *f-measure*. Berikut adalah rumus dari akurasi, presisi, *recall*, dan *f-measure*:

a. Akurasi

Akurasi merupakan rasio prediksi benar (negatif dan positif) dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan persamaan 3.14.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.14)$$

b. *Precision*

Precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Nilai *precision* dapat diperoleh dengan persamaan 3.15.

$$Precision = \frac{TP}{TP+FP} \quad (3.15)$$

c. *Recall*

Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* dapat diperoleh dengan persamaan 3.16.

$$Recall = \frac{TP}{TP+FN} \quad (3.16)$$

d. *F-Measure*

F-Measure merupakan rasio nilai tengah atau rata-rata *precision* dan *recall*. Nilai *f-measure* dapat diperoleh dengan persamaan 3.17.

$$F - Measure = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (3.17)$$

Contoh proses perhitungan *confusion matrix* pada tabel 3.11 sebagai berikut:

Tabel 3.12 Hasil Level pada Tabel 3.11

| Hasil Level | |
|-------------|-----------|
| DCA | Tanpa DCA |
| 1 | 2 |
| 1 | 3 |
| 4 | 4 |
| 7 | 4 |
| 7 | 4 |
| 7 | 5 |
| 7 | 5 |
| 6 | 5 |
| 6 | 6 |

| Hasil Level | |
|-------------|-----------|
| DCA | Tanpa DCA |
| 6 | 6 |
| 6 | 7 |
| 9 | 8 |
| 9 | 8 |
| 9 | 8 |
| 9 | 10 |
| 9 | 10 |
| 9 | 10 |
| 9 | 10 |
| 10 | 10 |
| 10 | 10 |

Selanjutnya, pada tabel hasil level yang dikeluarkan pada tabel 3.12 dibuat *matrix* sebagai berikut:

Tabel 3.13 *Confusion Matrix* Hasil Level (FP)

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 4 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

Dari hasil perhitungan *Matrix Confusion* pada tabel 3.13, diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 3 data berjumlah 5 (kolom berwarna putih). Selanjutnya dilakukan perhitungan sebagai berikut:

- a. Perhitungan Akurasi

$$\text{Akurasi} = \text{TP} / \text{TP} + \text{TN} + \text{FP} + \text{FN}$$

$$\text{Akurasi} = 5 / 20$$

$$\text{Akurasi} = \mathbf{0,25}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu yang diambil dari data kolom (x) 1 sampai 10 secara menurun sebagai berikut:

$$\text{FP (1)} = 0 ; \text{FP (2)} = 1 ; \text{FP (3)} = 1 ; \text{FP (4)} = 2 ; \text{FP(5)} = 3 ; \text{FP(6)} = 0 ; \\ \text{FP(7)} = 1 ; \text{FP(8)} = 3 ; \text{FP(9)} = 0 ; \text{FP(10)} = 4$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{P(1)} = 0/(0+0) = 0 ; \text{P(2)} = 0/(0+1) = 0 ; \text{P(3)} = 0/(0+1) = 0 ; \text{P(4)} = 1/(1+2) \\ = 0,333333333 ; \text{P(5)} = 0/(0+3) = 0 ; \text{P(6)} = 2/(2+0) = 1 ; \text{P(7)} = 0/(0+1) = \\ 0 ; \text{P(8)} = 0/(0+3) = 0 ; \text{P(9)} = 0/(0+0) = 0 ; \text{P(10)} = 2/(2+4) = 0,333333333$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (\text{P(1)} + \text{P(2)} + \text{P(3)} + \text{P(4)} + \text{P(5)} + \text{P(6)} + \text{P(7)} + \text{P(8)} + \text{P(9)} + \\ \text{P(10)}) / \text{Jumlah kelas}$$

$$\text{Precision} = (0+0+0+0,333333333+0+1+0+0+0+0,333333333) / 10 =$$

$$\mathbf{0,166666667}$$

Tabel 3.14 *Confusion Matrix* Hasil Level (FN)

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 4 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

Pada tabel 3.14 di atas, untuk mengetahui nilai *False Negative* (FN) diambil dari data (x) 1 sampai 10 secara mendatar, sehingga diperoleh data sebagai berikut:

c. Perhitungan *Recall*

FN(1) = 2 ; FN(2) = 0 ; FN (3) = 0 ; FN (4) = 0 ; FN (5) = 0 ; FN (6) = 2;
FN (7) = 4 ; FN (8) = 0 ; FN (9) = 7 ; FN (10) = 0;

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 0/(0+2) = 0 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 2/(2+2) = 0,5 ; R(7) = 0/(0+4) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+7) = 0 ; R(10) = 2/(2+0) = 1$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0+0+0+1+0+0,5+0+0+0+1) / 10 = \mathbf{0,25}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * \mathbf{0,166666667} * \mathbf{0,25}) / (\mathbf{0,166666667} + \mathbf{0,25}) = \mathbf{0,2}$$

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

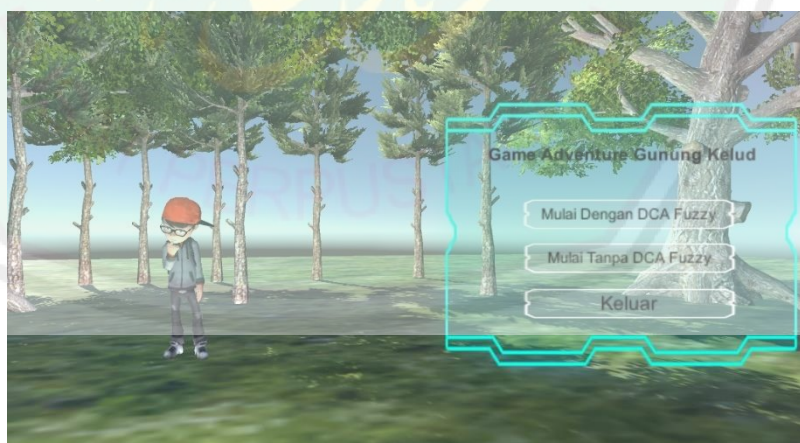
Bab ini membahas tentang implementasi dari perencanaan yang telah diajukan serta hasil pengujian dari implementasi metode *Dynamic Challenging Level Adapter* menggunakan *Fuzzy Sugeno* untuk menentukan tingkat kesulitan kabut atau *level* pada *game* dengan menggunakan NPC sebagai *player*.

4.2 Implementasi Tampilan *Game*

Pembuatan tampilan pada *game* ini, melibatkan beberapa komponen antar muka pada *game adventure* gunung kelud yaitu:

4.2.1 Tampilan Main Menu *Game*

Tampilan main menu *game* saat permainan dijalankan pertama kali sebagai berikut:



Gambar 4.1 Tampilan awal *main menu* pada *game*

Pada gambar 4.1, terdapat tampilan main menu pada *game* untuk memulai permainan yang berisikan tiga *button* yaitu *Mulai Dengan DCA Fuzzy*, *Mulai*

Tanpa *DCA Fuzzy* dan Keluar. Pada *button* Mulai Dengan *DCA Fuzzy* atau Mulai Tanpa *DCA Fuzzy* saat diklik akan menuju ke area sistem permainan yang menggunakan *DCA Fuzzy* atau tanpa *DCA Fuzzy*, sedangkan pada *button* Keluar ketika diklik akan keluar dari *game*.

4.2.2 Tampilan Awal *Game*

Pada gambar 4.2 tampilan awal saat memulai permainan terlihat pada suasana gunung yang dimana nanti seorang *player* NPC akan bergerak menjalankan misi mencari barang yang ada di dalam *game*. Saat *player* memasuki arena permainan, *player* akan mendapatkan batas waktu selama 300 detik dengan capaian *score target* sebanyak 60 dan memiliki kesehatan atau *Player Life* sebanyak 100.

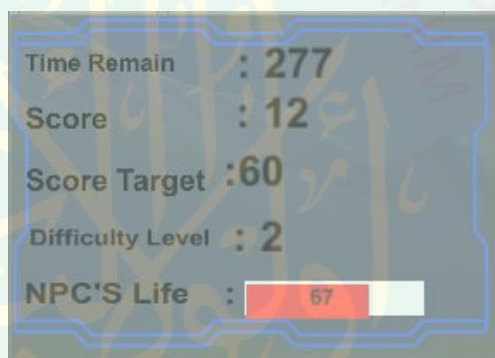


Gambar 4.2 Tampilan awal permainan

Gambar 4.2 di atas menunjukkan gambaran arena yang dimasuki oleh *player* pertama kali saat *game* berjalan. Terlihat arena yang berlatar termpat di Gunung Kelud yang terdapat beberapa objek terlihat dari jauh. Terdapat objek barang dan buah-buahan yang nantinya objek tersebut berperan sebagai sumber perolehan skor saat pemain menyentuh atau melewatinya dan terdapat objek hewan yang berperan sebagai rintangan permainan.

4.2.3 Tampilan Panel Variabel Dalam Game

Tampilan panel variabel pada gambar 4.3 menunjukkan adanya beberapa variabel yang dapat mempengaruhi perubahan kabut di dalam game. *Time remain* akan terus berkurang selama *player* bermain di dalam game, *score* akan bertambah sesuai yang di dapatkan *player* selama mencari barang, serta kesehatan atau *player life* akan berubah-ubah ketika *player* terkena rintangan dan akan mendapatkan tambahan darah saat *player* menemukan makanan atau obat. Peneliti menentukan setiap barang dengan 2 *point* untuk makanan dan 4 *point* untuk barang.



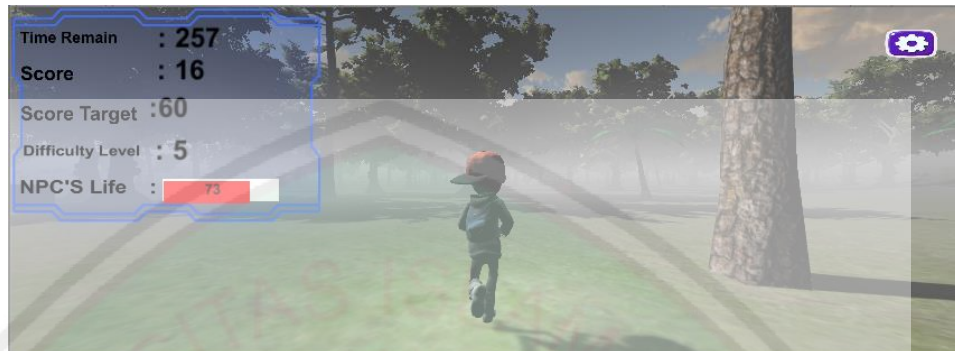
Gambar 4.3 Panel variabel di dalam game

Pada gambar 4.3 di atas, tampilan panel variabel di dalam game, di mana pada panel tersebut ada tiga variabel yang berpengaruh terhadap perubahan level kabut selama permainan berlangsung.

4.2.4 Tampilan Perubahan Level Kabut

Perubahan *level* kabut diperoleh sesuai *score* yang di dapatkan *player* selama menjalankan misi. Variabel yang terdapat dalam game saling berhubungan dalam mempengaruhi perubahan kabut yang sesuai saat kondisi

terpenuhi. Berikut pada gambar 4.4 *level* kabut akan berubah sesuai hasil yang dicapai oleh *player*.

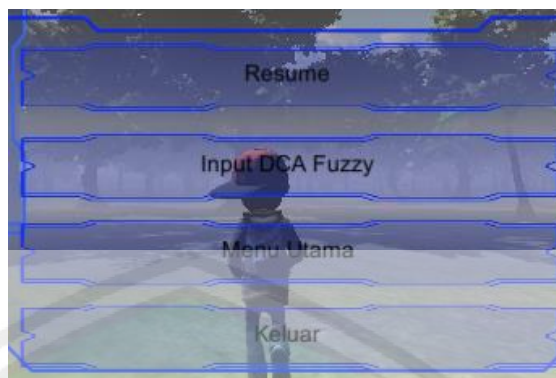


Gambar 4.4 Tampilan perubahan level kabut

Perubahan level kabut terlihat seperti pada gambar 4.4, semakin tinggi level yang dihasilkan maka kabut yang dikeluarkan akan semakin pekat atau tebal. Perubahan tersebut dipengaruhi oleh hasil permainan *player* mengumpulkan skor yang kemudian diakumulasikan dengan sisa waktu dan sisa darah *player*. Hasil yang diperoleh *player* dimasukkan ke dalam variable *Score*, *Time Remain*, dan *Player Life*, sehingga dapat menentukan level kabut yang dikeluarkan.

4.2.5 Tampilan *Canvas* Pada *Button Setting*

Tampilan *canvas button setting* pada gambar 4.5 terdapat empat tombol pilihan yaitu tombol *Resume* untuk kembali ke permainan, tombol *Input DCA Fuzzy* menuju ke *canvas* selanjutnya untuk *input* nilai variabel DCA, tombol *Menu Utama* untuk Kembali ke *main menu*, sedangkan tombol *Keluar* untuk keluar dari permainan.



Gambar 4.5 Tampilan *canvas* pada *button setting*

Pada gambar 4.5 di atas, tampilan panel *canvas* pada *button setting* digunakan untuk memilih pengaturan atau tindakan yang ingin dilakukan saat permainan telah dimulai. *User* atau pemain dapat menghentikan permainan dan memilih tindakan selanjutnya dengan memilih *button* atau tombol yang tersedia sesuai dengan penjelasan di atas.

4.2.6 Tampilan *Canvas Input Nilai Variabel Game*

Pada gambar 4.6 tampilan *canvas input* untuk menginputkan nilai-nilai tiap variabel yang digunakan pada sistem DCA menggunakan *Fuzzy* untuk mengetahui *level* kabut yang dihasilkan. Proses perhitungan akan dilakukan terlebih dahulu oleh sistem sebelum *level* ditentukan.

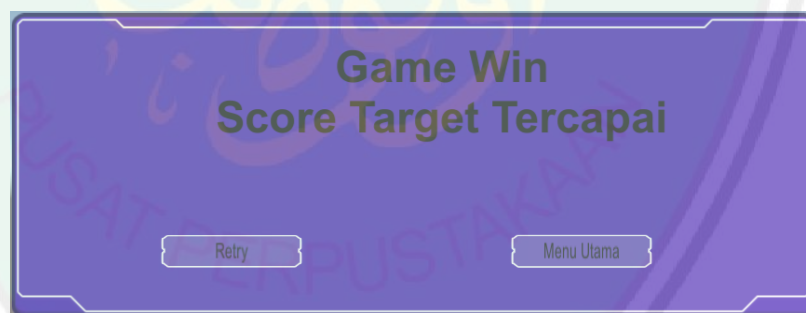


Gambar 4.6 Tampilan *canvas input* nilai variabel *DCA Fuzzy*

Pada gambar 4.6 tampilan *canvas input* nilai variable *DCA Fuzzy* merupakan hasil dari tindakan *user* setelah memilih *button input DCA Fuzzy* pada panel *button setting* sebelumnya pada gambar 4.5. Tampilan pada gambar 4.6 di atas digunakan untuk memasukkan nilai ke dalam tiap variabel masukan yang selanjutnya akan dilakukan untuk mengetahui hasil pengujian. Dari panel tersebut terdapat tiga variabel masukan yaitu, *Score*, *Time Remain* dan *Player Life*.

4.2.7 Tampilan *Canvas Game Win*

Tampilan *canvas game win* pada gambar 4.7 menandakan *score* target yang dicapai *player* tercapai dan memenangkan permainan. Pada tampilan tersebut berisikan dua tombol yaitu tombol *Retry* dan *Menu Utama*. Tombol *Retry* untuk mengulang permainan, sedangkan tombol *Menu Utama* untuk kembali ke *menu awal* atau *main menu*.



Gambar 4.7 Tampilan *canvas game win*

Pada gambar 4.7, tampilan *canvas game win* atau tampilan memenangkan permainan akan muncul ketika pemain berhasil menyelesaikan *game* dengan baik dan mencapai target yang telah ditentukan. Setelah memenangkan permainan, pemain dapat memilih bermain kembali atau kembali ke menu utama.

4.2.8 Tampilan Canvas Game Lose

Tampilan *canvas game lose* pada gambar 4.8 menandakan *score target* yang dicapai *player* tidak tercapai atau bisa jadi darah *player* habis dan *player* dinyatakan kalah dari permainan. Pada tampilan tersebut berisikan dua tombol yaitu tombol *Retry* dan *Menu Utama*. Tombol *Retry* untuk mengulang permainan, sedangkan tombol *Menu Utama* untuk kembali ke *menu awal* atau *main menu*.



Gambar 4.8 Tampilan *canvas lose game*

Pada gambar 4.8 merupakan tampilan ketika *player* tidak berhasil memenangkan permainan atau hasil selama bermain *game* tidak sesuai dengan target yang telah ditentukan, sehingga *player* harus memainkan ulang *game* tersebut untuk menyelesaikan permainan atau kembali ke menu utama.

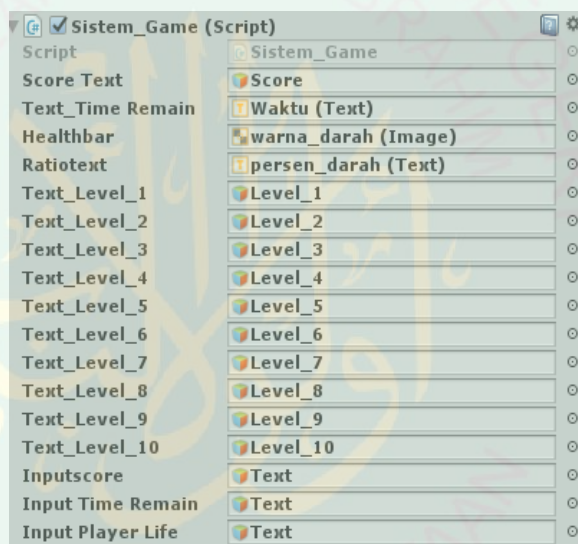
4.3 Tahap Pengujian

Tahap pengujian sistem level kabut pada *game adventure Gunung Kelud* yang menggunakan DCA dengan *Fuzzy Sugeno*. Hal yang diamati pada penelitian ini sesuai dengan tujuan penelitian yang telah dituliskan pada Bab 1 serta dengan perencanaan yang telah dibuat pada Bab 3. Pengujian dilakukan dengan memasukkan nilai tiap variabel pada sistem *game* dengan melalui tiga tahap yaitu, sistem tanpa DCA, sistem dengan DCA, dan sistem

yang menggunakan DCA Fuzzy. Peneliti akan melakukan beberapa tahap persiapan pengujian terlebih dahulu sebelum memasuki tahap pengujian sistem.

4.3.1 Pembuatan Sistem Level DCA

Sistem level DCA dibuat menggunakan *script* bahasa pemrograman C# yang diberi nama *Sistem_Game*. *Script* tersebut digunakan untuk sistem perubahan level dalam *game*. Konsep *script* sistem level *game* dapat dilihat pada gambar 4.9.



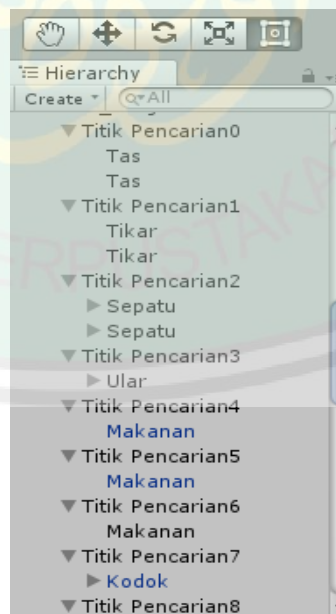
Gambar 4.9 Panel *Script* sistem level DCA pada *Unity*

Pada gambar 4.9 terdapat beberapa penamaan dari *script* yang telah dibuat diantaranya, *Score Text* yang menampilkan perolehan *score* yang didapat *player* melalui *point* tiap barang atau makanan, *Text_Time Remain* yang menampilkan *Time Remain* pada *game* atau sisa waktu yang ditempuh untuk menyelesaikan misi pada *game* dengan waktu awal 300 detik ditempuh sampai waktu habis, *Healthbar* dan *Ratiotext* yang menampilkan *Player Life*

atau sisa darah *player* selama bermain, kemudian *Text_Level_1* sampai *Text_Level_10* sebagai tampilan level tiap kabut yang berubah-ubah, dan untuk *Inputscore*, *Input Time Remain*, dan *Input Player Life* yang menggunakan *InputField* untuk memudahkan pengecekan kabut setiap perubahan level.

4.3.2 Objek Pendukung Perubahan Level Pada Sistem Game

Objek pendukung sistem level *game* sangat dibutuhkan untuk mempengaruhi perubahan level kabut secara *real time*. Objek yang dibutuhkan tersebut diantaranya objek barang atau makanan dan objek rintangan. Untuk objek barang atau makanan berpengaruh pada *score* yang didapatkan oleh *player* serta dapat menambah darah *player* sampai penuh selama bermain, sedangkan untuk objek rintangan berpengaruh pada pengurangan darah *player* jika menyentuh objek tersebut.



Gambar 4.10 Objek pendukung perubahan level

Seperti yang terlihat pada gambar 4.10 beberapa objek yang mempengaruhi perubahan level terdapat di beberapa titik pencarian yang harus dilewati oleh *player NPC*, dimana objek barang atau makanan berupa macam-macam barang dan macam-macam buah-buahan, sedangkan objek rintangan berupa hewan, dan kedua objek tersebut tersebar secara acak.

4.4 Hasil Pengujian

Pada pengujian sistem level kabut menggunakan *DCA Fuzzy* akan dibandingkan pada sistem level kabut tanpa *DCA Fuzzy* dengan nilai masukan variabel yang sama dalam beberapa percobaan sebagai berikut: Pada percobaan ini diproses perhitungan terlebih dahulu dengan dilakukan pembagian dan penjumlahan sehingga didapatkan total dari setiap masukan, kemudian dari hasil perhitungan tersebut dilakukan penentuan level menggunakan *rule fuzzy* yang dilakukan pengujian pada sistem.

1. Percobaan 1

Pada percobaan 1 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.1 Percobaan pengujian 1

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 0 | 297 | 82 | 1 | 1 | 1 |
| 2 | 0 | 286 | 80 | 1 | 1 | 1 |
| 3 | 1 | 278 | 72 | 1 | 1 | 1 |
| 4 | 2 | 257 | 67 | 1 | 1 | 2 |
| 5 | 5 | 218 | 63 | 2 | 1 | 1 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 6 | 10 | 196 | 62 | 3 | 2 | 1 |
| 7 | 15 | 190 | 62 | 4 | 3 | 1 |
| 8 | 15 | 190 | 58 | 4 | 3 | 1 |
| 9 | 17 | 162 | 55 | 4 | 4 | 6 |
| 10 | 23 | 138 | 55 | 5 | 5 | 6 |
| 11 | 31 | 129 | 49 | 7 | 6 | 6 |
| 12 | 33 | 114 | 45 | 7 | 6 | 6 |
| 13 | 37 | 112 | 43 | 8 | 6 | 9 |
| 14 | 38 | 104 | 43 | 8 | 6 | 9 |
| 15 | 41 | 95 | 23 | 9 | 8 | 10 |
| 16 | 47 | 90 | 21 | 10 | 8 | 10 |
| 17 | 58 | 82 | 14 | 10 | 10 | 10 |
| 18 | 59 | 68 | 7 | 10 | 10 | 10 |
| 19 | 59 | 6 | 2 | 10 | 10 | 10 |
| 20 | 60 | 5 | 2 | 10 | 10 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.1 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran, diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 8. Selanjutnya dilakukan perhitungan sebagai berikut:

- a. Perhitungan Akurasi

$$\text{Akurasi} = \frac{TP}{TP + TN + FP + FN}$$

$$\text{Akurasi} = 8 / 20$$

$$\text{Akurasi} = 0,4$$

- b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$FP(1) = 1 ; FP(2) = 1 ; FP(3) = 1 ; FP(4) = 3 ; FP(5) = 1 ; FP(6) = 0 ;$$

$$FP(7) = 2 ; FP(8) = 2 ; FP(9) = 1 ; FP(10) = 0$$

Setelah diketahui nilai FP masing- masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$Precision = TP / (TP + FP)$$

$$P(1) = 3/(3+1) = 0,75 ; P(2) = 0/(0+1) = 0 ; P(3) = 0/(0+1) = 0 ; P(4) = 0/(0+3) = 0 ; P(5) = 0/(0+1) = 0 ; P(6) = 0/(0+0) = 0 ; P(7) = 0/(0+2) = 0 ; P(8) = 0/(0+2) = 0 ; P(9) = 0/(0+1) = 0 ; P(10) = 5/(5+0) = 1$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$Precision = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$$

$$Precision = (0,75+0+0+0+0+0+0+0+0+1) / 10 = \mathbf{0,175}$$

c. Perhitungan *Recall*

$$FN(1) = 4 ; FN(2) = 1 ; FN(3) = 0 ; FN(4) = 0 ; FN(5) = 0 ; FN(6) = 2 ; FN(7) = 0 ; FN(8) = 0 ; FN(9) = 2 ; FN(10) = 1 ;$$

Setelah diketahui nilai FN masing- masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP / (TP + FN)$$

$$R(1) = 3/(3+4) = 0,428571 ; R(2) = 0/(0+1) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 0/(0+2) = 0 ; R(7) = 0/(0+0) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+2) = 0 ; R(10) = 5/(5+1) = 0,833333$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,428571+0+0+0+0+0+0+0+0+0,833333) / 10 = \mathbf{0,12619}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * \mathbf{0,175} * \mathbf{0,12619}) / (\mathbf{0,175} + \mathbf{0,12619}) = \mathbf{0,146640316}$$

2. Percobaan 2

Pada percobaan 2 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.2 Percobaan pengujian 2

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 0 | 275 | 97 | 1 | 1 | 1 |
| 2 | 6 | 238 | 92 | 2 | 1 | 3 |
| 3 | 8 | 227 | 91 | 2 | 1 | 3 |
| 4 | 12 | 211 | 89 | 3 | 2 | 3 |
| 5 | 15 | 205 | 88 | 4 | 2 | 7 |
| 6 | 18 | 187 | 86 | 4 | 2 | 7 |
| 7 | 19 | 161 | 86 | 4 | 2 | 7 |
| 8 | 23 | 157 | 82 | 5 | 2 | 7 |
| 9 | 23 | 154 | 72 | 5 | 3 | 7 |
| 10 | 24 | 152 | 69 | 5 | 4 | 6 |
| 11 | 36 | 152 | 68 | 8 | 4 | 9 |
| 12 | 39 | 142 | 64 | 8 | 4 | 9 |
| 13 | 39 | 114 | 63 | 8 | 4 | 9 |
| 14 | 44 | 98 | 63 | 9 | 4 | 9 |
| 15 | 45 | 92 | 60 | 10 | 5 | 9 |
| 16 | 46 | 90 | 59 | 10 | 5 | 9 |
| 17 | 48 | 88 | 58 | 10 | 5 | 9 |
| 18 | 53 | 66 | 57 | 10 | 5 | 9 |
| 19 | 54 | 62 | 49 | 10 | 6 | 9 |
| 20 | 55 | 34 | 15 | 10 | 8 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.2 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 4 data berjumlah 4. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$Akurasi = TP / TP + TN + FP + FN$$

$$Akurasi = 4 / 20$$

$$Akurasi = \mathbf{0,2}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$FP(1) = 0 ; FP(2) = 2 ; FP(3) = 0 ; FP(4) = 3 ; FP(5) = 3 ; FP(6) = 0 ; FP(7) = 0 ; FP(8) = 3 ; FP(9) = 0 ; FP(10) = 5$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$Precision = TP / (TP + FP)$$

$$P(1) = 1/(1+0) = 1 ; P(2) = 0/(0+2) = 0 ; P(3) = 1/(1+0) = 1 ; P(4) = 0/(0+3) = 0 ; P(5) = 0/(0+3) = 0 ; P(6) = 0/(0+0) = 0 ; P(7) = 0/(0+0) = 0 ; P(8) = 0/(0+3) = 0 ; P(9) = 1/(1+0) = 1 ; P(10) = 1/(1+5) = \mathbf{0,166666667}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$$

$$\text{Precision} = (1+0+1+0+0+0+0+0+1+0,166666667) / 10 = \mathbf{0,316666667}$$

c. Perhitungan *Recall*

$$\text{FN}(1) = 0 ; \text{FN}(2) = 0 ; \text{FN}(3) = 2 ; \text{FN}(4) = 0 ; \text{FN}(5) = 0 ; \text{FN}(6) = 1 ; \\ \text{FN}(7) = 5 ; \text{FN}(8) = 0 ; \text{FN}(9) = 8 ; \text{FN}(10) = 0 ;$$

Setelah diketahui nilai FN masing- masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\text{Recall} = TP / (TP + FN)$$

$$\text{R}(1) = 1 / (1+0) = 1 ; \text{R}(2) = 0 / (0+0) = 0 ; \text{R}(3) = 1 / (1+2) = 0,333333333 ; \\ \text{R}(4) = 0 / (0+0) = 0 ; \text{R}(5) = 0 / (0+0) = 0 ; \text{R}(6) = 0 / (0+1) = 0 ; \text{R}(7) = 0 / (0+5) \\ = 0 ; \text{R}(8) = 0 / (0+0) = 0 ; \text{R}(9) = 1 / (1+8) = 0,111111111 ; \text{R}(10) = 1 / (1+0) \\ = 1$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$\text{Recall} = (1+0+0,333333333+0+0+0+0+0+0,111111111+1) / 10 = \\ \mathbf{0,244444444}$$

d. Perhitungan *F-Measure*

$$\text{F-Measure} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

$$\text{F-Measure} = (2 * \mathbf{0,316666667} * \mathbf{0,244444444}) / (\mathbf{0,316666667} + \\ \mathbf{0,244444444}) = \mathbf{0,275907591}$$

3. Percobaan 3

Pada percobaan 3 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.3 Percobaan pengujian 3

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 2 | 292 | 96 | 1 | 1 | 1 |
| 2 | 8 | 290 | 90 | 2 | 1 | 1 |
| 3 | 8 | 287 | 89 | 2 | 1 | 1 |
| 4 | 8 | 237 | 83 | 2 | 2 | 3 |
| 5 | 9 | 217 | 83 | 2 | 2 | 3 |
| 6 | 9 | 187 | 81 | 2 | 2 | 1 |
| 7 | 10 | 177 | 71 | 3 | 2 | 1 |
| 8 | 14 | 174 | 55 | 3 | 3 | 1 |
| 9 | 16 | 149 | 54 | 4 | 4 | 6 |
| 10 | 23 | 130 | 54 | 5 | 5 | 6 |
| 11 | 26 | 122 | 52 | 6 | 5 | 6 |
| 12 | 28 | 117 | 49 | 6 | 6 | 6 |
| 13 | 28 | 107 | 48 | 6 | 6 | 6 |
| 14 | 28 | 74 | 47 | 6 | 6 | 6 |
| 15 | 29 | 59 | 45 | 6 | 6 | 6 |
| 16 | 30 | 49 | 40 | 7 | 6 | 6 |
| 17 | 38 | 47 | 39 | 8 | 7 | 9 |
| 18 | 41 | 41 | 26 | 9 | 8 | 10 |
| 19 | 42 | 39 | 14 | 9 | 9 | 10 |
| 20 | 44 | 23 | 7 | 9 | 9 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.3 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran,

diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 6. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 6 / 20$$

$$\text{Akurasi} = \mathbf{0,3}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP}(1) = 0 ; \text{FP}(2) = 5 ; \text{FP}(3) = 2 ; \text{FP}(4) = 1 ; \text{FP}(5) = 1 ; \text{FP}(6) = 0 ; \text{FP}(7) = 1 ; \text{FP}(8) = 1 ; \text{FP}(9) = 3 ; \text{FP}(10) = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$\begin{aligned} P(1) &= 1/(1+0) = 1 ; P(2) = 0/(0+5) = 0 ; P(3) = 0/(0+2) = 0 ; P(4) = 0/(0+1) \\ &= 0 ; P(5) = 0/(0+1) = 0 ; P(6) = 5/(5+0) = 1 ; P(7) = 0/(0+1) = 0 ; P(8) = \\ &0/(0+1) = 0 ; P(9) = 0/(0+3) = 0 ; P(10) = 0/(0+0) = 0 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (1+0+0+0+0+1+0+0+0+0) / 10 = \mathbf{0,2}$$

c. Perhitungan *Recall*

$$\begin{aligned} \text{FN}(1) &= 5 ; \text{FN}(2) = 0 ; \text{FN}(3) = 2 ; \text{FN}(4) = 0 ; \text{FN}(5) = 0 ; \text{FN}(6) = 3 ; \\ \text{FN}(7) &= 0 ; \text{FN}(8) = 0 ; \text{FN}(9) = 1 ; \text{FN}(10) = 3 ; \end{aligned}$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 1/(1+5) = 0,166666667 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+2) = 0 ;$$

$$R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 5/(5+3) = 0,625 ; R(7) =$$

$$0/(0+0) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+1) = 0 ; R(10) = 0/(0+3) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,166666667+0+0+0+0+0,625+0+0+0+0) / 10 = \mathbf{0,079166667}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * 0,2 * 0,079166667) / (0,2+ 0,079166667) = \mathbf{0,113432836}$$

4. Percobaan 4

Pada percobaan 4 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.4 Percobaan pengujian 4

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 0 | 298 | 100 | 1 | 1 | 1 |
| 2 | 8 | 281 | 98 | 2 | 1 | 1 |
| 3 | 10 | 271 | 96 | 3 | 1 | 1 |
| 4 | 18 | 254 | 95 | 4 | 1 | 4 |
| 5 | 22 | 241 | 91 | 5 | 1 | 4 |
| 6 | 26 | 239 | 90 | 6 | 2 | 7 |
| 7 | 27 | 235 | 82 | 6 | 2 | 7 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 8 | 32 | 232 | 79 | 7 | 3 | 7 |
| 9 | 34 | 183 | 72 | 7 | 3 | 7 |
| 10 | 34 | 156 | 71 | 7 | 3 | 7 |
| 11 | 35 | 143 | 67 | 8 | 4 | 9 |
| 12 | 36 | 140 | 61 | 8 | 4 | 9 |
| 13 | 38 | 130 | 59 | 8 | 5 | 9 |
| 14 | 42 | 109 | 42 | 9 | 6 | 9 |
| 15 | 45 | 105 | 30 | 10 | 8 | 10 |
| 16 | 48 | 101 | 26 | 10 | 8 | 10 |
| 17 | 49 | 87 | 17 | 10 | 9 | 10 |
| 18 | 51 | 86 | 14 | 10 | 10 | 10 |
| 19 | 52 | 78 | 8 | 10 | 10 | 10 |
| 20 | 60 | 24 | 7 | 10 | 10 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.4 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 5 data berjumlah 12. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 12 / 20$$

$$\text{Akurasi} = \mathbf{0,6}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP}(1) = 0 ; \text{FP}(2) = 1 ; \text{FP}(3) = 1 ; \text{FP}(4) = 0 ; \text{FP}(5) = 0 ; \text{FP}(6) = 2 ; \text{FP}(7) = 0 ; \text{FP}(8) = 3 ; \text{FP}(9) = 0 ; \text{FP}(10) = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\textit{Precision} = \textit{TP} / (\textit{TP} + \textit{FP})$$

$$\begin{aligned} P(1) &= 1/(1+0) = 1 ; P(2) = 0/(0+1) = 0 ; P(3) = 0/(0+1) = 0 ; P(4) = 1/(1+0) \\ &= 1 ; P(5) = 0/(0+0) = 0 ; P(6) = 0/(0+2) = 0 ; P(7) = 3/(3+0) = 1 ; P(8) = \\ &0/(0+3) = 0 ; P(9) = 1/(1+0) = 1 ; P(10) = 6/(6+0) = 1 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\textit{Precision} = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \textit{Jumlah kelas}$$

$$\textit{Precision} = (1+0+0+1+0+0+1+0+1+1) / 10 = \mathbf{0,5}$$

c. Perhitungan *Recall*

$$\begin{aligned} \textit{FN}(1) &= 2 ; \textit{FN}(2) = 0 ; \textit{FN}(3) = 0 ; \textit{FN}(4) = 1 ; \textit{FN}(5) = 0 ; \textit{FN}(6) = 0 ; \\ \textit{FN}(7) &= 2 ; \textit{FN}(8) = 0 ; \textit{FN}(9) = 3 ; \textit{FN}(10) = 0 ; \end{aligned}$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\textit{Recall} = \textit{TP}/(\textit{TP}+\textit{FN})$$

$$\begin{aligned} R(1) &= 1/(1+2) = 0,333333333 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+0) = 0 ; \\ R(4) &= 1/(1+1) = 0,5 ; R(5) = 0/(0+0) = 0 ; R(6) = 0/(0+0) = 0 ; R(7) = \\ 3/(3+2) &= 0,6 ; R(8) = 0/(0+0) = 0 ; R(9) = 1/(1+3) = 0,25 ; R(10) = 6/(6+0) \\ &= 1 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,333333333+0+0+0,5+0+0+0,6+0+0,25+1) / 10 = \mathbf{0,268333333}$$

d. Perhitungan *F-Measure*

$$F\text{-Measure} = (2 * Precision * Recall) / (Precision + Recall)$$

$$F\text{-Measure} = (2 * \mathbf{0,5} * \mathbf{0,268333333}) / (\mathbf{0,5} + \mathbf{0,268333333}) = \mathbf{0,349240781}$$

5. Percobaan 5

Pada percobaan 5 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.5 Percobaan pengujian 5

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 4 | 250 | 100 | 1 | 1 | 1 |
| 2 | 6 | 245 | 95 | 2 | 1 | 1 |
| 3 | 6 | 181 | 86 | 2 | 2 | 3 |
| 4 | 8 | 179 | 80 | 2 | 2 | 3 |
| 5 | 8 | 169 | 77 | 2 | 2 | 3 |
| 6 | 9 | 166 | 71 | 2 | 2 | 3 |
| 7 | 13 | 162 | 70 | 3 | 3 | 1 |
| 8 | 13 | 160 | 70 | 3 | 3 | 3 |
| 9 | 15 | 141 | 67 | 4 | 3 | 3 |
| 10 | 17 | 133 | 61 | 4 | 4 | 6 |
| 11 | 20 | 131 | 61 | 5 | 4 | 6 |
| 12 | 20 | 122 | 61 | 5 | 4 | 6 |
| 13 | 30 | 114 | 60 | 7 | 5 | 6 |
| 14 | 37 | 105 | 45 | 8 | 6 | 9 |
| 15 | 37 | 102 | 32 | 8 | 7 | 9 |
| 16 | 41 | 89 | 29 | 9 | 8 | 10 |
| 17 | 42 | 74 | 22 | 9 | 9 | 10 |
| 18 | 44 | 73 | 21 | 9 | 8 | 10 |
| 19 | 47 | 70 | 5 | 10 | 10 | 10 |
| 20 | 50 | 69 | 2 | 10 | 10 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.5 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 3 data berjumlah 4. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 4 / 20$$

$$\text{Akurasi} = \mathbf{0,2}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP}(1) = 0 ; \text{FP}(2) = 5 ; \text{FP}(3) = 1 ; \text{FP}(4) = 2 ; \text{FP}(5) = 2 ; \text{FP}(6) = 0 ; \text{FP}(7) = 1 ; \text{FP}(8) = 2 ; \text{FP}(9) = 3 ; \text{FP}(10) = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$\begin{aligned} P(1) &= 1/(1+0) = 1 ; P(2) = 0/(0+5) = 0 ; P(3) = 1/(1+1) = 0,5 ; P(4) = 0/(0+2) \\ &= 0 ; P(5) = 0/(0+2) = 0 ; P(6) = 0/(0+0) = 0 ; P(7) = 0/(0+1) = 0 ; P(8) = \\ &0/(0+2) = 0 ; P(9) = 0/(0+3) = 0 ; P(10) = 2/(2+0) = 1 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$Precision = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$

$$Precision = (1+0+0,5+0+0+0+0+0+0+1) / 10 = \mathbf{0,25}$$

c. Perhitungan *Recall*

FN (1) = 2 ; FN(2) = 0 ; FN (3) = 5 ; FN (4) = 0 ; FN (5) = 0 ; FN (6) = 4;
FN (7) = 0 ; FN (8) = 0 ; FN (9) = 2 ; FN (10) = 3;

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 1/(1+2) = 0,333333333; R(2) = 0/(0+0) = 0 ; R(3) = 1/(1+5) = 0,166666667 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 0/(0+4) = 0 ; R(7) = 0/(0+0) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+2) = 0 ; R(10) = 2/(2+3) = 0,4$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,333333333+0+0,166666667+0+0+0+0+0+0+0,4) / 10 = \mathbf{0,132352941}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * \mathbf{0,25} * \mathbf{0,09}) / (\mathbf{0,25} + \mathbf{0,09}) = \mathbf{0,132352941}$$

6. Percobaan 6

Pada percobaan 6 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan

penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.6 Percobaan pengujian 6

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 0 | 296 | 88 | 1 | 1 | 1 |
| 2 | 2 | 292 | 84 | 1 | 1 | 1 |
| 3 | 6 | 275 | 74 | 2 | 2 | 1 |
| 4 | 15 | 270 | 74 | 4 | 2 | 4 |
| 5 | 13 | 265 | 73 | 3 | 2 | 1 |
| 6 | 24 | 246 | 68 | 5 | 3 | 5 |
| 7 | 25 | 241 | 60 | 6 | 3 | 5 |
| 8 | 31 | 240 | 56 | 7 | 4 | 6 |
| 9 | 31 | 240 | 55 | 7 | 4 | 6 |
| 10 | 33 | 240 | 37 | 7 | 4 | 6 |
| 11 | 35 | 240 | 37 | 8 | 4 | 7 |
| 12 | 39 | 188 | 28 | 8 | 6 | 8 |
| 13 | 40 | 178 | 22 | 9 | 7 | 8 |
| 14 | 46 | 170 | 22 | 10 | 7 | 8 |
| 15 | 48 | 170 | 20 | 10 | 7 | 8 |
| 16 | 48 | 168 | 16 | 10 | 7 | 8 |
| 17 | 52 | 168 | 14 | 10 | 7 | 8 |
| 18 | 54 | 147 | 13 | 10 | 8 | 10 |
| 19 | 56 | 110 | 12 | 10 | 10 | 10 |
| 20 | 56 | 100 | 4 | 10 | 10 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.6 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 5 data berjumlah 8. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 8 / 20$$

Akurasi = 0,4

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

FP(1) = 0 ; FP(2) = 1 ; FP(3) = 1 ; FP (4) = 0 ; FP(5) = 0 ; FP(6) = 1 ; FP(7) = 3 ; FP(8) = 1 ; FP(9) = 1 ; FP(10) = 4

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$Precision = TP / (TP + FP)$$

P(1) = 2/(2+0) = 1 ; P(2) = 0/(0+1) = 0 ; P(3) = 0/(0+1) = 0 ; P(4) = 1/(1+0) = 1 ; P(5) = 1/(1+0) = 1 ; P(6) = 0/(0+1) = 0 ; P(7) = 0/(0+3) = 0 ; P(8) = 1/(1+1) = 0,5 ; P(9) = 0/(0+1) = 0 ; P(10) = 3/(3+4) = 0,428571429

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$Precision = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$$

$$Precision = (1+0+0+1+1+0+0+0,5+0+0,428571429) / 10 = \mathbf{0,392857143}$$

c. Perhitungan *Recall*

FN(1) = 2 ; FN(2) = 0 ; FN (3) = 0 ; FN (4) = 0 ; FN (5) = 1 ; FN (6) = 3 ; FN (7) = 1 ; FN (8) = 5 ; FN (9) = 0 ; FN (10) = 0;

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 2/(2+2) = 0,5 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 1/(1+0) = 1 ; R(5) = 1/(1+1) = 0,5 ; R(6) = 0/(0+3) = 0 ; R(7) = 0/(0+1) = 0 ; R(8) = 1/(1+5) = 0,166666667 ; R(9) = 0/(0+0) = 0 ; R(10) = 3/(3+0) = 1$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,5+0+0+1+0,5+0+0+0,166666667+0+1) / 10 = \mathbf{0,316666667}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * \mathbf{0,392857143} * \mathbf{0,316666667}) / (\mathbf{0,392857143} + \mathbf{0,316666667}) = \mathbf{0,350671141}$$

7. Percobaan 7

Pada percobaan 7 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.7 Percobaan pengujian 7

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 3 | 243 | 96 | 1 | 1 | 1 |
| 2 | 7 | 235 | 90 | 2 | 2 | 3 |
| 3 | 12 | 230 | 89 | 3 | 2 | 3 |
| 4 | 17 | 217 | 83 | 4 | 2 | 7 |
| 5 | 18 | 187 | 83 | 4 | 2 | 7 |
| 6 | 19 | 178 | 81 | 4 | 2 | 7 |
| 7 | 19 | 174 | 71 | 4 | 3 | 7 |
| 8 | 20 | 156 | 55 | 5 | 4 | 6 |
| 9 | 21 | 143 | 54 | 5 | 5 | 6 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 10 | 23 | 134 | 54 | 5 | 5 | 6 |
| 11 | 24 | 131 | 52 | 5 | 5 | 6 |
| 12 | 24 | 115 | 49 | 5 | 5 | 6 |
| 13 | 27 | 89 | 48 | 6 | 6 | 6 |
| 14 | 30 | 82 | 47 | 7 | 6 | 6 |
| 15 | 32 | 82 | 45 | 7 | 6 | 6 |
| 16 | 32 | 82 | 40 | 7 | 7 | 6 |
| 17 | 36 | 63 | 39 | 8 | 7 | 9 |
| 18 | 40 | 55 | 26 | 9 | 8 | 10 |
| 19 | 42 | 34 | 14 | 9 | 9 | 10 |
| 20 | 48 | 26 | 7 | 10 | 10 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.7 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 4 data berjumlah 4. Selanjutnya dilakukan perhitungan sebagai berikut:

- a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 4 / 20$$

$$\text{Akurasi} = \mathbf{0,2}$$

- b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP}(1) = 0 ; \text{FP}(2) = 1 ; \text{FP}(3) = 0 ; \text{FP}(4) = 4 ; \text{FP}(5) = 5 ; \text{FP}(6) = 0 ; \text{FP}(7) = 3 ; \text{FP}(8) = 1 ; \text{FP}(9) = 2 ; \text{FP}(10) = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\textit{Precision} = TP / (TP + FP)$$

$$P(1) = 1/(1+0) = 1 ; P(2) = 0/(0+1) = 0 ; P(3) = 1/(1+0) = 1 ; P(4) = 0/(0+4) = 0 ; P(5) = 0/(0+5) = 0 ; P(6) = 1/(1+0) = 1 ; P(7) = 0/(0+3) = 0 ; P(8) = 0/(0+1) = 0 ; P(9) = 0/(0+2) = 0 ; P(10) = 1/(1+0) = 1$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\textit{Precision} = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$$

$$\textit{Precision} = (1+0+1+0+0+1+0+0+0+1) / 10 = \mathbf{0,4}$$

c. Perhitungan *Recall*

$$FN(1) = 0 ; FN(2) = 0 ; FN(3) = 1 ; FN(4) = 0 ; FN(5) = 0 ; FN(6) = 8 ; FN(7) = 4 ; FN(8) = 0 ; FN(9) = 1 ; FN(10) = 2 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\textit{Recall} = TP/(TP+FN)$$

$$R(1) = 1/(1+0) = 1 ; R(2) = 0/(0+0) = 0 ; R(3) = 1/(1+1) = 0,5 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 1/(1+8) = 0,111111111 ; R(7) = 0/(0+4) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+1) = 0 ; R(10) = 1/(1+2) = 0,333333333$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$\textit{Recall} = (1+0+0+0+0+0,111111111+0+0+0+0,333333333) / 10 = \mathbf{0,144444444}$$

d. Perhitungan *F-Measure*

$$F\text{-Measure} = (2 * \textit{Precision} * \textit{Recall}) / (\textit{Precision} + \textit{Recall})$$

$$F\text{-Measure} = (2 * 0,4 * 0,1444444444) / (0,4 + 0,1444444444) = 0,212244898$$

8. Percobaan 8

Pada percobaan 8 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.8 Percobaan pengujian 8

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 1 | 198 | 99 | 1 | 1 | 3 |
| 2 | 5 | 194 | 84 | 2 | 1 | 3 |
| 3 | 10 | 193 | 82 | 3 | 2 | 3 |
| 4 | 15 | 160 | 70 | 4 | 3 | 3 |
| 5 | 19 | 150 | 70 | 4 | 4 | 6 |
| 6 | 20 | 136 | 63 | 5 | 4 | 6 |
| 7 | 23 | 124 | 60 | 5 | 5 | 6 |
| 8 | 23 | 121 | 57 | 5 | 5 | 6 |
| 9 | 26 | 109 | 54 | 6 | 5 | 6 |
| 10 | 29 | 96 | 49 | 6 | 6 | 6 |
| 11 | 30 | 92 | 47 | 7 | 6 | 6 |
| 12 | 33 | 90 | 46 | 7 | 6 | 6 |
| 13 | 34 | 83 | 28 | 7 | 7 | 4 |
| 14 | 34 | 77 | 24 | 7 | 7 | 4 |
| 15 | 36 | 76 | 23 | 8 | 8 | 10 |
| 16 | 36 | 52 | 22 | 8 | 8 | 10 |
| 17 | 38 | 50 | 19 | 8 | 8 | 10 |
| 18 | 39 | 32 | 18 | 8 | 8 | 10 |
| 19 | 40 | 27 | 17 | 9 | 8 | 10 |
| 20 | 42 | 10 | 15 | 9 | 9 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.23 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 3. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 3 / 20$$

$$\text{Akurasi} = \mathbf{0,15}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP}(1) = 1 ; \text{FP}(2) = 1 ; \text{FP}(3) = 0 ; \text{FP}(4) = 2 ; \text{FP}(5) = 3 ; \text{FP}(6) = 0 ; \text{FP}(7) = 4 ; \text{FP}(8) = 4 ; \text{FP}(9) = 2 ; \text{FP}(10) = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$\begin{aligned} P(1) &= 0/(0+1) = 0 ; P(2) = 0/(0+1) = 0 ; P(3) = 1/(1+0) = 1 ; P(4) = 0/(0+2) \\ &= 0 ; P(5) = 0/(0+3) = 0 ; P(6) = 2/(2+0) = 1 ; P(7) = 0/(0+4) = 0 ; P(8) = \\ &0/(0+4) = 0 ; P(9) = 0/(0+2) = 0 ; P(10) = 0/(0+0) = 0 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$Precision = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$

$Precision = (0+0+1+0+0+1+0+0+0+0) / 10 = \mathbf{0,2}$

c. Perhitungan *Recall*

FN(1) = 0 ; FN(2) = 0 ; FN (3) = 3 ; FN (4) = 2 ; FN (5) = 0 ; FN (6) = 6;
FN (7) = 0 ; FN (8) = 0 ; FN (9) = 0 ; FN (10) = 6;

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$Recall = TP/(TP+FN)$

$R(1) = 0/(0+0) = 0$; $R(2) = 0/(0+0) = 0$; $R(3) = 1/(1+3) = 0,25$; $R(4) = 0/(0+2) = 0$; $R(5) = 0/(0+0) = 0$; $R(6) = 2/(2+6) = 0,25$; $R(7) = 0/(0+0) = 0$; $R(8) = 0/(0+0) = 0$; $R(9) = 0/(0+0) = 0$; $R(10) = 0/(0+6) = 0$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$Recall = (0+0+0,25+0+0+0,25+0+0+0+0) / 10 = \mathbf{0,05}$

d. Perhitungan *F-Measure*

$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$

$F-Measure = (2 * \mathbf{0,2} * \mathbf{0,05}) / (\mathbf{0,2} + \mathbf{0,05}) = \mathbf{0,08}$

9. Percobaan 9

Pada percobaan 9 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang

terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.9 Percobaan pengujian 9

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 1 | 293 | 100 | 1 | 1 | 1 |
| 2 | 1 | 291 | 94 | 1 | 1 | 1 |
| 3 | 4 | 269 | 88 | 1 | 1 | 1 |
| 4 | 6 | 267 | 87 | 2 | 2 | 1 |
| 5 | 7 | 260 | 84 | 2 | 2 | 1 |
| 6 | 7 | 259 | 82 | 2 | 2 | 1 |
| 7 | 8 | 258 | 79 | 2 | 2 | 1 |
| 8 | 8 | 256 | 78 | 2 | 2 | 1 |
| 9 | 8 | 254 | 70 | 2 | 2 | 2 |
| 10 | 9 | 250 | 69 | 2 | 2 | 2 |
| 11 | 11 | 250 | 63 | 3 | 3 | 2 |
| 12 | 13 | 248 | 59 | 3 | 3 | 2 |
| 13 | 13 | 242 | 59 | 3 | 3 | 2 |
| 14 | 14 | 238 | 58 | 3 | 3 | 1 |
| 15 | 15 | 234 | 57 | 4 | 3 | 1 |
| 16 | 21 | 221 | 57 | 5 | 4 | 6 |
| 17 | 24 | 216 | 48 | 5 | 5 | 6 |
| 18 | 30 | 156 | 36 | 7 | 6 | 6 |
| 19 | 36 | 100 | 26 | 8 | 8 | 10 |
| 20 | 40 | 76 | 22 | 9 | 8 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.9 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 5. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 5 / 20$$

Akurasi = **0,25**

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

FP(1) = 0 ; FP(2) = 5 ; FP(3) = 4 ; FP (4) = 1 ; FP(5) = 2 ; FP(6) = 0 ; FP(7) = 1 ; FP(8) = 1 ; FP(9) = 1 ; FP(10) = 0

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$Precision = TP / (TP + FP)$

P(1) = 3/(3+0) = 1 ; P(2) = 2/(2+5) = 0,285714286 ; P(3) = 0/(0+4) = 0 ;

P(4) = 0/(0+1) = 0 ; P(5) = 0/(0+2) = 0 ; P(6) = 0/(0+0) = 0 ; P(7) = 0/(0+1) = 0 ; P(8) = 0/(0+1) = 0 ; P(9) = 0/(0+1) = 0 ; P(10) = 0/(0+0) = 0

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$Precision = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$

$Precision = (1+0,285714286+0+0+0+0+0+0+0+0) / 10 = \mathbf{0,128571429}$

c. Perhitungan *Recall*

FN(1) = 7 ; FN(2) = 3 ; FN (3) = 0 ; FN (4) = 0 ; FN (5) = 0 ; FN (6) = 3;

FN (7) = 0 ; FN (8) = 0 ; FN (9) = 0 ; FN (10) = 2

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$Recall = TP/(TP+FN)$

$$R(1) = 3/(3+7) = 0,3 ; R(2) = 2/(2+3) = 0,4 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 0/(0+3) = 0 ; R(7) = 0/(0+0) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+0) = 0 ; R(10) = 0/(0+2) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,3+0,4+0+0+0+0+0+0+0+0) / 10 = 0,07$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * 0,128571429 * 0,07) / (0,128571429 + 0,07) = 0,090647482$$

10. Percobaan 10

Pada percobaan 10 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.10 Percobaan pengujian 10

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 1 | 278 | 84 | 1 | 1 | 1 |
| 2 | 4 | 277 | 84 | 1 | 1 | 1 |
| 3 | 6 | 269 | 83 | 2 | 2 | 1 |
| 4 | 7 | 260 | 74 | 2 | 2 | 1 |
| 5 | 8 | 253 | 67 | 2 | 2 | 2 |
| 6 | 9 | 248 | 65 | 2 | 2 | 2 |
| 7 | 9 | 237 | 60 | 2 | 2 | 1 |
| 8 | 14 | 233 | 53 | 3 | 3 | 1 |
| 9 | 19 | 230 | 49 | 4 | 4 | 6 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 10 | 19 | 222 | 46 | 4 | 4 | 6 |
| 11 | 23 | 220 | 46 | 5 | 5 | 6 |
| 12 | 26 | 216 | 44 | 6 | 5 | 6 |
| 13 | 26 | 212 | 44 | 6 | 5 | 6 |
| 14 | 29 | 200 | 41 | 6 | 6 | 6 |
| 15 | 29 | 188 | 39 | 6 | 6 | 6 |
| 16 | 31 | 177 | 37 | 7 | 7 | 6 |
| 17 | 36 | 175 | 36 | 8 | 7 | 7 |
| 18 | 38 | 173 | 32 | 8 | 7 | 7 |
| 19 | 39 | 171 | 29 | 8 | 7 | 8 |
| 20 | 40 | 120 | 20 | 9 | 8 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.10 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 4 data berjumlah 6. Selanjutnya dilakukan perhitungan sebagai berikut:

- a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 9 / 20$$

$$\text{Akurasi} = \mathbf{0,45}$$

- b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP (1)} = 0 ; \text{FP (2)} = 3 ; \text{FP(3)} = 1 ; \text{FP(4)} = 2 ; \text{FP(5)} = 1 ; \text{FP(6)} = 0 ; \text{FP(7)} = 1 ; \text{FP(8)} = 2 ; \text{FP(9)} = 1 ; \text{FP(10)} = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\textit{Precision} = TP / (TP + FP)$$

$$P(1) = 2/(2+0) = 1 ; P(2) = 2/(2+3) = 0,4 ; P(3) = 0/(0+1) = 0 ; P(4) = 0/(0+2) = 0 ; P(5) = 0/(0+1) = 0 ; P(6) = 4/(4+0) = 1 ; P(7) = 0/(0+1) = 0 ; P(8) = 1/(1+2) = 0,333333333 ; P(9) = 0/(0+1) = 0 ; P(10) = 0/(0+0) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\textit{Precision} = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$$

$$\textit{Precision} = (1+0,4+0+0+0+1+0+0,333333333+0+0)/10 = \mathbf{0,273333333}$$

c. Perhitungan *Recall*

$$FN(1) = 4 ; FN(2) = 0 ; FN(3) = 0 ; FN(4) = 0 ; FN(5) = 0 ; FN(6) = 4 ; FN(7) = 2 ; FN(8) = 0 ; FN(9) = 0 ; FN(10) = 1 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\textit{Recall} = TP/(TP+FN)$$

$$R(1) = 2/(2+4) = 0,333333333 ; R(2) = 2/(2+0) = 1 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 4/(4+4) = 0,5 ; R(7) = 0/(0+2) = 0 ; R(8) = 1/(1+0) = 1 ; R(9) = 0/(0+0) = 0 ; R(10) = 0/(0+1) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$\textit{Recall} = (0,333333333+1+0+0+0+0,5+0+1+0+0) / 10 = \mathbf{0,283333333}$$

d. Perhitungan *F-Measure*

$$\textit{F-Measure} = (2 * \textit{Precision} * \textit{Recall}) / (\textit{Precision} + \textit{Recall})$$

$$F\text{-Measure} = (2 * 0,273333333 * 0,283333333) / (0,273333333 + 0,283333333)$$

$$= 0,278243513$$

11. Percobaan 11

Pada percobaan 11 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.11 Percobaan pengujian 11

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 9 | 269 | 87 | 2 | 2 | 1 |
| 2 | 14 | 256 | 84 | 3 | 2 | 1 |
| 3 | 15 | 248 | 81 | 4 | 2 | 4 |
| 4 | 16 | 229 | 81 | 4 | 2 | 7 |
| 5 | 16 | 221 | 80 | 4 | 3 | 7 |
| 6 | 20 | 219 | 79 | 5 | 3 | 7 |
| 7 | 24 | 184 | 74 | 5 | 3 | 7 |
| 8 | 24 | 173 | 70 | 5 | 4 | 6 |
| 9 | 26 | 150 | 63 | 6 | 4 | 6 |
| 10 | 28 | 147 | 62 | 6 | 4 | 6 |
| 11 | 34 | 141 | 59 | 7 | 5 | 6 |
| 12 | 37 | 140 | 55 | 8 | 5 | 9 |
| 13 | 37 | 136 | 52 | 8 | 5 | 9 |
| 14 | 39 | 131 | 49 | 8 | 6 | 9 |
| 15 | 46 | 127 | 44 | 10 | 6 | 9 |
| 16 | 46 | 124 | 43 | 10 | 6 | 9 |
| 17 | 46 | 123 | 37 | 10 | 7 | 9 |
| 18 | 47 | 116 | 32 | 10 | 7 | 9 |
| 19 | 48 | 115 | 30 | 10 | 8 | 10 |
| 20 | 48 | 114 | 29 | 10 | 8 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.11 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data

dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 3 data berjumlah 5. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / (TP + TN + FP + FN)$$

$$\text{Akurasi} = 5 / 20$$

$$\text{Akurasi} = \mathbf{0,25}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP (1)} = 0 ; \text{FP (2)} = 1 ; \text{FP (3)} = 1 ; \text{FP (4)} = 2 ; \text{FP(5)} = 3 ; \text{FP(6)} = 0 ; \\ \text{FP(7)} = 1 ; \text{FP(8)} = 3 ; \text{FP(9)} = 0 ; \text{FP(10)} = 4$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$P(1) = 0/(0+0) = 0 ; P(2) = 0/(0+1) = 0 ; P(3) = 0/(0+1) = 0 ; P(4) = 1/(1+2) \\ = 0,333333333 ; P(5) = 0/(0+3) = 0 ; P(6) = 2/(2+0) = 1 ; P(7) = 0/(0+1) = \\ 0 ; P(8) = 0/(0+3) = 0 ; P(9) = 0/(0+0) = 0 ; P(10) = 2/(2+4) = 0,333333333$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + \\ P(10)) / \text{Jumlah kelas}$$

$$\text{Precision} = (0+0+0+0,333333333+0+1+0+0+0+0,333333333) / 10 = \mathbf{0,166666667}$$

c. Perhitungan *Recall*

$$\text{FN}(1) = 2 ; \text{FN}(2) = 0 ; \text{FN}(3) = 0 ; \text{FN}(4) = 0 ; \text{FN}(5) = 0 ; \text{FN}(6) = 2 ; \\ \text{FN}(7) = 4 ; \text{FN}(8) = 0 ; \text{FN}(9) = 7 ; \text{FN}(10) = 0 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{R}(1) = 0 / (0+2) = 0 ; \text{R}(2) = 0 / (0+0) = 0 ; \text{R}(3) = 0 / (0+0) = 0 ; \text{R}(4) = 0 / (0+0) = 0 ; \text{R}(5) = 0 / (0+0) = 0 ; \text{R}(6) = 2 / (2+2) = 0,5 ; \text{R}(7) = 0 / (0+4) = 0 ; \text{R}(8) = 0 / (0+0) = 0 ; \text{R}(9) = 0 / (0+7) = 0 ; \text{R}(10) = 2 / (2+0) = 1$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$\text{Recall} = (0+0+0+1+0+0,5+0+0+0+1) / 10 = \mathbf{0,25}$$

d. Perhitungan *F-Measure*

$$\text{F-Measure} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

$$\text{F-Measure} = (2 * \mathbf{0,166666667} * \mathbf{0,25}) / (\mathbf{0,166666667} + \mathbf{0,25}) = \mathbf{0,2}$$

12. Percobaan 12

Pada percobaan 12 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.12 Percobaan pengujian 12

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 2 | 290 | 71 | 1 | 1 | 1 |
| 2 | 5 | 285 | 70 | 2 | 1 | 1 |
| 3 | 5 | 283 | 65 | 2 | 1 | 1 |
| 4 | 6 | 279 | 62 | 2 | 2 | 2 |
| 5 | 6 | 276 | 61 | 2 | 2 | 2 |
| 6 | 9 | 245 | 59 | 2 | 2 | 2 |
| 7 | 9 | 239 | 52 | 2 | 2 | 1 |
| 8 | 10 | 236 | 52 | 3 | 2 | 1 |
| 9 | 11 | 230 | 47 | 3 | 3 | 1 |
| 10 | 15 | 222 | 46 | 4 | 3 | 1 |
| 11 | 16 | 213 | 43 | 4 | 4 | 1 |
| 12 | 17 | 210 | 40 | 4 | 4 | 6 |
| 13 | 19 | 206 | 38 | 4 | 4 | 6 |
| 14 | 19 | 200 | 35 | 4 | 4 | 6 |
| 15 | 20 | 198 | 34 | 5 | 4 | 6 |
| 16 | 21 | 194 | 33 | 5 | 5 | 6 |
| 17 | 22 | 191 | 33 | 5 | 5 | 6 |
| 18 | 22 | 188 | 33 | 5 | 5 | 6 |
| 19 | 38 | 187 | 32 | 8 | 6 | 7 |
| 20 | 44 | 158 | 20 | 9 | 8 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.12 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 4. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 4 / 20$$

$$\text{Akurasi} = \mathbf{0,2}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP (1)} = 0 ; \text{FP (2)} = 3 ; \text{FP (3)} = 2 ; \text{FP (4)} = 5 ; \text{FP(5)} = 4 ; \text{FP(6)} = 0 ; \\ \text{FP(7)} = 0 ; \text{FP(8)} = 1 ; \text{FP(9)} = 1 ; \text{FP(10)} = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{P(1)} = 1/(1+0) = 1 ; \text{P(2)} = 3/(3+3) = 0,5 ; \text{P(3)} = 0/(0+2) = 0 ; \text{P(4)} = 0/(0+5) \\ = 0 ; \text{P(5)} = 0/(0+4) = 0 ; \text{P(6)} = 0/(0+0) = 0 ; \text{P(7)} = 0/(0+0) = 0 ; \text{P(8)} = \\ 0/(0+1) = 0 ; \text{P(9)} = 0/(0+1) = 0 ; \text{P(10)} = 0/(0+0) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (\text{P(1)} + \text{P(2)} + \text{P(3)} + \text{P(4)} + \text{P(5)} + \text{P(6)} + \text{P(7)} + \text{P(8)} + \text{P(9)} + \\ \text{P(10)}) / \text{Jumlah kelas}$$

$$\text{Precision} = (1+0,5+0+0+0+0+0+0+0+0) / 10 = \mathbf{0,15}$$

c. Perhitungan *Recall*

$$\text{FN(1)} = 7 ; \text{FN(2)} = 0 ; \text{FN (3)} = 0 ; \text{FN (4)} = 0 ; \text{FN (5)} = 0 ; \text{FN (6)} = 7 ; \\ \text{FN (7)} = 1 ; \text{FN (8)} = 0 ; \text{FN (9)} = 0 ; \text{FN (10)} = 1 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$$

$$R(1) = 1/(1+7) = 0,125 ; R(2) = 3/(3+0) = 1 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 0/(0+7) = 0 ; R(7) = 0/(0+1) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+0) = 0 ; R(10) = 0/(0+1) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,125+1+0+0+0+0+0+0+0+0) / 10 = 0,1125$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * 0,15 * 0,1125) / (0,15 + 0,1125) = 0,128571429$$

13. Percobaan 13

Pada percobaan 13 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.13 Percobaan pengujian 13

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 0 | 260 | 40 | 1 | 1 | 1 |
| 2 | 1 | 253 | 45 | 1 | 1 | 1 |
| 3 | 2 | 244 | 48 | 1 | 1 | 1 |
| 4 | 4 | 231 | 49 | 1 | 1 | 1 |
| 5 | 4 | 224 | 51 | 1 | 1 | 1 |
| 6 | 5 | 218 | 53 | 2 | 1 | 1 |
| 7 | 9 | 211 | 59 | 2 | 2 | 1 |
| 8 | 10 | 205 | 60 | 3 | 2 | 1 |
| 9 | 11 | 200 | 62 | 3 | 3 | 1 |
| 10 | 12 | 198 | 63 | 3 | 3 | 1 |
| 11 | 13 | 197 | 64 | 3 | 3 | 1 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 12 | 15 | 195 | 64 | 4 | 3 | 1 |
| 13 | 19 | 193 | 67 | 4 | 4 | 6 |
| 14 | 20 | 193 | 68 | 5 | 4 | 6 |
| 15 | 21 | 193 | 71 | 5 | 3 | 7 |
| 16 | 24 | 188 | 72 | 5 | 3 | 7 |
| 17 | 25 | 186 | 73 | 6 | 3 | 7 |
| 18 | 30 | 186 | 75 | 7 | 3 | 7 |
| 19 | 36 | 185 | 75 | 8 | 3 | 9 |
| 20 | 45 | 155 | 75 | 10 | 3 | 8 |

Dari hasil level yang telah diperoleh pada tabel 4.13 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 6. Selanjutnya dilakukan perhitungan sebagai berikut:

- a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 6 / 20$$

$$\text{Akurasi} = 0,3$$

- b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$FP(1) = 0 ; FP(2) = 2 ; FP(3) = 4 ; FP(4) = 2 ; FP(5) = 3 ; FP(6) = 1 ;$$

$$FP(7) = 0 ; FP(8) = 1 ; FP(9) = 0 ; FP(10) = 1$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$P(1) = 5/(5+0) = 1 ; P(2) = 0/(0+2) = 0 ; P(3) = 0/(0+4) = 0 ; P(4) = 0/(0+2) = 0 ; P(5) = 0/(0+3) = 0 ; P(6) = 0/(0+1) = 0 ; P(7) = 1/(1+0) = 1 ; P(8) = 0/(0+1) = 0 ; P(9) = 0/(0+0) = 0 ; P(10) = 0/(0+1) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$Precision = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$$

$$Precision = (1+0+0+0+0+0+1+0+0+0) / 10 = \mathbf{0,2}$$

c. Perhitungan *Recall*

$$FN(1) = 7 ; FN(2) = 0 ; FN(3) = 0 ; FN(4) = 0 ; FN(5) = 0 ; FN(6) = 2 ; FN(7) = 3 ; FN(8) = 1 ; FN(9) = 1 ; FN(10) = 0 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 5/(5+7) = 0,416666667 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 0/(0+2) = 0 ; R(7) = 1/(1+3) = 0,25 ; R(8) = 0/(0+1) = 0 ; R(9) = 0/(0+1) = 0 ; R(10) = 0/(0+0) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,416666667+0+0+0+0+0+0,25+0+0+0) / 10 = \mathbf{0,066666667}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * \mathbf{0,2} * \mathbf{0,066666667}) / (\mathbf{0,2} + \mathbf{0,066666667}) = \mathbf{0,1}$$

14. Percobaan 14

Pada percobaan 14 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.14 Percobaan pengujian 14

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 6 | 5 | 12 | 2 | 1 | 1 |
| 2 | 11 | 8 | 16 | 3 | 2 | 1 |
| 3 | 18 | 15 | 17 | 4 | 4 | 4 |
| 4 | 21 | 20 | 17 | 5 | 4 | 4 |
| 5 | 28 | 23 | 17 | 6 | 6 | 4 |
| 6 | 28 | 37 | 19 | 6 | 6 | 4 |
| 7 | 32 | 56 | 24 | 7 | 7 | 4 |
| 8 | 34 | 63 | 29 | 7 | 7 | 4 |
| 9 | 34 | 91 | 30 | 7 | 7 | 4 |
| 10 | 35 | 116 | 34 | 8 | 7 | 6 |
| 11 | 38 | 125 | 35 | 8 | 7 | 9 |
| 12 | 39 | 134 | 44 | 8 | 6 | 9 |
| 13 | 39 | 148 | 45 | 8 | 6 | 9 |
| 14 | 40 | 152 | 46 | 9 | 6 | 9 |
| 15 | 40 | 152 | 48 | 9 | 6 | 9 |
| 16 | 42 | 167 | 51 | 9 | 5 | 7 |
| 17 | 47 | 169 | 53 | 10 | 5 | 7 |
| 18 | 48 | 177 | 60 | 10 | 5 | 7 |
| 19 | 55 | 177 | 62 | 10 | 4 | 7 |
| 20 | 59 | 180 | 68 | 10 | 4 | 7 |

Dari hasil level yang telah diperoleh pada tabel 4.14 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran

Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 3. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 6 / 20$$

$$\text{Akurasi} = \mathbf{0,15}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$FP(1) = 0 ; FP(2) = 1 ; FP(3) = 1 ; FP(4) = 0 ; FP(5) = 1 ; FP(6) = 2 ;$$

$$FP(7) = 3 ; FP(8) = 4 ; FP(9) = 1 ; FP(10) = 4$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$P(1) = 0/(0+0) = 0 ; P(2) = 0/(0+1) = 0 ; P(3) = 0/(0+1) = 0 ; P(4) = 1/(1+0)$$

$$= 1 ; P(5) = 0/(0+1) = 0 ; P(6) = 5/(5+0) = 1 ; P(7) = 0/(0+3) = 0 ; P(8) =$$

$$0/(0+4) = 0 ; P(9) = 2/(2+1) = 0,666666667 ; P(10) = 0/(0+4) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) +$$

$$P(10)) / \text{Jumlah kelas}$$

$$\text{Precision} = (0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0,666666667 + 0) / 10 =$$

$$\mathbf{0,166666667}$$

c. Perhitungan *Recall*

FN(1) = 2 ; FN(2) = 0 ; FN (3) = 0 ; FN (4) = 6 ; FN (5) = 0 ; FN (6) = 1;
FN (7) = 5 ; FN (8) = 0 ; FN (9) = 3; FN (10) = 0;

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

R(1) = 0/(0+2) = 0 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 1/(1+6)
= 0,142857143; R(5) = 0/(0+0) = 0; R(6) = 0/(0+1) = 0; R(7) = 0/(0+5) = 0;
R(8) = 0/(0+0) = 0; R(9) = 2/(2+3) = 0,4 ; R(10) = 0/(0+0) = 0

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0+0+0+0,142857143+0+0+0+0+0,4+0) / 10 = \mathbf{0,054285714}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * \mathbf{0,166666667} * \mathbf{0,054285714}) / (\mathbf{0,166666667} + \mathbf{0,054285714}) = \mathbf{0,081896552}$$

15. Percobaan 15

Pada percobaan 15 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.15 Percobaan pengujian 15

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 0 | 120 | 100 | 1 | 1 | 3 |
| 2 | 3 | 180 | 97 | 1 | 1 | 3 |
| 3 | 5 | 204 | 95 | 2 | 1 | 3 |
| 4 | 7 | 223 | 95 | 2 | 1 | 3 |
| 5 | 7 | 224 | 88 | 2 | 2 | 3 |
| 6 | 8 | 232 | 87 | 2 | 2 | 3 |
| 7 | 9 | 235 | 85 | 2 | 2 | 3 |
| 8 | 10 | 240 | 83 | 3 | 2 | 3 |
| 9 | 11 | 251 | 83 | 3 | 2 | 1 |
| 10 | 11 | 259 | 81 | 3 | 2 | 1 |
| 11 | 11 | 259 | 77 | 3 | 3 | 1 |
| 12 | 11 | 262 | 77 | 3 | 2 | 1 |
| 13 | 12 | 263 | 76 | 3 | 2 | 1 |
| 14 | 15 | 265 | 75 | 4 | 2 | 4 |
| 15 | 16 | 268 | 73 | 4 | 2 | 4 |
| 16 | 21 | 271 | 70 | 5 | 2 | 5 |
| 17 | 25 | 275 | 61 | 6 | 2 | 5 |
| 18 | 33 | 277 | 55 | 7 | 2 | 5 |
| 19 | 39 | 279 | 22 | 8 | 2 | 9 |
| 20 | 46 | 299 | 20 | 10 | 1 | 9 |

Dari hasil level yang telah diperoleh pada tabel 4.15 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 3 data berjumlah 4. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 4 / 20$$

$$\text{Akurasi} = \mathbf{0,2}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP (1)} = 2 ; \text{FP (2)} = 5 ; \text{FP (3)} = 5 ; \text{FP (4)} = 0 ; \text{FP(5)} = 0 ; \text{FP(6)} = 1 ; \\ \text{FP(7)} = 1 ; \text{FP(8)} = 1 ; \text{FP(9)} = 0 ; \text{FP(10)} = 1$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{P(1)} = 0/(0+2) = 0 ; \text{P(2)} = 0/(0+5) = 0 ; \text{P(3)} = 1/(1+5) = 0,166666667 ; \\ \text{P(4)} = 2/(2+0) = 1 ; \text{P(5)} = 1/(1+0) = 1 ; \text{P(6)} = 0/(0+1) = 0 ; \text{P(7)} = 0/(0+1) \\ = 0 ; \text{P(8)} = 0/(0+1) = 0 ; \text{P(9)} = 0/(0+0) = 0 ; \text{P(10)} = 0/(0+1) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (\text{P(1)} + \text{P(2)} + \text{P(3)} + \text{P(4)} + \text{P(5)} + \text{P(6)} + \text{P(7)} + \text{P(8)} + \text{P(9)} + \\ \text{P(10)}) / \text{Jumlah kelas}$$

$$\text{Precision} = (0+0+0,166666667+1+1+0+0+0+0+0) / 10 = \mathbf{0,216666667}$$

c. Perhitungan *Recall*

$$\text{FN(1)} = 5 ; \text{FN(2)} = 0 ; \text{FN (3)} = 7 ; \text{FN (4)} = 0 ; \text{FN (5)} = 2 ; \text{FN (6)} = 0 ; \\ \text{FN (7)} = 0 ; \text{FN (8)} = 0 ; \text{FN (9)} = 2 ; \text{FN (10)} = 0 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$$

$$R(1) = 0/(0+5) = 0 ; R(2) = 0/(0+0) = 0 ; R(3) = 1/(1+7) = 0,125 ; R(4) = 2/(2+0) = 1 ; R(5) = 1/(1+2) = 0,333333333 ; R(6) = 0/(0+0) = 0 ; R(7) = 0/(0+0) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+2) = 0 ; R(10) = 0/(0+0) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0+0+0,125+1+0,333333333+0+0+0+0+0) / 10 = \mathbf{0,145833333}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * \mathbf{0,216666667} * \mathbf{0,145833333}) / (\mathbf{0,216666667} + \mathbf{0,145833333}) = 0,113432836$$

16. Percobaan 16

Pada percobaan 16 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.16 Percobaan pengujian 16

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 7 | 280 | 75 | 2 | 2 | 1 |
| 2 | 14 | 200 | 69 | 3 | 3 | 1 |
| 3 | 15 | 200 | 68 | 4 | 3 | 1 |
| 4 | 21 | 199 | 67 | 5 | 4 | 6 |
| 5 | 22 | 193 | 67 | 5 | 4 | 6 |
| 6 | 24 | 191 | 66 | 5 | 4 | 6 |
| 7 | 26 | 189 | 64 | 6 | 4 | 6 |
| 8 | 28 | 187 | 61 | 6 | 4 | 6 |
| 9 | 28 | 186 | 61 | 6 | 4 | 6 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 10 | 29 | 186 | 60 | 6 | 5 | 6 |
| 11 | 29 | 185 | 59 | 6 | 5 | 6 |
| 12 | 29 | 184 | 57 | 6 | 5 | 6 |
| 13 | 30 | 182 | 55 | 7 | 5 | 6 |
| 14 | 31 | 176 | 55 | 7 | 5 | 6 |
| 15 | 31 | 174 | 54 | 7 | 5 | 6 |
| 16 | 31 | 173 | 52 | 7 | 5 | 6 |
| 17 | 31 | 166 | 49 | 7 | 6 | 6 |
| 18 | 32 | 165 | 47 | 7 | 6 | 6 |
| 19 | 36 | 165 | 38 | 8 | 7 | 7 |
| 20 | 41 | 100 | 14 | 9 | 9 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.16 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 1 data berjumlah 6. Selanjutnya dilakukan perhitungan sebagai berikut:

- a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 6 / 20$$

$$\text{Akurasi} = 0,3$$

- b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$\text{FP (1)} = 0 ; \text{FP (2)} = 1 ; \text{FP (3)} = 1 ; \text{FP (4)} = 1 ; \text{FP(5)} = 3 ; \text{FP(6)} = 0 ; \text{FP(7)} = 6 ; \text{FP(8)} = 1 ; \text{FP(9)} = 1 ; \text{FP(10)} = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\textit{Precision} = TP / (TP + FP)$$

$$P(1) = 0/(0+0) = 0 ; P(2) = 0/(0+1) = 0 ; P(3) = 0/(0+1) = 0 ; P(4) = 0/(0+1) = 0 ; P(5) = 0/(0+3) = 0 ; P(6) = 6/(6+0) = 1 ; P(7) = 0/(0+6) = 0 ; P(8) = 0/(0+1) = 0 ; P(9) = 0/(0+1) = 0 ; P(10) = 0/(0+0) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\textit{Precision} = (P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10)) / \text{Jumlah kelas}$$

$$\textit{Precision} = (0+0+0+0+0+1+0+0+0+0) / 10 = \mathbf{0,1}$$

c. Perhitungan *Recall*

$$FN(1) = 3 ; FN(2) = 0 ; FN(3) = 0 ; FN(4) = 0 ; FN(5) = 0 ; FN(6) = 9 ; FN(7) = 1 ; FN(8) = 0 ; FN(9) = 0 ; FN(10) = 1 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\textit{Recall} = TP/(TP+FN)$$

$$R(1) = 0/(0+3) = 0 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 6/(6+9) = 0,4 ; R(7) = 0/(0+1) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+0) = 0 ; R(10) = 0/(0+1) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$\textit{Recall} = (0+0+0+0+0+0,4+0+0+0+0) / 10 = \mathbf{0,04}$$

d. Perhitungan *F-Measure*

$$\textit{F-Measure} = (2 * \textit{Precision} * \textit{Recall}) / (\textit{Precision} + \textit{Recall})$$

$$F\text{-Measure} = (2 * 0,1 * 0,04) / (0,1 + 0,04) = 0,057142857$$

17. Percobaan 17

Pada percobaan 17 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.17 Percobaan pengujian 17

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 2 | 156 | 40 | 1 | 1 | 3 |
| 2 | 8 | 139 | 39 | 2 | 2 | 3 |
| 3 | 16 | 130 | 37 | 4 | 3 | 3 |
| 4 | 20 | 124 | 36 | 5 | 4 | 6 |
| 5 | 21 | 122 | 35 | 5 | 4 | 6 |
| 6 | 23 | 113 | 35 | 5 | 5 | 6 |
| 7 | 24 | 104 | 35 | 5 | 5 | 6 |
| 8 | 25 | 102 | 34 | 6 | 5 | 6 |
| 9 | 27 | 102 | 33 | 6 | 6 | 6 |
| 10 | 28 | 100 | 30 | 6 | 6 | 4 |
| 11 | 29 | 85 | 26 | 6 | 6 | 4 |
| 12 | 31 | 84 | 23 | 7 | 7 | 4 |
| 13 | 31 | 84 | 15 | 7 | 7 | 4 |
| 14 | 31 | 78 | 13 | 7 | 7 | 4 |
| 15 | 34 | 73 | 12 | 7 | 7 | 4 |
| 16 | 34 | 70 | 12 | 7 | 7 | 4 |
| 17 | 36 | 57 | 11 | 8 | 7 | 4 |
| 18 | 36 | 30 | 10 | 8 | 7 | 4 |
| 19 | 42 | 14 | 6 | 9 | 9 | 10 |
| 20 | 46 | 11 | 5 | 10 | 10 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.17 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran.

Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 2 data berjumlah 3. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 3 / 20$$

$$\text{Akurasi} = \mathbf{0,15}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$FP(1) = 1 ; FP(2) = 1 ; FP(3) = 0 ; FP(4) = 1 ; FP(5) = 4 ; FP(6) = 2 ;$$

$$FP(7) = 5 ; FP(8) = 2 ; FP(9) = 1 ; FP(10) = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$P(1) = 0/(0+1) = 0 ; P(2) = 0/(0+1) = 0 ; P(3) = 0/(0+0) = 0 ; P(4) = 0/(0+1)$$

$$= 0 ; P(5) = 0/(0+4) = 0 ; P(6) = 2/(2+2) = 0,5 ; P(7) = 0/(0+5) = 0 ; P(8) =$$

$$0/(0+2) = 0 ; P(9) = 0/(0+1) = 0 ; P(10) = 1/(1+0) = 1$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (0+0+0+0+0+0,5+0+0+0+1) / 10 = \mathbf{0,15}$$

c. Perhitungan *Recall*

$$FN(1) = 0 ; FN(2) = 0 ; FN(3) = 3 ; FN(4) = 9 ; FN(5) = 0 ; FN(6) = 4 ;$$

$$FN(7) = 0 ; FN(8) = 0 ; FN(9) = 0 ; FN(10) = 1 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 0/(0+0) = 0 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+3) = 0 ; R(4) = 0/(0+9) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 2/(2+4) = 0,333333333 ; R(7) = 0/(0+0) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+0) = 0 ; R(10) = 1/(1+1) = 0,5$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0+0+0+0+0+0,333333333+0+0+0+0,5) / 10 = \mathbf{0,083333333}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * 0,15 * 0,083333333) / (0,15 + 0,083333333) = \mathbf{0,107142857}$$

18. Percobaan 18

Pada percobaan 18 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.18 Percobaan pengujian 18

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 0 | 300 | 99 | 1 | 1 | 1 |
| 2 | 0 | 291 | 95 | 1 | 1 | 1 |
| 3 | 3 | 289 | 94 | 1 | 1 | 1 |
| 4 | 5 | 275 | 92 | 2 | 1 | 1 |
| 5 | 6 | 274 | 91 | 2 | 1 | 1 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 6 | 9 | 269 | 88 | 2 | 2 | 1 |
| 7 | 12 | 265 | 82 | 3 | 2 | 1 |
| 8 | 14 | 255 | 81 | 3 | 2 | 1 |
| 9 | 17 | 247 | 81 | 4 | 2 | 4 |
| 10 | 18 | 236 | 78 | 4 | 3 | 7 |
| 11 | 18 | 201 | 76 | 4 | 3 | 7 |
| 12 | 18 | 191 | 68 | 4 | 4 | 6 |
| 13 | 19 | 188 | 67 | 4 | 4 | 6 |
| 14 | 20 | 183 | 65 | 5 | 4 | 6 |
| 15 | 21 | 175 | 60 | 5 | 4 | 6 |
| 16 | 22 | 172 | 59 | 5 | 5 | 6 |
| 17 | 23 | 172 | 58 | 5 | 5 | 6 |
| 18 | 25 | 167 | 55 | 6 | 5 | 6 |
| 19 | 37 | 150 | 52 | 8 | 5 | 9 |
| 20 | 40 | 140 | 28 | 9 | 8 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.18 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 3 data berjumlah 5. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 5 / 20$$

$$\text{Akurasi} = \mathbf{0,25}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$FP(1) = 0 ; FP(2) = 3 ; FP(3) = 2 ; FP(4) = 4 ; FP(5) = 4 ; FP(6) = 0 ;$$

$$FP(7) = 0 ; FP(8) = 1 ; FP(9) = 1 ; FP(10) = 0$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$Precision = TP / (TP + FP)$$

$$P(1) = 3/(3+0) = 1 ; P(2) = 0/(0+3) = 0 ; P(3) = 0/(0+2) = 0 ; P(4) = 1/(1+4) = 0,2 ; P(5) = 0/(0+4) = 0 ; P(6) = 1/(1+0) = 1 ; P(7) = 0/(0+0) = 0 ; P(8) = 0/(0+1) = 0 ; P(9) = 0/(0+1) = 0 ; P(10) = 0/(0+0) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$Precision = (1+0+0+0,2+0+1+0+0+0+0) / 10 = \mathbf{0,22}$$

c. Perhitungan *Recall*

$$FN(1) = 5 ; FN(2) = 0 ; FN(3) = 0 ; FN(4) = 0 ; FN(5) = 0 ; FN(6) = 6 ; FN(7) = 2 ; FN(8) = 0 ; FN(9) = 1 ; FN(10) = 1 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 3/(3+5) = 0,375 ; R(2) = 0/(0+0) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 1/(1+0) = 1 ; R(5) = 0/(0+0) = 0 ; R(6) = 1/(1+6) = 0,142857143 ; R(7) = 0/(0+2) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 0/(0+1) = 0 ; R(10) = 0/(0+1) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0,375+0+0+1+0+0,142857143+0+0+0+0) / 10 = \mathbf{0,151785714}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F\text{-Measure} = (2 * 0,22 * 0,151785714) / (0,22 + 0,151785714) = 0,179634966$$

19. Percobaan 19

Pada percobaan 19 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.19 Percobaan pengujian 19

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 9 | 148 | 30 | 2 | 2 | 1 |
| 2 | 10 | 146 | 30 | 3 | 2 | 1 |
| 3 | 11 | 135 | 31 | 3 | 2 | 3 |
| 4 | 15 | 125 | 32 | 4 | 3 | 3 |
| 5 | 15 | 110 | 36 | 4 | 3 | 3 |
| 6 | 15 | 108 | 41 | 4 | 3 | 3 |
| 7 | 16 | 105 | 44 | 4 | 3 | 3 |
| 8 | 17 | 100 | 47 | 4 | 4 | 6 |
| 9 | 21 | 88 | 47 | 5 | 4 | 6 |
| 10 | 25 | 81 | 47 | 6 | 5 | 6 |
| 11 | 25 | 81 | 53 | 6 | 5 | 6 |
| 12 | 25 | 81 | 54 | 6 | 5 | 6 |
| 13 | 26 | 64 | 57 | 6 | 5 | 6 |
| 14 | 29 | 59 | 57 | 6 | 5 | 6 |
| 15 | 30 | 46 | 59 | 7 | 5 | 6 |
| 16 | 35 | 38 | 60 | 8 | 5 | 6 |
| 17 | 44 | 30 | 63 | 9 | 4 | 9 |
| 18 | 48 | 28 | 63 | 10 | 4 | 9 |
| 19 | 48 | 28 | 64 | 10 | 4 | 9 |
| 20 | 50 | 25 | 68 | 10 | 4 | 9 |

Dari hasil level yang telah diperoleh pada tabel 4.19 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran.

Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 3 data berjumlah 7. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

$$\text{Akurasi} = TP / TP + TN + FP + FN$$

$$\text{Akurasi} = 7 / 20$$

$$\text{Akurasi} = \mathbf{0,35}$$

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

$$FP(1) = 0 ; FP(2) = 1 ; FP(3) = 1 ; FP(4) = 5 ; FP(5) = 1 ; FP(6) = 0 ;$$

$$FP(7) = 1 ; FP(8) = 1 ; FP(9) = 0 ; FP(10) = 3$$

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\text{Precision} = TP / (TP + FP)$$

$$P(1) = 0/(0+0) = 0 ; P(2) = 0/(0+1) = 0 ; P(3) = 1/(1+1) = 0,5 ; P(4) = 0/(0+5)$$

$$= 0 ; P(5) = 0/(0+1) = 0 ; P(6) = 5/(5+0) = 1 ; P(7) = 0/(0+1) = 0 ; P(8) =$$

$$0/(0+1) = 0 ; P(9) = 1/(1+0) = 1 ; P(10) = 0/(0+3) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\text{Precision} = (0+0+0,5+0+0+1+0+0+1+0) / 10 = \mathbf{0,25}$$

c. Perhitungan *Recall*

$$FN(1) = 2 ; FN(2) = 0 ; FN(3) = 4 ; FN(4) = 0 ; FN(5) = 0 ; FN(6) = 4 ;$$

$$FN(7) = 0 ; FN(8) = 0 ; FN(9) = 3 ; FN(10) = 0 ;$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$Recall = TP/(TP+FN)$$

$$R(1) = 0/(0+2) = 0 ; R(2) = 0/(0+0) = 0 ; R(3) = 1/(1+4) = 0,2 ; R(4) = 0/(0+0) = 0 ; R(5) = 0/(0+0) = 0 ; R(6) = 5/(5+4) = 0,555555556 ; R(7) = 0/(0+0) = 0 ; R(8) = 0/(0+0) = 0 ; R(9) = 1/(1+4) = 0,25 ; R(10) = 0/(0+0) = 0$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$Recall = (0+0+0,2+0+0+0,555555556+0+0+0,25+0) / 10 = \mathbf{0,100555556}$$

d. Perhitungan *F-Measure*

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

$$F-Measure = (2 * 0,25 * 0,100555556) / (0,25 + 0,100555556) = \mathbf{0,143423138}$$

20. Percobaan 20

Pada percobaan 20 ini hasil keseluruhan perhitungan data *input* ditulis pada lembar lampiran. Kemudian dari hasil perhitungan tersebut dilakukan penentuan level dan dilakukan pengujian pada sistem sesuai ketentuan yang terdapat pada bab 3 nomor 3.8, sehingga diperoleh hasil keluaran level sebagai berikut:

Tabel 4.20 Percobaan pengujian 20

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 1 | 3 | 203 | 26 | 1 | 1 | 2 |
| 2 | 10 | 203 | 29 | 3 | 2 | 2 |
| 3 | 10 | 213 | 32 | 3 | 2 | 1 |
| 4 | 11 | 220 | 37 | 3 | 2 | 1 |
| 5 | 13 | 221 | 38 | 3 | 3 | 1 |

| No | Input | | | Hasil Level | | |
|----|-------|-------------|-------------|-------------|-----|-----------|
| | Score | Time Remain | Player Life | Tanpa DCA | DCA | DCA Fuzzy |
| 6 | 14 | 224 | 39 | 3 | 3 | 1 |
| 7 | 14 | 226 | 44 | 3 | 3 | 1 |
| 8 | 16 | 230 | 45 | 4 | 4 | 6 |
| 9 | 16 | 238 | 46 | 4 | 4 | 6 |
| 10 | 16 | 238 | 51 | 4 | 4 | 6 |
| 11 | 18 | 238 | 52 | 4 | 4 | 6 |
| 12 | 19 | 240 | 52 | 4 | 4 | 6 |
| 13 | 24 | 245 | 59 | 5 | 3 | 6 |
| 14 | 25 | 259 | 59 | 6 | 3 | 5 |
| 15 | 25 | 263 | 61 | 6 | 2 | 5 |
| 16 | 33 | 268 | 62 | 7 | 2 | 5 |
| 17 | 41 | 268 | 63 | 9 | 2 | 5 |
| 18 | 45 | 273 | 65 | 10 | 2 | 8 |
| 19 | 46 | 291 | 73 | 10 | 1 | 10 |
| 20 | 47 | 299 | 78 | 10 | 1 | 10 |

Dari hasil level yang telah diperoleh pada tabel 4.20 selanjutnya dilakukan pengujian perhitungan dengan menggunakan *Confusion Matrix*. Semua data dari hasil perhitungan *Matrix Confusion* ditulis pada lembar lampiran. Diketahui data uji berjumlah 20 dengan data *True Positive* (TP) sebanyak 1 data berjumlah 2. Selanjutnya dilakukan perhitungan sebagai berikut:

a. Perhitungan Akurasi

Akurasi = TP / Total Data

Akurasi = 2 / 20

Akurasi = **0,1**

b. Perhitungan *Precision*

Perhitungan presisi dihitung dengan mencari nilai *False Positive* (FP) terlebih dahulu sebagai berikut:

FP (1) = 1 ; FP (2) = 0 ; FP (3) = 0 ; FP (4) = 5 ; FP(5) = 1 ; FP(6) = 2 ;

FP(7) = 1 ; FP(8) = 0 ; FP(9) = 1 ; FP(10) = 1

Setelah diketahui nilai FP masing-masing setiap kelas, selanjutnya dihitung presisi tiap kelas.

$$\textit{Precision} = \textit{TP} / (\textit{TP} + \textit{FP})$$

$$\begin{aligned} P(1) &= 0/(0+1) = 0 ; P(2) = 0/(0+0) = 0 ; P(3) = 0/(0+0) = 0 ; P(4) = 0/(0+5) \\ &= 0 ; P(5) = 0/(0+1) = 0 ; P(6) = 0/(0+2) = 0 ; P(7) = 0/(0+1) = 0 ; P(8) = \\ &0/(0+0) = 0 ; P(9) = 0/(0+1) = 0 ; P(10) = 2/(2+1) = 0,666666667 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai presisi sebagai berikut:

$$\textit{Precision} = (0+0+0+0+0+0+0+0+0+0,666666667) / 10 = \mathbf{0,066666667}$$

c. Perhitungan *Recall*

$$\begin{aligned} \textit{FN}(1) &= 5 ; \textit{FN}(2) = 2 ; \textit{FN}(3) = 0 ; \textit{FN}(4) = 0 ; \textit{FN}(5) = 4 ; \textit{FN}(6) = 6 ; \\ \textit{FN}(7) &= 0 ; \textit{FN}(8) = 1 ; \textit{FN}(9) = 0 ; \textit{FN}(10) = 0 ; \end{aligned}$$

Setelah diketahui nilai FN masing-masing setiap kelas, selanjutnya dilakukan perhitungan *Recall* tiap kelas sebagai berikut:

$$\textit{Recall} = \textit{TP}/(\textit{TP}+\textit{FN})$$

$$\begin{aligned} R(1) &= 0/(0+5) = 0 ; R(2) = 0/(0+2) = 0 ; R(3) = 0/(0+0) = 0 ; R(4) = 0/(0+0) \\ &= 0 ; R(5) = 0/(0+4) = 0 ; R(6) = 0/(0+6) = 0 ; R(7) = 0/(0+0) = 0 ; R(8) = \\ &0/(0+1) = 0 ; R(9) = 0/(0+0) = 0 ; R(10) = 2/(2+0) = 1 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian dijumlahkan dan dihitung kembali untuk menentukan nilai *Recall* sebagai berikut:

$$\textit{Recall} = (0+0+0+0+0+0+0+0+0+1) / 10 = \mathbf{0,1}$$

d. Perhitungan *F-Measure*

$$\textit{F-Measure} = (2 * \textit{Precision} * \textit{Recall}) / (\textit{Precision} + \textit{Recall})$$

$$F\text{-Measure} = (2 * 0,066666667 * 0,1) / (0,066666667 + 0,1) = 0,08$$

4.5 Pembahasan

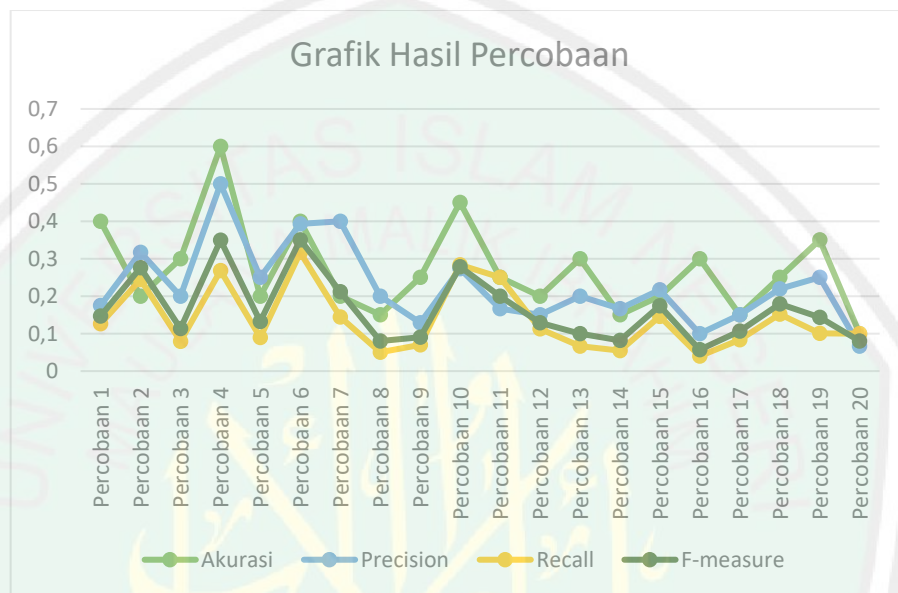
Berdasarkan hasil pengujian yang telah dilakukan, diperoleh data rekapitulasi hasil pengujian. Hasil rekap tersebut digunakan untuk mengetahui nilai-nilai tiap percobaan pengujian yang memiliki nilai terbesar dan terkecil, kemudian dapat diambil nilai rata-rata tiap perhitungan. Berikut rekapitulasi hasil dari 20 percobaan pengujian:

Tabel 4.21 Rekapitulasi Hasil Percobaan

| Uraian | Akurasi | Precision | Recall | F-Measure |
|------------------|-------------|------------------|------------------|------------------|
| Percobaan 1 | 0,4 | 0,175 | 0,12619 | 0,14664 |
| Percobaan 2 | 0,2 | 0,31667 | 0,244444 | 0,275908 |
| Percobaan 3 | 0,3 | 0,2 | 0,079167 | 0,113433 |
| Percobaan 4 | 0,6 | 0,5 | 0,268333 | 0,349241 |
| Percobaan 5 | 0,2 | 0,25 | 0,09 | 0,132353 |
| Percobaan 6 | 0,4 | 0,3928571 | 0,3166667 | 0,3506711 |
| Percobaan 7 | 0,2 | 0,4 | 0,1444444 | 0,2122449 |
| Percobaan 8 | 0,15 | 0,2 | 0,05 | 0,08 |
| Percobaan 9 | 0,25 | 0,1285714 | 0,07 | 0,0906475 |
| Percobaan 10 | 0,45 | 0,2733333 | 0,2833333 | 0,2782435 |
| Percobaan 11 | 0,25 | 0,1666667 | 0,25 | 0,2 |
| Percobaan 12 | 0,2 | 0,15 | 0,1125 | 0,1285714 |
| Percobaan 13 | 0,3 | 0,2 | 0,0666667 | 0,1 |
| Percobaan 14 | 0,15 | 0,1666667 | 0,0542857 | 0,0818966 |
| Percobaan 15 | 0,2 | 0,2166667 | 0,1458333 | 0,1743295 |
| Percobaan 16 | 0,3 | 0,1 | 0,04 | 0,0571429 |
| Percobaan 17 | 0,15 | 0,15 | 0,0833333 | 0,1071429 |
| Percobaan 18 | 0,25 | 0,22 | 0,1517857 | 0,179635 |
| Percobaan 19 | 0,35 | 0,25 | 0,1005556 | 0,1434231 |
| Percobaan 20 | 0,1 | 0,0666667 | 0,1 | 0,08 |
| Total | 5,4 | 4,5230986 | 2,7775388 | 3,2815233 |
| Rata-rata | 0,27 | 0,2261549 | 0,1388769 | 0,1640762 |

Rekapitulasi hasil semua percobaan yang telah diketahui seperti pada tabel 4.21, nilai rata-rata diperoleh dari hasil penjumlahan hasil total akurasi, *Precision*, *Recall* dan *F-Measure* mulai dari percobaan 1 sampai percobaan 20 kemudian

dibagi dengan 20. Sehingga didapatkan rata-rata nilai akurasi sebesar **0,27**, *Precision* sebesar **0,2261549**, *Recall* sebesar **0,1388769**, dan *F-Measure* sebesar **0,1640762**. Keseluruhan hasil pengujian tersebut didapatkan nilai terbesar dan terkecil dari hasil percobaan yang telah dilakukan terangkum pada gambar 4.11.



Gambar 4.11 Grafik hasil pengujian percobaan

Perolehan nilai terbesar dan terkecil seperti pada gambar 4.11 diuraikan sebagai berikut:

1. Akurasi

Akurasi terbesar berada pada percobaan 4 terletak pada level yang sama berjumlah 12 dan dengan nilai sebesar **0,6**

Akurasi terkecil berada pada percobaan 20 terletak pada level yang sama berjumlah 2 dan dengan nilai sebesar **0,1**

2. *Precision*

Precision terbesar berada pada percobaan 4 yang terdapat pada level dengan FP berjumlah 7 dan dengan nilai sebesar **0,5**

Precision terkecil berada pada percobaan 20 yang terdapat pada level dengan FP berjumlah 12 dan dengan nilai sebesar **0,0666667**

3. *Recall*

Recall terbesar berada pada percobaan 6 yang terdapat pada level dengan FN berjumlah 12 dan dengan nilai sebesar **0,3166667**

Recall terkecil berada pada percobaan 16 yang terdapat pada level dengan FN berjumlah 14 dan dengan nilai sebesar **0,04**

4. *F-Measure*

F-Measure terbesar berada pada percobaan 6 yang diperoleh dari menghitung nilai *Precision* dan *Recall* dengan nilai yang dihasilkan sebesar **0,3506711**

F-Measure terkecil berada pada percobaan 16 yang diperoleh dari menghitung nilai *Precision* dan *Recall* dengan nilai yang dihasilkan sebesar **0,0571429**

Pembahasan dari keseluruhan pengujian dalam penelitian ini dengan menggunakan metode *Dynamic Challenging Level Adapter* berhasil diterapkan dengan bantuan *Fuzzy Sugeno* sebagai kecerdasan buatan. *Game* yang telah dibuat berhasil berjalan secara dinamis sesuai dengan kemampuan *player*. Data pengujian yang diambil secara acak berhasil mengeluarkan *output* level yang sesuai target pada *rule fuzzy* yang telah ditentukan.

4.6 Integrasi Islam

Hiburan atau permainan digemari banyak kalangan untuk sekedar bermain dan mengisi waktu luang. Dalam suatu *game* terdapat banyak unsur yang mempengaruhi seseorang merasa senang dalam bermain. Salah satu unsur yang berpengaruh adalah rintangan atau tantangan. Rintangan dalam permainan dibuat

untuk membuat suasana bermain semakin seru dan menyenangkan untuk mencapai posisi kemenangan yang diinginkan. Akan tetapi sering kali rintangan yang dibuat memang terasa susah untuk dihadapi, itu semua untuk menentukan seberapa hebat tingkat kemampuan *player* dalam menjalankan misi.

Dalam Al-Qur'an surat Al-Baqarah ayat 155 dijelaskan mengenai rintangan, ujian atau cobaan, Allah *سبحانه و تعالی* berfirman:

وَلَنَبْلُوَنَّكُمْ بِشَيْءٍ مِّنَ الْخَوْفِ وَالْجُوعِ وَنَقْصٍ مِّنَ الْأَمْوَالِ وَالْأَنْفُسِ وَالثَّمَرَاتِ وَبَشِّرِ الصَّابِرِينَ

Artinya: “Dan sungguh akan Kami berikan cobaan kepadamu, dengan sedikit ketakutan, kelaparan, kekurangan harta, jiwa dan buah-buahan. Dan berikanlah berita gembira kepada orang-orang yang sabar.” (QS. Al-Baqarah: 155)

Pada ayat tersebut dijelaskan bahwa Allah *سبحانه و تعالی* akan memberikan cobaan pada hamba-Nya dan menyeruhkan hambanya untuk bersabar menghadapi ujian, dan bagi hambanya yang sabar ia akan menemukan kebahagiaan. Tafsir Al-Qur'an surat Al-Baqarah ayat 155 menurut Tafsir Jalalain (Dan sungguh Kami akan memberimu cobaan berupa sedikit ketakutan) terhadap musuh, (kelaparan) paceklik, (kekurangan harta) disebabkan datangnya malapetaka, (dan jiwa) disebabkan pembunuhan, kematian dan penyakit, (serta buah-buahan) karena bahaya kekeringan, artinya Kami akan menguji kamu, apakah kamu bersabar atau tidak. (Dan sampaikanlah berita gembira kepada orang-orang yang sabar) bahwa mereka akan menerima ganjaran kesabaran itu berupa surga.

Hadis lain juga menjelaskan tentang balasan pahala besar ujian yang berat, yaitu dari hadis Anas bin Malik, beliau bersabda:

إِنَّ عِظَمَ الْجَزَاءِ مَعَ عِظَمِ الْبَلَاءِ وَإِنَّ اللَّهَ إِذَا أَحَبَّ قَوْمًا ابْتَلَاهُمْ فَمَنْ رَضِيَ فَلَهُ الرِّضَا وَمَنْ سَخِطَ فَلَهُ السَّخَطُ

Artinya: “Sesungguhnya pahala besar karena balasan untuk ujian yang berat. Sungguh, jika Allah mencintai suatu kaum, maka Dia akan menimpakan ujian untuk

mereka. Barangsiapa yang ridho, maka ia yang akan meraih ridho Allah. Barangsiapa siapa yang tidak suka, maka Allah pun akan murka.” (HR. Ibnu Majah no. 4031, hasan kata Syaikh Al Albani).

Keutamaan dari hadis tersebut adalah cobaan dan musibah dinilai sebagai ujian dari Allah untuk hamba-Nya yang beriman, itu merupakan bukti tanda Allah mencintai hamba-Nya. Saat hamba Allah diberi ujian dan cobaan jika hamba tersebut bisa melewati ujian tersebut dengan sabar dan ikhlas maka dia akan mendapatkan pahala besar dari Allah. Begitupun juga penerapannya pada *game* yang dibuat dengan sistem level, dimana seorang *player* menjalankan permainan lalu ia akan menemui beberapa rintangan yang harus dilewati. Jika *player* bisa menyelesaikan dan melewati rintangan dengan baik semakin lama ia akan mendapatkan level yang lebih tinggi dari sebelumnya sehingga ia dapat menyelesaikan permainan dan dapat memenangkan permainan dengan reward skor tinggi dan mencapai level tertinggi

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penerapan atau implementasi dan pengujian yang telah dilakukan pada penelitian ini yaitu pada sistem *game* dengan mengimplementasikan DCA menggunakan *Fuzzy Sugeno* pada level *game* adventure gunung kelud, dapat diambil kesimpulan sebagai berikut:

1. Pembuatan sistem level kabut pada *game* adventure Gunung Kelud dengan metode *Dynamic Challenging Level Adapter* (DCA) dan atau tanpa metode berhasil diterapkan. Sistem level yang tanpa menggunakan metode, hanya berjalan secara statis tidak menyesuaikan perilaku *player*. Sedangkan dengan menggunakan DCA dan *Fuzzy Sugeno* sebagai pengambil keputusan level kabut yang dihasilkan, sistem level pada *game* berjalan sesuai perilaku *player* dengan mengambil tiga variabel *input* untuk menentukan tingkatan level yang dihasilkan sehingga permainan berjalan secara dinamis.
2. Pengujian menggunakan *Confussion Matrix* yang telah dilakukan, digunakan untuk mengukur Akurasi, Presisi, *Recall*, dan *F-Measure*. Tingkat keberhasilan metode yang telah diterapkan pada sistem yang berbeda menghasilkan tingkatan level permainan yang berbeda dan bervariasi dengan melihat perilaku *player*. Pada pengujian tersebut diperoleh nilai Akurasi dan *Precision* terbesar

berada pada percobaan 4 dengan nilai Akurasi sebesar 0,6 dan *Precision* sebesar 0,5, serta untuk nilai terkecil berada pada percobaan 20 dengan nilai Akurasi sebesar 0,2 dan *Precision* sebesar 0,06666667. Sedangkan untuk nilai *Recall* dan *F-Measure* terbesar berada pada percobaan 6 dengan nilai *Recall* sebesar 0,3166667 dan *F-Measure* sebesar 0,3506711, serta untuk nilai terkecil berada pada percobaan 16 dengan nilai *Recall* sebesar 0,04 dan *F-Measure* sebesar 0,0571420039.

5.2 Saran

Pada penelitian ini, Penulis yakin dengan penuh kesadaran bahwa dalam pembuatan sistem *game* yang berfokus pada level atau tingkat kesulitan permainan ini memiliki banyak kekurangan yang nantinya perlu untuk dilakukan pengembangan lagi, diantaranya sebagai berikut:

1. Diharapkan dalam pembuatan sistem *game* yang mengacu pada sistem level atau tingkat kesulitan permainan dapat berfokus pada object-object kesulitan level di dalam *game*, sedangkan pada penelitian ini hanya berfokus pada kesulitan kabut saja.
2. Perlu adanya pengembangan sistem *game* yang lebih baik salah satunya variabel masukan, dimana pada penelitian ini variabel *score* hanya bisa bertambah dan variabel *time remain* yang hanya bisa berkurang. Kedua variabel tersebut diharapkan dapat mempengaruhi perubahan yang dialami *player* dalam bermain dengan adanya pengurangan *score* atau penambahan waktu.

DAFTAR PUSTAKA

- Ahmadi, I. A., Jonemaro, E. M. A., & Akbar, M. A. (2018). Penerapan Algoritme Logika Fuzzy Untuk Dynamic Difficulty Scaling Pada *Game* Labirin. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, 2(10), 3609–3617. <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2685>
- Attoriq, M. R., Anggara, F., & Jumeilah, F. S. (2016). *Penerapan Algoritma Fuzzy Logic Sugeno dan Algoritma A* pada Game Battle City*. x, 1–9.
- Cahyani, B. G. (2020). *IMPLEMENTASI PERLIN NOISE PADA SIMULASI KABUT HETEROGEN GUNUNG KELUD*.
- Chang, S. H., & Yang, N. Y. (2012). DCA: *Dynamic Challenging Level Adapter* for realtime strategy games. *Proceedings - 15th IEEE International Conference on Computational Science and Engineering, CSE 2012 and 10th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2012*, 30–35. <https://doi.org/10.1109/ICCSE.2012.15>
- Guo, F., Tang, J., & Xiao, X. (2014). Foggy scene rendering based on transmission map estimation. *International Journal of Computer Games Technology, 2014*. <https://doi.org/10.1155/2014/308629>
- Hartley, T. P., & Mehdi, Q. H. (2012). In-game adaptation of a navigation mesh cell path. *Proceedings of CGAMES'2012 USA - 17th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational and Serious Games*, 230–236. <https://doi.org/10.1109/CGames.2012.6314580>
- Heckel, F. W. ., Youngblood, G. M., & Ketkar, N. S. (2009). Artificial Intelligence for Computer Games. In *Springer* (pp. 31–54). Springer.
- Kusumadewi, S., & Purnomo, H. (2010). *Aplikasi Logika Fuzzy untuk Pendukung Keputusan edisi 2*. Graha Ilmu.
- Mahmud, Z., Nikentari, N., & Suswaini, E. (2016). Analisa Perbandingan Metode Sugeno Dan Mamdani Dalam Sistem Prediksi Cuaca (Studi Kasus BMKG Kelas III Tanjungpinang). *Teknik Informatika*, 1–9.
- Nugroho, A. K. (2010). Pengendali Logika Fuzzy Suhu Hipertermia Berbasis Visual Basic dan Akuisisi Berbasis USB. *Prosiding Seminar Nasional Sains Dan Teknologi*, 1–9.
- Oktavia, C. A., & Maulidi, R. (2019). Penerapan Logika Fuzzy Sugeno Untuk Penentuan Reward Pada *Game* Edukasi Aku Bisa. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 17(2), 117. <https://doi.org/10.12962/j24068535.v17i2.a825>
- Owen, T. (2009). Artificial Intelligence by Patrick Henry Winston (second

edition) Addison-Wesley Publishing Company, Massachusetts, USA, July 1984 (student hardback edition). In *Robotica* (Vol. 6, Issue 2).
<https://doi.org/10.1017/s0263574700004070>

- Pratama, W. (2014). *Game adventure* Misteri Kotak Pandora. *Jurnal Telematika*, 7(2), 13–31.
- Rahman, A. (2019). *PENENTUAN REKOMENDASI BARANG MENGGUNAKAN SIMPLE ADDITIVE WEIGHTING PADA GAME SIMULASI BENCANA*.
- Riedl, M. O., & Zook, A. (2013). AI for Game Production. *Conference on Computational Intelligence in Games (CIG), 2013 IEEE*.
<https://doi.org/10.1109/CIG.2013.6633663>
- Saurik, H. T. T., Yuniarno, E. M., & Susiki, S. M. (2015). KEPUASAN PEMAIN TERHADAP DESAIN RINTANGAN PADA SKENARIO GAME PETUALANG. *Seminar Nasional Informatika*, 156–164.
- Simske, S. (2019). *Meta-analytic design patterns*.
- Suyanto. (2014). Artificial Intelligence: Searching, Reasoning, Planning and Learning. In *Informatika* (Edisi 2).
- Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*. <https://doi.org/10.1016/j.aci.2018.08.003>
- Warpefelt, H. (2016). *The Non-Player Character: Exploring the believability of NPC presentation and behavior* (Issue 16).

LAMPIRAN

Lampiran hasil perhitungan data *input* percobaan pengujian sebanyak 20 data pengujian.

Hasil perhitungan pengujian ke-1

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 0 | 297 | 82 | 0 | 0,99 | 0,82 | 1,81 |
| 2 | 0 | 286 | 80 | 0 | 0,953333333 | 0,8 | 1,753333333 |
| 3 | 1 | 278 | 72 | 0,5 | 0,926666667 | 0,72 | 2,146666667 |
| 4 | 2 | 257 | 67 | 1 | 0,856666667 | 0,67 | 2,526666667 |
| 5 | 5 | 218 | 63 | 2,5 | 0,726666667 | 0,63 | 3,856666667 |
| 6 | 10 | 196 | 62 | 5 | 0,653333333 | 0,62 | 6,273333333 |
| 7 | 15 | 190 | 62 | 7,5 | 0,633333333 | 0,62 | 8,753333333 |
| 8 | 15 | 190 | 58 | 7,5 | 0,633333333 | 0,58 | 8,713333333 |
| 9 | 17 | 162 | 55 | 8,5 | 0,54 | 0,55 | 9,59 |
| 10 | 23 | 138 | 55 | 11,5 | 0,46 | 0,55 | 12,51 |
| 11 | 31 | 129 | 49 | 15,5 | 0,43 | 0,49 | 16,42 |
| 12 | 33 | 114 | 45 | 16,5 | 0,38 | 0,45 | 17,33 |
| 13 | 37 | 112 | 43 | 18,5 | 0,373333333 | 0,43 | 19,30333333 |
| 14 | 38 | 104 | 43 | 19 | 0,346666667 | 0,43 | 19,77666667 |
| 15 | 41 | 95 | 23 | 20,5 | 0,316666667 | 0,23 | 21,04666667 |
| 16 | 47 | 90 | 21 | 23,5 | 0,3 | 0,21 | 24,01 |
| 17 | 58 | 82 | 14 | 29 | 0,273333333 | 0,14 | 29,41333333 |
| 18 | 59 | 68 | 7 | 29,5 | 0,226666667 | 0,07 | 29,79666667 |
| 19 | 59 | 6 | 2 | 29,5 | 0,02 | 0,02 | 29,54 |
| 20 | 60 | 5 | 2 | 30 | 0,016666667 | 0,02 | 30,03666667 |

Hasil perhitungan pengujian ke-2

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 0 | 275 | 97 | 0 | 0,916666667 | 0,97 | 1,886666667 |
| 2 | 6 | 238 | 92 | 3 | 0,793333333 | 0,92 | 4,713333333 |
| 3 | 8 | 227 | 91 | 4 | 0,756666667 | 0,91 | 5,666666667 |
| 4 | 12 | 211 | 89 | 6 | 0,703333333 | 0,89 | 7,593333333 |
| 5 | 15 | 205 | 88 | 7,5 | 0,683333333 | 0,88 | 9,063333333 |
| 6 | 18 | 187 | 86 | 9 | 0,623333333 | 0,86 | 10,48333333 |
| 7 | 19 | 161 | 86 | 9,5 | 0,536666667 | 0,86 | 10,89666667 |
| 8 | 23 | 157 | 82 | 11,5 | 0,523333333 | 0,82 | 12,84333333 |
| 9 | 23 | 154 | 72 | 11,5 | 0,513333333 | 0,72 | 12,73333333 |
| 10 | 24 | 152 | 69 | 12 | 0,506666667 | 0,69 | 13,19666667 |
| 11 | 36 | 152 | 68 | 18 | 0,506666667 | 0,68 | 19,18666667 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 12 | 39 | 142 | 64 | 19,5 | 0,4733333333 | 0,64 | 20,61333333 |
| 13 | 39 | 114 | 63 | 19,5 | 0,38 | 0,63 | 20,51 |
| 14 | 44 | 98 | 63 | 22 | 0,326666667 | 0,63 | 22,95666667 |
| 15 | 45 | 92 | 60 | 22,5 | 0,306666667 | 0,6 | 23,40666667 |
| 16 | 46 | 90 | 59 | 23 | 0,3 | 0,59 | 23,89 |
| 17 | 48 | 88 | 58 | 24 | 0,2933333333 | 0,58 | 24,87333333 |
| 18 | 53 | 66 | 57 | 26,5 | 0,22 | 0,57 | 27,29 |
| 19 | 54 | 62 | 49 | 27 | 0,206666667 | 0,49 | 27,69666667 |
| 20 | 55 | 34 | 15 | 27,5 | 0,1133333333 | 0,15 | 27,76333333 |

Hasil perhitungan pengujian ke-3

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|-----------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 2 | 292 | 96 | 1 | 0,9733333333 | 0,96 | 2,9333333 |
| 2 | 8 | 290 | 90 | 4 | 0,966666667 | 0,9 | 5,866667 |
| 3 | 8 | 287 | 89 | 4 | 0,956666667 | 0,89 | 5,846667 |
| 4 | 8 | 237 | 83 | 4 | 0,79 | 0,83 | 5,62 |
| 5 | 9 | 217 | 83 | 4,5 | 0,7233333333 | 0,83 | 6,0533333 |
| 6 | 9 | 187 | 81 | 4,5 | 0,6233333333 | 0,81 | 5,9333333 |
| 7 | 10 | 177 | 71 | 5 | 0,59 | 0,71 | 6,3 |
| 8 | 14 | 174 | 55 | 7 | 0,58 | 0,55 | 8,13 |
| 9 | 16 | 149 | 54 | 8 | 0,496666667 | 0,54 | 9,036667 |
| 10 | 23 | 130 | 54 | 11,5 | 0,4333333333 | 0,54 | 12,473333 |
| 11 | 26 | 122 | 52 | 13 | 0,406666667 | 0,52 | 13,92667 |
| 12 | 28 | 117 | 49 | 14 | 0,39 | 0,49 | 14,88 |
| 13 | 28 | 107 | 48 | 14 | 0,356666667 | 0,48 | 14,83667 |
| 14 | 28 | 74 | 47 | 14 | 0,246666667 | 0,47 | 14,71667 |
| 15 | 29 | 59 | 45 | 14,5 | 0,196666667 | 0,45 | 15,14667 |
| 16 | 30 | 49 | 40 | 15 | 0,1633333333 | 0,4 | 15,563333 |
| 17 | 38 | 47 | 39 | 19 | 0,156666667 | 0,39 | 19,54667 |
| 18 | 41 | 41 | 26 | 20,5 | 0,136666667 | 0,26 | 20,89667 |
| 19 | 42 | 39 | 14 | 21 | 0,13 | 0,14 | 21,27 |
| 20 | 44 | 23 | 7 | 22 | 0,076666667 | 0,07 | 22,14667 |

Hasil perhitungan pengujian ke-4

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 0 | 298 | 100 | 0 | 0,9933333333 | 1 | 1,9933333333 |
| 2 | 8 | 281 | 98 | 4 | 0,936666667 | 0,98 | 5,916666667 |
| 3 | 10 | 271 | 96 | 5 | 0,9033333333 | 0,96 | 6,8633333333 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 4 | 18 | 254 | 95 | 9 | 0,846666667 | 0,95 | 10,796666667 |
| 5 | 22 | 241 | 91 | 11 | 0,803333333 | 0,91 | 12,713333333 |
| 6 | 26 | 239 | 90 | 13 | 0,796666667 | 0,9 | 14,696666667 |
| 7 | 27 | 235 | 82 | 13,5 | 0,783333333 | 0,82 | 15,103333333 |
| 8 | 32 | 232 | 79 | 16 | 0,773333333 | 0,79 | 17,563333333 |
| 9 | 34 | 183 | 72 | 17 | 0,61 | 0,72 | 18,33 |
| 10 | 34 | 156 | 71 | 17 | 0,52 | 0,71 | 18,23 |
| 11 | 35 | 143 | 67 | 17,5 | 0,476666667 | 0,67 | 18,646666667 |
| 12 | 36 | 140 | 61 | 18 | 0,466666667 | 0,61 | 19,076666667 |
| 13 | 38 | 130 | 59 | 19 | 0,433333333 | 0,59 | 20,023333333 |
| 14 | 42 | 109 | 42 | 21 | 0,363333333 | 0,42 | 21,783333333 |
| 15 | 45 | 105 | 30 | 22,5 | 0,35 | 0,3 | 23,15 |
| 16 | 48 | 101 | 26 | 24 | 0,336666667 | 0,26 | 24,596666667 |
| 17 | 49 | 87 | 17 | 24,5 | 0,29 | 0,17 | 24,96 |
| 18 | 51 | 86 | 14 | 25,5 | 0,286666667 | 0,14 | 25,926666667 |
| 19 | 52 | 78 | 8 | 26 | 0,26 | 0,08 | 26,34 |
| 20 | 60 | 24 | 7 | 30 | 0,08 | 0,07 | 30,15 |

Hasil perhitungan pengujian ke-5

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 4 | 250 | 100 | 2 | 0,833333333 | 1 | 3,833333333 |
| 2 | 6 | 245 | 95 | 3 | 0,816666667 | 0,95 | 4,766666667 |
| 3 | 6 | 181 | 86 | 3 | 0,603333333 | 0,86 | 4,463333333 |
| 4 | 8 | 179 | 80 | 4 | 0,596666667 | 0,8 | 5,396666667 |
| 5 | 8 | 169 | 77 | 4 | 0,563333333 | 0,77 | 5,333333333 |
| 6 | 9 | 166 | 71 | 4,5 | 0,553333333 | 0,71 | 5,763333333 |
| 7 | 13 | 162 | 70 | 6,5 | 0,54 | 0,7 | 7,74 |
| 8 | 13 | 160 | 70 | 6,5 | 0,533333333 | 0,7 | 7,733333333 |
| 9 | 15 | 141 | 67 | 7,5 | 0,47 | 0,67 | 8,64 |
| 10 | 17 | 133 | 61 | 8,5 | 0,443333333 | 0,61 | 9,553333333 |
| 11 | 20 | 131 | 61 | 10 | 0,436666667 | 0,61 | 11,046666667 |
| 12 | 20 | 122 | 61 | 10 | 0,406666667 | 0,61 | 11,016666667 |
| 13 | 30 | 114 | 60 | 15 | 0,38 | 0,6 | 15,98 |
| 14 | 37 | 105 | 45 | 18,5 | 0,35 | 0,45 | 19,3 |
| 15 | 37 | 102 | 32 | 18,5 | 0,34 | 0,32 | 19,16 |
| 16 | 41 | 89 | 29 | 20,5 | 0,296666667 | 0,29 | 21,086666667 |
| 17 | 42 | 74 | 22 | 21 | 0,246666667 | 0,22 | 21,466666667 |
| 18 | 44 | 73 | 21 | 22 | 0,243333333 | 0,21 | 22,453333333 |
| 19 | 47 | 70 | 5 | 23,5 | 0,233333333 | 0,05 | 23,783333333 |
| 20 | 50 | 69 | 2 | 25 | 0,23 | 0,02 | 25,25 |

Hasil perhitungan pengujian ke-6

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 0 | 296 | 88 | 0 | 0,986666667 | 0,88 | 1,866666667 |
| 2 | 2 | 292 | 84 | 1 | 0,973333333 | 0,84 | 2,813333333 |
| 3 | 6 | 275 | 74 | 3 | 0,916666667 | 0,74 | 4,656666667 |
| 4 | 15 | 270 | 74 | 7,5 | 0,9 | 0,74 | 9,14 |
| 5 | 13 | 265 | 73 | 6,5 | 0,883333333 | 0,73 | 8,113333333 |
| 6 | 24 | 246 | 68 | 12 | 0,82 | 0,68 | 13,5 |
| 7 | 25 | 241 | 60 | 12,5 | 0,803333333 | 0,6 | 13,90333333 |
| 8 | 31 | 240 | 56 | 15,5 | 0,8 | 0,56 | 16,86 |
| 9 | 31 | 240 | 55 | 15,5 | 0,8 | 0,55 | 16,85 |
| 10 | 33 | 240 | 37 | 16,5 | 0,8 | 0,37 | 17,67 |
| 11 | 35 | 240 | 37 | 17,5 | 0,8 | 0,37 | 18,67 |
| 12 | 39 | 188 | 28 | 19,5 | 0,626666667 | 0,28 | 20,40666667 |
| 13 | 40 | 178 | 22 | 20 | 0,593333333 | 0,22 | 20,81333333 |
| 14 | 46 | 170 | 22 | 23 | 0,566666667 | 0,22 | 23,78666667 |
| 15 | 48 | 170 | 20 | 24 | 0,566666667 | 0,2 | 24,76666667 |
| 16 | 48 | 168 | 16 | 24 | 0,56 | 0,16 | 24,72 |
| 17 | 52 | 168 | 14 | 26 | 0,56 | 0,14 | 26,7 |
| 18 | 54 | 147 | 13 | 27 | 0,49 | 0,13 | 27,62 |
| 19 | 56 | 110 | 12 | 28 | 0,366666667 | 0,12 | 28,48666667 |
| 20 | 56 | 100 | 4 | 28 | 0,333333333 | 0,04 | 28,37333333 |

Hasil perhitungan pengujian ke-7

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 3 | 243 | 96 | 1,5 | 0,81 | 0,96 | 3,27 |
| 2 | 7 | 235 | 90 | 3,5 | 0,783333333 | 0,9 | 5,183333333 |
| 3 | 12 | 230 | 89 | 6 | 0,766666667 | 0,89 | 7,656666667 |
| 4 | 17 | 217 | 83 | 8,5 | 0,723333333 | 0,83 | 10,05333333 |
| 5 | 18 | 187 | 83 | 9 | 0,623333333 | 0,83 | 10,45333333 |
| 6 | 19 | 178 | 81 | 9,5 | 0,593333333 | 0,81 | 10,90333333 |
| 7 | 19 | 174 | 71 | 9,5 | 0,58 | 0,71 | 10,79 |
| 8 | 20 | 156 | 55 | 10 | 0,52 | 0,55 | 11,07 |
| 9 | 21 | 143 | 54 | 10,5 | 0,476666667 | 0,54 | 11,51666667 |
| 10 | 23 | 134 | 54 | 11,5 | 0,446666667 | 0,54 | 12,48666667 |
| 11 | 24 | 131 | 52 | 12 | 0,436666667 | 0,52 | 12,95666667 |
| 12 | 24 | 115 | 49 | 12 | 0,383333333 | 0,49 | 12,87333333 |
| 13 | 27 | 89 | 48 | 13,5 | 0,296666667 | 0,48 | 14,27666667 |
| 14 | 30 | 82 | 47 | 15 | 0,273333333 | 0,47 | 15,74333333 |
| 15 | 32 | 82 | 45 | 16 | 0,273333333 | 0,45 | 16,72333333 |
| 16 | 32 | 82 | 40 | 16 | 0,273333333 | 0,4 | 16,67333333 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|---------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 17 | 36 | 63 | 39 | 18 | 0,21 | 0,39 | 18,6 |
| 18 | 40 | 55 | 26 | 20 | 0,1833333333 | 0,26 | 20,4433333333 |
| 19 | 42 | 34 | 14 | 21 | 0,1133333333 | 0,14 | 21,2533333333 |
| 20 | 48 | 26 | 7 | 24 | 0,0866666667 | 0,07 | 24,1566666667 |

Hasil perhitungan pengujian ke-8

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|---------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 1 | 198 | 99 | 0,5 | 0,66 | 0,99 | 2,15 |
| 2 | 5 | 194 | 84 | 2,5 | 0,6466666667 | 0,84 | 3,9866666667 |
| 3 | 10 | 193 | 82 | 5 | 0,6433333333 | 0,82 | 6,4633333333 |
| 4 | 15 | 160 | 70 | 7,5 | 0,5333333333 | 0,7 | 8,7333333333 |
| 5 | 19 | 150 | 70 | 9,5 | 0,5 | 0,7 | 10,7 |
| 6 | 20 | 136 | 63 | 10 | 0,4533333333 | 0,63 | 11,0833333333 |
| 7 | 23 | 124 | 60 | 11,5 | 0,4133333333 | 0,6 | 12,5133333333 |
| 8 | 23 | 121 | 57 | 11,5 | 0,4033333333 | 0,57 | 12,4733333333 |
| 9 | 26 | 109 | 54 | 13 | 0,3633333333 | 0,54 | 13,9033333333 |
| 10 | 29 | 96 | 49 | 14,5 | 0,32 | 0,49 | 15,31 |
| 11 | 30 | 92 | 47 | 15 | 0,3066666667 | 0,47 | 15,7766666667 |
| 12 | 33 | 90 | 46 | 16,5 | 0,3 | 0,46 | 17,26 |
| 13 | 34 | 83 | 28 | 17 | 0,2766666667 | 0,28 | 17,5566666667 |
| 14 | 34 | 77 | 24 | 17 | 0,2566666667 | 0,24 | 17,4966666667 |
| 15 | 36 | 76 | 23 | 18 | 0,2533333333 | 0,23 | 18,4833333333 |
| 16 | 36 | 52 | 22 | 18 | 0,1733333333 | 0,22 | 18,3933333333 |
| 17 | 38 | 50 | 19 | 19 | 0,1666666667 | 0,19 | 19,3566666667 |
| 18 | 39 | 32 | 18 | 19,5 | 0,1066666667 | 0,18 | 19,7866666667 |
| 19 | 40 | 27 | 17 | 20 | 0,09 | 0,17 | 20,26 |
| 20 | 42 | 10 | 15 | 21 | 0,0333333333 | 0,15 | 21,1833333333 |

Hasil perhitungan pengujian ke-9

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 1 | 293 | 100 | 0,5 | 0,9766666667 | 1 | 2,4766666667 |
| 2 | 1 | 291 | 94 | 0,5 | 0,97 | 0,94 | 2,41 |
| 3 | 4 | 269 | 88 | 2 | 0,8966666667 | 0,88 | 3,7766666667 |
| 4 | 6 | 267 | 87 | 3 | 0,89 | 0,87 | 4,76 |
| 5 | 7 | 260 | 84 | 3,5 | 0,8666666667 | 0,84 | 5,2066666667 |
| 6 | 7 | 259 | 82 | 3,5 | 0,8633333333 | 0,82 | 5,1833333333 |
| 7 | 8 | 258 | 79 | 4 | 0,86 | 0,79 | 5,65 |
| 8 | 8 | 256 | 78 | 4 | 0,8533333333 | 0,78 | 5,6333333333 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 9 | 8 | 254 | 70 | 4 | 0,846666667 | 0,7 | 5,546666667 |
| 10 | 9 | 250 | 69 | 4,5 | 0,833333333 | 0,69 | 6,023333333 |
| 11 | 11 | 250 | 63 | 5,5 | 0,833333333 | 0,63 | 6,963333333 |
| 12 | 13 | 248 | 59 | 6,5 | 0,826666667 | 0,59 | 7,916666667 |
| 13 | 13 | 242 | 59 | 6,5 | 0,806666667 | 0,59 | 7,896666667 |
| 14 | 14 | 238 | 58 | 7 | 0,793333333 | 0,58 | 8,373333333 |
| 15 | 15 | 234 | 57 | 7,5 | 0,78 | 0,57 | 8,85 |
| 16 | 21 | 221 | 57 | 10,5 | 0,736666667 | 0,57 | 11,80666667 |
| 17 | 24 | 216 | 48 | 12 | 0,72 | 0,48 | 13,2 |
| 18 | 30 | 156 | 36 | 15 | 0,52 | 0,36 | 15,88 |
| 19 | 36 | 100 | 26 | 18 | 0,333333333 | 0,26 | 18,59333333 |
| 20 | 40 | 76 | 22 | 20 | 0,253333333 | 0,22 | 20,47333333 |

Hasil perhitungan pengujian ke-10

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 1 | 278 | 84 | 0,5 | 0,926666667 | 0,84 | 2,266666667 |
| 2 | 4 | 277 | 84 | 2 | 0,923333333 | 0,84 | 3,763333333 |
| 3 | 6 | 269 | 83 | 3 | 0,896666667 | 0,83 | 4,726666667 |
| 4 | 7 | 260 | 74 | 3,5 | 0,866666667 | 0,74 | 5,106666667 |
| 5 | 8 | 253 | 67 | 4 | 0,843333333 | 0,67 | 5,513333333 |
| 6 | 9 | 248 | 65 | 4,5 | 0,826666667 | 0,65 | 5,976666667 |
| 7 | 9 | 237 | 60 | 4,5 | 0,79 | 0,6 | 5,89 |
| 8 | 14 | 233 | 53 | 7 | 0,776666667 | 0,53 | 8,306666667 |
| 9 | 19 | 230 | 49 | 9,5 | 0,766666667 | 0,49 | 10,75666667 |
| 10 | 19 | 222 | 46 | 9,5 | 0,74 | 0,46 | 10,7 |
| 11 | 23 | 220 | 46 | 11,5 | 0,733333333 | 0,46 | 12,69333333 |
| 12 | 26 | 216 | 44 | 13 | 0,72 | 0,44 | 14,16 |
| 13 | 26 | 212 | 44 | 13 | 0,706666667 | 0,44 | 14,14666667 |
| 14 | 29 | 200 | 41 | 14,5 | 0,666666667 | 0,41 | 15,57666667 |
| 15 | 29 | 188 | 39 | 14,5 | 0,626666667 | 0,39 | 15,51666667 |
| 16 | 31 | 177 | 37 | 15,5 | 0,59 | 0,37 | 16,46 |
| 17 | 36 | 175 | 36 | 18 | 0,583333333 | 0,36 | 18,94333333 |
| 18 | 38 | 173 | 32 | 19 | 0,576666667 | 0,32 | 19,89666667 |
| 19 | 39 | 171 | 29 | 19,5 | 0,57 | 0,29 | 20,36 |
| 20 | 40 | 120 | 20 | 20 | 0,4 | 0,2 | 20,6 |

Hasil perhitungan pengujian ke-11

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 9 | 269 | 87 | 4,5 | 0,896666667 | 0,87 | 6,766666667 |
| 2 | 14 | 256 | 84 | 7 | 0,853333333 | 0,84 | 8,693333333 |
| 3 | 15 | 248 | 81 | 7,5 | 0,826666667 | 0,81 | 9,136666667 |
| 4 | 16 | 229 | 81 | 8 | 0,763333333 | 0,81 | 9,573333333 |
| 5 | 16 | 221 | 80 | 8 | 0,736666667 | 0,8 | 9,536666667 |
| 6 | 20 | 219 | 79 | 10 | 0,73 | 0,79 | 11,52 |
| 7 | 24 | 184 | 74 | 12 | 0,613333333 | 0,74 | 13,35333333 |
| 8 | 24 | 173 | 70 | 12 | 0,576666667 | 0,7 | 13,27666667 |
| 9 | 26 | 150 | 63 | 13 | 0,5 | 0,63 | 14,13 |
| 10 | 28 | 147 | 62 | 14 | 0,49 | 0,62 | 15,11 |
| 11 | 34 | 141 | 59 | 17 | 0,47 | 0,59 | 18,06 |
| 12 | 37 | 140 | 55 | 18,5 | 0,466666667 | 0,55 | 19,51666667 |
| 13 | 37 | 136 | 52 | 18,5 | 0,453333333 | 0,52 | 19,47333333 |
| 14 | 39 | 131 | 49 | 19,5 | 0,436666667 | 0,49 | 20,42666667 |
| 15 | 46 | 127 | 44 | 23 | 0,423333333 | 0,44 | 23,86333333 |
| 16 | 46 | 124 | 43 | 23 | 0,413333333 | 0,43 | 23,84333333 |
| 17 | 46 | 123 | 37 | 23 | 0,41 | 0,37 | 23,78 |
| 18 | 47 | 116 | 32 | 23,5 | 0,386666667 | 0,32 | 24,20666667 |
| 19 | 48 | 115 | 30 | 24 | 0,383333333 | 0,3 | 24,68333333 |
| 20 | 48 | 114 | 29 | 24 | 0,38 | 0,29 | 24,67 |

Hasil perhitungan pengujian ke-12

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 2 | 290 | 71 | 1 | 0,966666667 | 0,71 | 2,676666667 |
| 2 | 5 | 285 | 70 | 2,5 | 0,95 | 0,7 | 4,15 |
| 3 | 5 | 283 | 65 | 2,5 | 0,943333333 | 0,65 | 4,093333333 |
| 4 | 6 | 279 | 62 | 3 | 0,93 | 0,62 | 4,55 |
| 5 | 6 | 276 | 61 | 3 | 0,92 | 0,61 | 4,53 |
| 6 | 9 | 245 | 59 | 4,5 | 0,816666667 | 0,59 | 5,906666667 |
| 7 | 9 | 239 | 52 | 4,5 | 0,796666667 | 0,52 | 5,816666667 |
| 8 | 10 | 236 | 52 | 5 | 0,786666667 | 0,52 | 6,306666667 |
| 9 | 11 | 230 | 47 | 5,5 | 0,766666667 | 0,47 | 6,736666667 |
| 10 | 15 | 222 | 46 | 7,5 | 0,74 | 0,46 | 8,7 |
| 11 | 16 | 213 | 43 | 8 | 0,71 | 0,43 | 9,14 |
| 12 | 17 | 210 | 40 | 8,5 | 0,7 | 0,4 | 9,6 |
| 13 | 19 | 206 | 38 | 9,5 | 0,686666667 | 0,38 | 10,56666667 |
| 14 | 19 | 200 | 35 | 9,5 | 0,666666667 | 0,35 | 10,51666667 |
| 15 | 20 | 198 | 34 | 10 | 0,66 | 0,34 | 11 |
| 16 | 21 | 194 | 33 | 10,5 | 0,646666667 | 0,33 | 11,47666667 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 17 | 22 | 191 | 33 | 11 | 0,636666667 | 0,33 | 11,96666667 |
| 18 | 22 | 188 | 33 | 11 | 0,626666667 | 0,33 | 11,95666667 |
| 19 | 38 | 187 | 32 | 19 | 0,623333333 | 0,32 | 19,94333333 |
| 20 | 44 | 158 | 20 | 22 | 0,526666667 | 0,2 | 22,72666667 |

Hasil perhitungan pengujian ke-13

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 0 | 260 | 40 | 0 | 0,866666667 | 0,4 | 1,266666667 |
| 2 | 1 | 253 | 45 | 0,5 | 0,843333333 | 0,45 | 1,793333333 |
| 3 | 2 | 244 | 48 | 1 | 0,813333333 | 0,48 | 2,293333333 |
| 4 | 4 | 231 | 49 | 2 | 0,77 | 0,49 | 3,26 |
| 5 | 4 | 224 | 51 | 2 | 0,746666667 | 0,51 | 3,256666667 |
| 6 | 5 | 218 | 53 | 2,5 | 0,726666667 | 0,53 | 3,756666667 |
| 7 | 9 | 211 | 59 | 4,5 | 0,703333333 | 0,59 | 5,793333333 |
| 8 | 10 | 205 | 60 | 5 | 0,683333333 | 0,6 | 6,283333333 |
| 9 | 11 | 200 | 62 | 5,5 | 0,666666667 | 0,62 | 6,786666667 |
| 10 | 12 | 198 | 63 | 6 | 0,66 | 0,63 | 7,29 |
| 11 | 13 | 197 | 64 | 6,5 | 0,656666667 | 0,64 | 7,796666667 |
| 12 | 15 | 195 | 64 | 7,5 | 0,65 | 0,64 | 8,79 |
| 13 | 19 | 193 | 67 | 9,5 | 0,643333333 | 0,67 | 10,81333333 |
| 14 | 20 | 193 | 68 | 10 | 0,643333333 | 0,68 | 11,32333333 |
| 15 | 21 | 193 | 71 | 10,5 | 0,643333333 | 0,71 | 11,85333333 |
| 16 | 24 | 188 | 72 | 12 | 0,626666667 | 0,72 | 13,34666667 |
| 17 | 25 | 186 | 73 | 12,5 | 0,62 | 0,73 | 13,85 |
| 18 | 30 | 186 | 75 | 15 | 0,62 | 0,75 | 16,37 |
| 19 | 36 | 185 | 75 | 18 | 0,616666667 | 0,75 | 19,36666667 |
| 20 | 45 | 155 | 75 | 22,5 | 0,516666667 | 0,75 | 23,76666667 |

Hasil perhitungan pengujian ke-14

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 6 | 5 | 12 | 3 | 0,016666667 | 0,12 | 3,136666667 |
| 2 | 11 | 8 | 16 | 5,5 | 0,026666667 | 0,16 | 5,686666667 |
| 3 | 18 | 15 | 17 | 9 | 0,05 | 0,17 | 9,22 |
| 4 | 21 | 20 | 17 | 10,5 | 0,066666667 | 0,17 | 10,73666667 |
| 5 | 28 | 23 | 17 | 14 | 0,076666667 | 0,17 | 14,24666667 |
| 6 | 28 | 37 | 19 | 14 | 0,123333333 | 0,19 | 14,31333333 |
| 7 | 32 | 56 | 24 | 16 | 0,186666667 | 0,24 | 16,42666667 |
| 8 | 34 | 63 | 29 | 17 | 0,21 | 0,29 | 17,5 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 9 | 34 | 91 | 30 | 17 | 0,303333333 | 0,3 | 17,60333333 |
| 10 | 35 | 116 | 34 | 17,5 | 0,386666667 | 0,34 | 18,22666667 |
| 11 | 38 | 125 | 35 | 19 | 0,416666667 | 0,35 | 19,76666667 |
| 12 | 39 | 134 | 44 | 19,5 | 0,446666667 | 0,44 | 20,38666667 |
| 13 | 39 | 148 | 45 | 19,5 | 0,493333333 | 0,45 | 20,44333333 |
| 14 | 40 | 152 | 46 | 20 | 0,506666667 | 0,46 | 20,96666667 |
| 15 | 40 | 152 | 48 | 20 | 0,506666667 | 0,48 | 20,98666667 |
| 16 | 42 | 167 | 51 | 21 | 0,556666667 | 0,51 | 22,06666667 |
| 17 | 47 | 169 | 53 | 23,5 | 0,563333333 | 0,53 | 24,59333333 |
| 18 | 48 | 177 | 60 | 24 | 0,59 | 0,6 | 25,19 |
| 19 | 55 | 177 | 62 | 27,5 | 0,59 | 0,62 | 28,71 |
| 20 | 59 | 180 | 68 | 29,5 | 0,6 | 0,68 | 30,78 |

Hasil perhitungan pengujian ke-15

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 0 | 120 | 100 | 0 | 0,4 | 1 | 1,4 |
| 2 | 3 | 180 | 97 | 1,5 | 0,6 | 0,97 | 3,07 |
| 3 | 5 | 204 | 95 | 2,5 | 0,68 | 0,95 | 4,13 |
| 4 | 7 | 223 | 95 | 3,5 | 0,743333333 | 0,95 | 5,193333333 |
| 5 | 7 | 224 | 88 | 3,5 | 0,746666667 | 0,88 | 5,126666667 |
| 6 | 8 | 232 | 87 | 4 | 0,773333333 | 0,87 | 5,643333333 |
| 7 | 9 | 235 | 85 | 4,5 | 0,783333333 | 0,85 | 6,133333333 |
| 8 | 10 | 240 | 83 | 5 | 0,8 | 0,83 | 6,63 |
| 9 | 11 | 251 | 83 | 5,5 | 0,836666667 | 0,83 | 7,166666667 |
| 10 | 11 | 259 | 81 | 5,5 | 0,863333333 | 0,81 | 7,173333333 |
| 11 | 11 | 259 | 77 | 5,5 | 0,863333333 | 0,77 | 7,133333333 |
| 12 | 11 | 262 | 77 | 5,5 | 0,873333333 | 0,77 | 7,143333333 |
| 13 | 12 | 263 | 76 | 6 | 0,876666667 | 0,76 | 7,636666667 |
| 14 | 15 | 265 | 75 | 7,5 | 0,883333333 | 0,75 | 9,133333333 |
| 15 | 16 | 268 | 73 | 8 | 0,893333333 | 0,73 | 9,623333333 |
| 16 | 21 | 271 | 70 | 10,5 | 0,903333333 | 0,7 | 12,10333333 |
| 17 | 25 | 275 | 61 | 12,5 | 0,916666667 | 0,61 | 14,02666667 |
| 18 | 33 | 277 | 55 | 16,5 | 0,923333333 | 0,55 | 17,97333333 |
| 19 | 39 | 279 | 22 | 19,5 | 0,93 | 0,22 | 20,65 |
| 20 | 46 | 299 | 20 | 23 | 0,996666667 | 0,2 | 24,19666667 |

Hasil perhitungan pengujian ke-16

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 7 | 280 | 75 | 3,5 | 0,933333333 | 0,75 | 5,183333333 |
| 2 | 14 | 200 | 69 | 7 | 0,666666667 | 0,69 | 8,356666667 |
| 3 | 15 | 200 | 68 | 7,5 | 0,666666667 | 0,68 | 8,846666667 |
| 4 | 21 | 199 | 67 | 10,5 | 0,663333333 | 0,67 | 11,83333333 |
| 5 | 22 | 193 | 67 | 11 | 0,643333333 | 0,67 | 12,31333333 |
| 6 | 24 | 191 | 66 | 12 | 0,636666667 | 0,66 | 13,29666667 |
| 7 | 26 | 189 | 64 | 13 | 0,63 | 0,64 | 14,27 |
| 8 | 28 | 187 | 61 | 14 | 0,623333333 | 0,61 | 15,23333333 |
| 9 | 28 | 186 | 61 | 14 | 0,62 | 0,61 | 15,23 |
| 10 | 29 | 186 | 60 | 14,5 | 0,62 | 0,6 | 15,72 |
| 11 | 29 | 185 | 59 | 14,5 | 0,616666667 | 0,59 | 15,70666667 |
| 12 | 29 | 184 | 57 | 14,5 | 0,613333333 | 0,57 | 15,68333333 |
| 13 | 30 | 182 | 55 | 15 | 0,606666667 | 0,55 | 16,15666667 |
| 14 | 31 | 176 | 55 | 15,5 | 0,586666667 | 0,55 | 16,63666667 |
| 15 | 31 | 174 | 54 | 15,5 | 0,58 | 0,54 | 16,62 |
| 16 | 31 | 173 | 52 | 15,5 | 0,576666667 | 0,52 | 16,59666667 |
| 17 | 31 | 166 | 49 | 15,5 | 0,553333333 | 0,49 | 16,54333333 |
| 18 | 32 | 165 | 47 | 16 | 0,55 | 0,47 | 17,02 |
| 19 | 36 | 165 | 38 | 18 | 0,55 | 0,38 | 18,93 |
| 20 | 41 | 100 | 14 | 20,5 | 0,333333333 | 0,14 | 20,97333333 |

Hasil perhitungan pengujian ke-17

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 2 | 156 | 40 | 1 | 0,52 | 0,4 | 1,92 |
| 2 | 8 | 139 | 39 | 4 | 0,463333333 | 0,39 | 4,853333333 |
| 3 | 16 | 130 | 37 | 8 | 0,433333333 | 0,37 | 8,803333333 |
| 4 | 20 | 124 | 36 | 10 | 0,413333333 | 0,36 | 10,77333333 |
| 5 | 21 | 122 | 35 | 10,5 | 0,406666667 | 0,35 | 11,25666667 |
| 6 | 23 | 113 | 35 | 11,5 | 0,376666667 | 0,35 | 12,22666667 |
| 7 | 24 | 104 | 35 | 12 | 0,346666667 | 0,35 | 12,69666667 |
| 8 | 25 | 102 | 34 | 12,5 | 0,34 | 0,34 | 13,18 |
| 9 | 27 | 102 | 33 | 13,5 | 0,34 | 0,33 | 14,17 |
| 10 | 28 | 100 | 30 | 14 | 0,333333333 | 0,3 | 14,63333333 |
| 11 | 29 | 85 | 26 | 14,5 | 0,283333333 | 0,26 | 15,04333333 |
| 12 | 31 | 84 | 23 | 15,5 | 0,28 | 0,23 | 16,01 |
| 13 | 31 | 84 | 15 | 15,5 | 0,28 | 0,15 | 15,93 |
| 14 | 31 | 78 | 13 | 15,5 | 0,26 | 0,13 | 15,89 |
| 15 | 34 | 73 | 12 | 17 | 0,243333333 | 0,12 | 17,36333333 |
| 16 | 34 | 70 | 12 | 17 | 0,233333333 | 0,12 | 17,35333333 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 17 | 36 | 57 | 11 | 18 | 0,19 | 0,11 | 18,3 |
| 18 | 36 | 30 | 10 | 18 | 0,1 | 0,1 | 18,2 |
| 19 | 42 | 14 | 6 | 21 | 0,046666667 | 0,06 | 21,106666667 |
| 20 | 46 | 11 | 5 | 23 | 0,036666667 | 0,05 | 23,086666667 |

Hasil perhitungan pengujian ke-18

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 0 | 300 | 99 | 0 | 1 | 0,99 | 1,99 |
| 2 | 0 | 291 | 95 | 0 | 0,97 | 0,95 | 1,92 |
| 3 | 3 | 289 | 94 | 1,5 | 0,963333333 | 0,94 | 3,403333333 |
| 4 | 5 | 275 | 92 | 2,5 | 0,916666667 | 0,92 | 4,336666667 |
| 5 | 6 | 274 | 91 | 3 | 0,913333333 | 0,91 | 4,823333333 |
| 6 | 9 | 269 | 88 | 4,5 | 0,896666667 | 0,88 | 6,276666667 |
| 7 | 12 | 265 | 82 | 6 | 0,883333333 | 0,82 | 7,703333333 |
| 8 | 14 | 255 | 81 | 7 | 0,85 | 0,81 | 8,66 |
| 9 | 17 | 247 | 81 | 8,5 | 0,823333333 | 0,81 | 10,13333333 |
| 10 | 18 | 236 | 78 | 9 | 0,786666667 | 0,78 | 10,566666667 |
| 11 | 18 | 201 | 76 | 9 | 0,67 | 0,76 | 10,43 |
| 12 | 18 | 191 | 68 | 9 | 0,636666667 | 0,68 | 10,316666667 |
| 13 | 19 | 188 | 67 | 9,5 | 0,626666667 | 0,67 | 10,796666667 |
| 14 | 20 | 183 | 65 | 10 | 0,61 | 0,65 | 11,26 |
| 15 | 21 | 175 | 60 | 10,5 | 0,583333333 | 0,6 | 11,68333333 |
| 16 | 22 | 172 | 59 | 11 | 0,573333333 | 0,59 | 12,16333333 |
| 17 | 23 | 172 | 58 | 11,5 | 0,573333333 | 0,58 | 12,65333333 |
| 18 | 25 | 167 | 55 | 12,5 | 0,556666667 | 0,55 | 13,606666667 |
| 19 | 37 | 150 | 52 | 18,5 | 0,5 | 0,52 | 19,52 |
| 20 | 40 | 140 | 28 | 20 | 0,466666667 | 0,28 | 20,746666667 |

Hasil perhitungan pengujian ke-19

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|-------------|-------------|-------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 9 | 148 | 30 | 4,5 | 0,493333333 | 0,3 | 5,293333333 |
| 2 | 10 | 146 | 30 | 5 | 0,486666667 | 0,3 | 5,786666667 |
| 3 | 11 | 135 | 31 | 5,5 | 0,45 | 0,31 | 6,26 |
| 4 | 15 | 125 | 32 | 7,5 | 0,416666667 | 0,32 | 8,236666667 |
| 5 | 15 | 110 | 36 | 7,5 | 0,366666667 | 0,36 | 8,226666667 |
| 6 | 15 | 108 | 41 | 7,5 | 0,36 | 0,41 | 8,27 |
| 7 | 16 | 105 | 44 | 8 | 0,35 | 0,44 | 8,79 |
| 8 | 17 | 100 | 47 | 8,5 | 0,333333333 | 0,47 | 9,303333333 |

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 9 | 21 | 88 | 47 | 10,5 | 0,2933333333 | 0,47 | 11,263333333 |
| 10 | 25 | 81 | 47 | 12,5 | 0,27 | 0,47 | 13,24 |
| 11 | 25 | 81 | 53 | 12,5 | 0,27 | 0,53 | 13,3 |
| 12 | 25 | 81 | 54 | 12,5 | 0,27 | 0,54 | 13,31 |
| 13 | 26 | 64 | 57 | 13 | 0,2133333333 | 0,57 | 13,783333333 |
| 14 | 29 | 59 | 57 | 14,5 | 0,1966666667 | 0,57 | 15,266666667 |
| 15 | 30 | 46 | 59 | 15 | 0,1533333333 | 0,59 | 15,743333333 |
| 16 | 35 | 38 | 60 | 17,5 | 0,1266666667 | 0,6 | 18,226666667 |
| 17 | 44 | 30 | 63 | 22 | 0,1 | 0,63 | 22,73 |
| 18 | 48 | 28 | 63 | 24 | 0,0933333333 | 0,63 | 24,723333333 |
| 19 | 48 | 28 | 64 | 24 | 0,0933333333 | 0,64 | 24,733333333 |
| 20 | 50 | 25 | 68 | 25 | 0,0833333333 | 0,68 | 25,763333333 |

Hasil perhitungan pengujian ke-20

| No | Input | | | Perhitungan DCA | | | |
|----|-------|-------------|-------------|-----------------|--------------|-------------|--------------|
| | Score | Time Remain | Player Life | Score | Time Remain | Player Life | Total |
| 1 | 3 | 203 | 26 | 1,5 | 0,6766666667 | 0,26 | 2,4366666667 |
| 2 | 10 | 203 | 29 | 5 | 0,6766666667 | 0,29 | 5,9666666667 |
| 3 | 10 | 213 | 32 | 5 | 0,71 | 0,32 | 6,03 |
| 4 | 11 | 220 | 37 | 5,5 | 0,7333333333 | 0,37 | 6,6033333333 |
| 5 | 13 | 221 | 38 | 6,5 | 0,7366666667 | 0,38 | 7,6166666667 |
| 6 | 14 | 224 | 39 | 7 | 0,7466666667 | 0,39 | 8,1366666667 |
| 7 | 14 | 226 | 44 | 7 | 0,7533333333 | 0,44 | 8,1933333333 |
| 8 | 16 | 230 | 45 | 8 | 0,7666666667 | 0,45 | 9,2166666667 |
| 9 | 16 | 238 | 46 | 8 | 0,7933333333 | 0,46 | 9,2533333333 |
| 10 | 16 | 238 | 51 | 8 | 0,7933333333 | 0,51 | 9,3033333333 |
| 11 | 18 | 238 | 52 | 9 | 0,7933333333 | 0,52 | 10,313333333 |
| 12 | 19 | 240 | 52 | 9,5 | 0,8 | 0,52 | 10,82 |
| 13 | 24 | 245 | 59 | 12 | 0,8166666667 | 0,59 | 13,406666667 |
| 14 | 25 | 259 | 59 | 12,5 | 0,8633333333 | 0,59 | 13,953333333 |
| 15 | 25 | 263 | 61 | 12,5 | 0,8766666667 | 0,61 | 13,986666667 |
| 16 | 33 | 268 | 62 | 16,5 | 0,8933333333 | 0,62 | 18,013333333 |
| 17 | 41 | 268 | 63 | 20,5 | 0,8933333333 | 0,63 | 22,023333333 |
| 18 | 45 | 273 | 65 | 22,5 | 0,91 | 0,65 | 24,06 |
| 19 | 46 | 291 | 73 | 23 | 0,97 | 0,73 | 24,7 |
| 20 | 47 | 299 | 78 | 23,5 | 0,9966666667 | 0,78 | 25,276666667 |

Hasil *Confusion Matrix* pada setiap percobaan pengujian sebanyak 20 kali percobaan.

Hasil *Confusion Matrix* pengujian ke-1

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 3 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 5 |

Hasil *Confusion Matrix* pengujian ke-2

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 5 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Hasil *Confusion Matrix* pengujian ke-3

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 1 | 5 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 |

Hasil *Confusion Matrix* pengujian ke-4

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 |

Hasil *Confusion Matrix* pengujian ke-5

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 4 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 |

Hasil *Confusion Matrix* pengujian ke-6

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

Hasil *Confusion Matrix* pengujian ke-7

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 5 | 1 | 3 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 |

Hasil *Confusion Matrix* pengujian ke-8

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 3 | 2 | 2 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 2 | 0 |

Hasil *Confusion Matrix* pengujian ke-9

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 3 | 5 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Hasil *Confusion Matrix* pengujian ke-10

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 2 | 1 | 4 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Hasil *Confusion Matrix* pengujian ke-11

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 4 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

Hasil *Confusion Matrix* pengujian ke-12

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 3 | 4 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Hasil *Confusion Matrix* pengujian ke-13

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 5 | 2 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Hasil *Confusion Matrix* pengujian ke-14

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 1 | 2 | 3 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Hasil *Confusion Matrix* pengujian ke-15

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Hasil *Confusion Matrix* pengujian ke-16

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 3 | 6 | 6 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Hasil *Confusion Matrix* pengujian ke-17

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 2 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Hasil *Confusion Matrix* pengujian ke-18

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 3 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 2 | 4 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Hasil *Confusion Matrix* pengujian ke-19

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 1 | 5 | 1 | 1 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Hasil *Confusion Matrix* pengujian ke-20

| <i>Matrix</i> | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|----|
| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 1 | 0 |
| 6 | 0 | 0 | 0 | 5 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |