

**PENERAPAN METODE BOX MULLER OF GAUSSIAN DISTRIBUTION  
UNTUK MENENTUKAN TINGKAT KESULITAN PADA GAME 3D  
MITIGASI BENCANA ALAM BERBASIS FITUR KABUT**

**SKRIPSI**

**Oleh :**

**AHMAD AHYA ULHAQ  
NIM. 16650092**



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

**2020**

**PENERAPAN METODE BOX MULLER OF GAUSSIAN DISTRIBUTION  
UNTUK MENENTUKAN TINGKAT KESULITAN PADA GAME 3D  
MITIGASI BENCANA ALAM BERBASIS FITUR KABUT**

**SKRIPSI**

**Diajukan kepada:**

**Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :**

**AHMAD AHYA ULHAQ  
NIM. 16650092**

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

**2020**

## HALAMAN PERSETUJUAN

### PENERAPAN METODE BOX MULLER OF GAUSSIAN DISTRIBUTION UNTUK MENENTUKAN TINGKAT KESULITAN PADA GAME 3D MITIGASI BENCANA ALAM BERBASIS FITUR KABUT

SKRIPSI

Oleh :

AHMAD AHYA ULHAQ  
NIM. 16650092

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : 2020

Dosen Pembimbing I

Dosen Pembimbing II

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

Agung Teguh Wibowo Almais, M.T  
NIDT. 19860103201802011235

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

Dr. Cahyo Crysdiyan  
NIP. 19740424 200901 1 008

**PENERAPAN METODE BOX MULLER OF GAUSSIAN DISTRIBUTION  
UNTUK MENENTUKAN TINGKAT KESULITAN PADA GAME 3D  
MITIGASI BENCANA ALAM BERBASIS FITUR KABUT**

**SKRIPSI**

**Oleh :**

**AHMAD AHYA ULHAQ  
NIM. 16650092**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal: 2020

**Susunan Dewan Penguji :**

**Tanda Tangan**

Penguji Utama	:	<u>Dr. Fachrul Kurniawan, M.MT</u> NIP. 19740602 200901 1 010	(	)
Ketua Penguji	:	<u>Khadijah F. H. Holle, M.Kom</u> NIDT. 19900626 20160801 2 077	(	)
Sekertaris Penguji	:	<u>Fresy Nugroho, M.T</u> NIP. 19710722 201101 1 001	(	)
Anggota Penguji	:	<u>Agung Teguh W. Almais, M.T</u> NIDT. 19860103201802011235	(	)

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

Dr. Cahyo Crysdiyan  
NIP. 19740424 200901 1 008

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini,

Nama : Ahmad Ahya Ulhaq  
NIM : 16650092  
Jurusan : Teknik Informatika  
Fakultas : Sains dan Teknologi  
Judul Skripsi : Penerapan Metode *Box Muller Of Gaussian Distribution*  
Untuk Menentukan Tingkat Kesulitan Pada Game 3D  
Mitigasi Bencana Alam Berbasis Fitur Kabut

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan 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 saya tersebut.

Malang, 22 Juli 2020

Yang membuat pernyataan,



Ahmad Ahya Ulhaq

NIM. 16650092

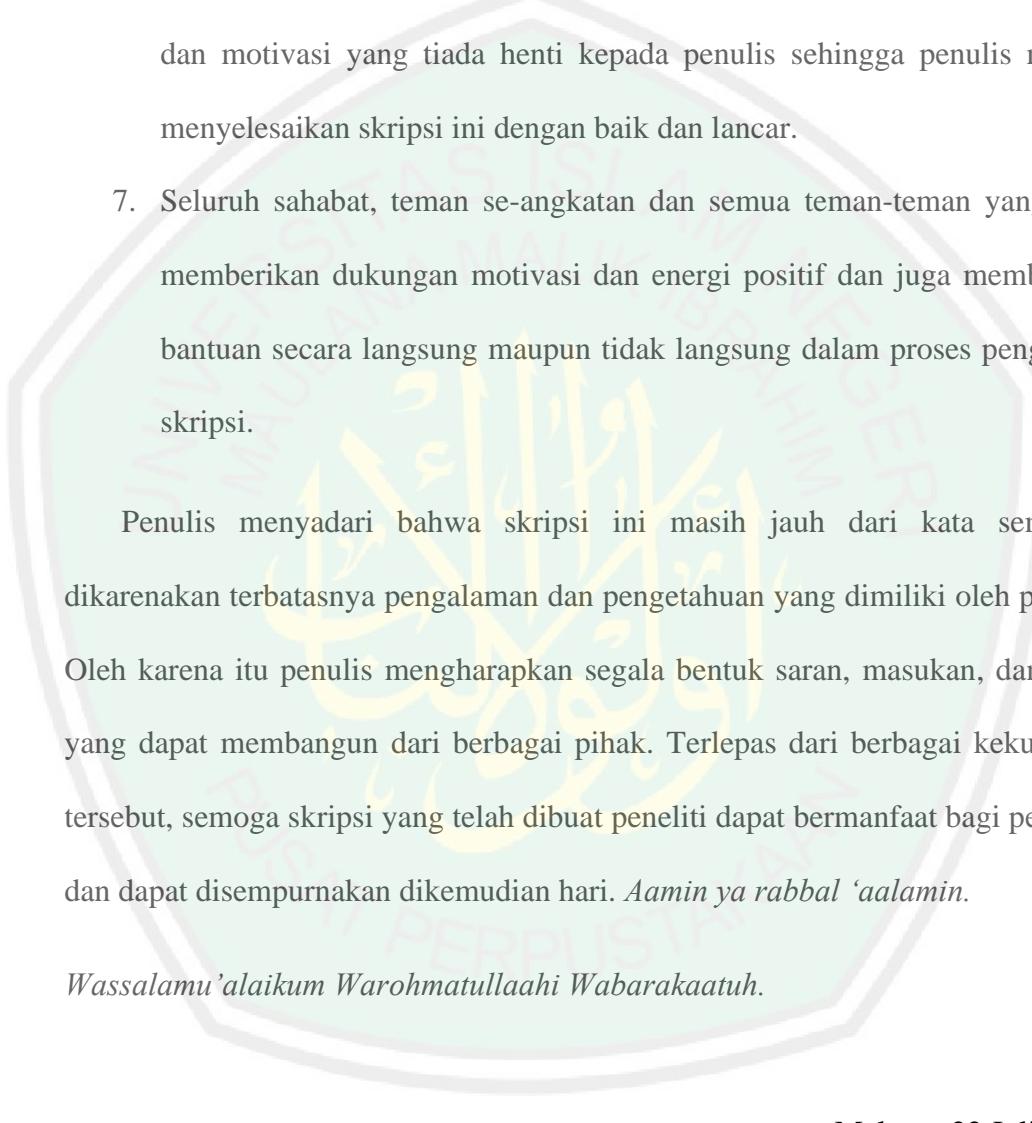
## KATA PENGANTAR

*Assalamu'alaikum Warohmatullaahi Wabarakaaatuh.*

Segala puji bagi Allah SWT Tuhan yang Maha Esa, karena dengan rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Metode *Box Muller of Gaussian Distribution* untuk Menentukan Tingkat Kesulitan pada *Game 3D Mitigasi Bencana Alam Berbasis Fitur Kabut*” dengan baik dan lancar. Shalawat serta salam selalu dipanjatkan kepada junjungan kita Nabi Muhammad SAW yang telah membawa umatnya keluar dari zaman jahiliah menuju zaman yang terang benderang yaitu Islam *rahmatan lil alamin*.

Dalam proses penyelesaian skripsi ini, penulis mendapatkan banyak bantuan, bimbingan serta dukungan berbagai pihak. Oleh sebab itu dengan rasa syukur dan kelapangan hati penulis mengucapkan terima kasih dan doa yang tulus kepada:

1. Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Fresy Nugroho, M.T dan Agung Teguh Wibowo Almais, M.T, selaku dosen pembimbing yang telah bersedia dan sabar dalam membimbing dan memberi masukan dan arahan kepada penulis dalam menyelesaikan skripsi ini hingga selesai.
3. Dr. Fachrul Kurniawan, M.MT dan Khadijah Fahmi Hayati Holle, M.Kom, selaku dosen penguji yang telah menguji ujian skripsi penulis mulai dari seminar proposal sampai dengan sidang skripsi dengan lancar dan profesional.

- 
4. Seluruh jajaran dosen Jurusan Teknik Informatika yang telah memberikan ilmu yang berharga bagi penulis selama menjalani studi.
  5. Seluruh jajaran staf Jurusan Teknik Informatika yang telibat langsung maupun tidak langsung dalam proses penggerjaan skripsi.
  6. Bapak, Ibu, Nenek dan keluarga besar tercinta yang selalu memberikan doa dan motivasi yang tiada henti kepada penulis sehingga penulis mampu menyelesaikan skripsi ini dengan baik dan lancar.
  7. Seluruh sahabat, teman se-angkatan dan semua teman-teman yang telah memberikan dukungan motivasi dan energi positif dan juga memberikan bantuan secara langsung maupun tidak langsung dalam proses penggerjaan skripsi.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki oleh peneliti. Oleh karena itu penulis mengharapkan segala bentuk saran, masukan, dan kritik yang dapat membangun dari berbagai pihak. Terlepas dari berbagai kekurangan tersebut, semoga skripsi yang telah dibuat peneliti dapat bermanfaat bagi pembaca dan dapat disempurnakan di kemudian hari. *Aamin ya rabbal 'aalamin.*

*Wassalamu 'alaikum Warohmatullaahi Wabarakaaatu.*

Malang, 22 Juli 2020

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PERSETUJUAN .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iii</b>
<b>PERNYATAAN KEASLIAN TULISAN .....</b>	<b>Error! Bookmark not defined.</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>ABSTRAK .....</b>	<b>xii</b>
<b>ABSTRACT .....</b>	<b>xiii</b>
<b>الملخص .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1.    Latar Belakang Masalah .....	1
1.2.    Pernyataan Masalah .....	3
1.3.    Tujuan Penelitian.....	4
1.4.    Manfaat Penelitian.....	4
1.4.1.    Untuk <i>desainer game</i> .....	4
1.4.2.    Untuk <i>player</i> .....	4
1.5.    Batasan Masalah .....	4
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>6</b>
2.1.    Penelitian Terkait.....	6
2.2. <i>Game Adventure</i> .....	6
2.3. <i>Non-Player Character (NPC)</i> .....	7
2.4.    Kabut ( <i>fog</i> ).....	7
2.5. <i>Box-Muller</i> .....	9
2.6. <i>Gaussian Distribution</i> .....	10
2.7. <i>Box-Muller of Gaussian Distribution</i> .....	11
2.8.    Rata-rata.....	12
2.9.    Modus.....	12
2.10.    Matriks Konfusi .....	12
2.10.1.    Akurasi.....	13
2.10.2.    Precision .....	13
2.10.3.    Recall.....	13

<b>2.10.4. F-Measure.....</b>	<b>13</b>
<b>BAB III METODE PENELITIAN .....</b>	<b>14</b>
<b>3.1. Penelitian Sebelumnya.....</b>	<b>14</b>
<b>3.2. Desain Sistem Penelitian Sebelumnya.....</b>	<b>14</b>
<b>3.3. Desain Sistem.....</b>	<b>15</b>
<b>3.4. Skenario Game .....</b>	<b>16</b>
<b>3.5. Perhitungan Manual <i>Box-Muller of Gaussian Distribution</i> .....</b>	<b>18</b>
<b>3.6. Perancangan Pengujian <i>Box-Muller of Gaussian Distribution</i> .....</b>	<b>25</b>
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>35</b>
<b>4.1. Implementasi .....</b>	<b>35</b>
<b>4.2. Implementasi Antarmuka Game.....</b>	<b>35</b>
<b>4.2.1. Tampilan Awal Game .....</b>	<b>35</b>
<b>4.2.2. Tampilan Tahap 1 .....</b>	<b>36</b>
<b>4.2.3. Tampilan Tahap 2 .....</b>	<b>36</b>
<b>4.2.4. Tampilan Tahap 3 .....</b>	<b>37</b>
<b>4.2.5. Tampilan Tahap 4 .....</b>	<b>38</b>
<b>4.2.6. Tampilan Game Win .....</b>	<b>39</b>
<b>4.2.7. Tampilan Game Lose .....</b>	<b>39</b>
<b>4.2.8. Tampilan Input Nilai Variabel .....</b>	<b>40</b>
<b>4.3. Tahap Pengujian .....</b>	<b>41</b>
<b>4.3.1. Pembuatan Sistem Level.....</b>	<b>41</b>
<b>4.3.2. Objek Pendukung Sistem Level .....</b>	<b>42</b>
<b>4.4. Hasil Pengujian .....</b>	<b>42</b>
<b>4.4.1. Percobaan 1.....</b>	<b>43</b>
<b>4.4.2. Percobaan 2 .....</b>	<b>47</b>
<b>4.4.3. Percobaan 3 .....</b>	<b>52</b>
<b>4.4.4. Percobaan 4.....</b>	<b>56</b>
<b>4.4.5. Percobaan 5.....</b>	<b>61</b>
<b>BAB V PENUTUP.....</b>	<b>69</b>
<b>5.1. Kesimpulan.....</b>	<b>69</b>
<b>5.2. Saran .....</b>	<b>69</b>
<b>DAFTAR PUSTAKA .....</b>	<b>71</b>

## DAFTAR GAMBAR

Gambar 2.1 Sampel Terdistribusi Normal .....	10
Gambar 2.2 Kurva Distribusi Gaussian .....	11
Gambar 3.1 <i>Road Map</i> Penelitian.....	14
Gambar 3.2 Desain Sistem <i>Game</i> yang digunakan pada Penelitian Sebelumnya.....	15
Gambar 3.3 Desain Sistem <i>Box-Muller of Gaussian Distribution</i> pada <i>Game</i> .....	16
Gambar 3.4 Skenario <i>Game</i> .....	17
Gambar 3.5 Algoritma perhitungan <i>Box-Muller of Gaussian Distribution</i> .....	18
Gambar 4.1 Tampilan <i>Scenes</i> Awal <i>Game</i> .....	35
Gambar 4.2 Tampilan Tahap 1 .....	36
Gambar 4.3 Tampilan Tahap 2 .....	36
Gambar 4.4 Rintangan di dalam <i>Game</i> .....	37
Gambar 4.5 Panel Variabel di dalam <i>Game</i> .....	37
Gambar 4.6 Panel Score terisi di dalam <i>Game</i> .....	38
Gambar 4.7 Perubahan Level Kabut di dalam <i>Game</i> .....	39
Gambar 4.8 Panel <i>Game Win</i> .....	39
Gambar 4.9 Panel <i>Game Lose</i> .....	40
Gambar 4.10 Panel Input Nilai Variabel <i>BMGD</i> .....	40
Gambar 4.11 Panel <i>Script</i> Sistem Level <i>BMGD</i> pada <i>Unity</i> .....	41
Gambar 4.12 Objek Pendukung Sistem Level <i>Game</i> .....	42

## DAFTAR TABEL

<b>Tabel 2.1 Densitas 12 Citra Uji .....</b>	<b>8</b>
<b>Tabel 2.2 Fitur Kabut dalam <i>Game</i>.....</b>	<b>9</b>
<b>Tabel 2.3 Klasifikasi dan Prediksi .....</b>	<b>12</b>
<b>Tabel 3.1 Nilai Bilangan Acak U dan V .....</b>	<b>18</b>
<b>Tabel 3.2 Nilai S .....</b>	<b>19</b>
<b>Tabel 3.3 Nilai z0 dan z1.....</b>	<b>19</b>
<b>Tabel 3.4 Nilai rr pada <math>\mu_1</math> .....</b>	<b>20</b>
<b>Tabel 3.5 Nilai rr pada <math>\mu_2</math> .....</b>	<b>21</b>
<b>Tabel 3.6 Nilai rr pada <math>\mu_3</math> .....</b>	<b>21</b>
<b>Tabel 3.7 Nilai rr pada <math>\mu_4</math> .....</b>	<b>21</b>
<b>Tabel 3.8 Nilai rr pada <math>\mu_5</math> .....</b>	<b>22</b>
<b>Tabel 3.9 Nilai rr pada <math>\mu_6</math> .....</b>	<b>22</b>
<b>Tabel 3.10 Nilai rr pada <math>\mu_7</math> .....</b>	<b>23</b>
<b>Tabel 3.11 Nilai rr pada <math>\mu_8</math> .....</b>	<b>23</b>
<b>Tabel 3.12 Nilai rr pada <math>\mu_9</math> .....</b>	<b>24</b>
<b>Tabel 3.13 Nilai rr pada <math>\mu_{10}</math> .....</b>	<b>24</b>
<b>Tabel 3.14 Fitur Kabut dalam <i>between</i> .....</b>	<b>25</b>
<b>Tabel 3.15 Penentuan Nilai Setiap Variabel .....</b>	<b>25</b>
<b>Tabel 3.16 Penentuan Level Kabut dari Total Poin .....</b>	<b>26</b>
<b>Tabel 3.17 Ketentuan Tanpa BMGD .....</b>	<b>28</b>
<b>Tabel 3.18 Contoh Hasil Level Kabut BMGD dan Tanpa BMGD.....</b>	<b>29</b>
<b>Tabel 3.19 Contoh <i>Confusion Matrix</i> Hasil Level .....</b>	<b>30</b>
<b>Tabel 3.20 <i>Confusion Matrix</i> (<i>FN</i>) .....</b>	<b>32</b>
<b>Tabel 4.1 Hasil Percobaan 1.....</b>	<b>43</b>
<b>Tabel 4.2 <i>Confusion Matrix</i> Percobaan 1 .....</b>	<b>44</b>
<b>Tabel 4.3 <i>Confusion Matrix</i> (<i>FN</i>) Percobaan 1.....</b>	<b>46</b>
<b>Tabel 4.4 Hasil Percobaan 2.....</b>	<b>47</b>
<b>Tabel 4.5 <i>Confusion Matrix</i> Percobaan 2 .....</b>	<b>48</b>
<b>Tabel 4.6 <i>Confusion Matrix</i> (<i>FN</i>) Percobaan 2.....</b>	<b>50</b>
<b>Tabel 4.7 Hasil Percobaan 3.....</b>	<b>52</b>
<b>Tabel 4.8 <i>Confusion Matrix</i> Percobaan 3 .....</b>	<b>53</b>
<b>Tabel 4.9 <i>Confusion Matrix</i> (<i>FN</i>) Percobaan 3.....</b>	<b>55</b>
<b>Tabel 4.10 Hasil Percobaan 4.....</b>	<b>56</b>

Tabel 4.11 <i>Confusion Matrix</i> Percobaan 4.....	57
Tabel 4.12 <i>Confusion Matrix (FN)</i> Percobaan 4.....	59
Tabel 4.13 Hasil Percobaan 5.....	61
Tabel 4.14 <i>Confusion Matrix</i> Percobaan 5 .....	62
Tabel 4.15 <i>Confusion Matrix (FN)</i> Percobaan 5.....	64
Tabel 4.16 Rekapitulasi Perhitungan <i>Confusion Matrix</i> .....	65
Tabel 4.17 Rekapitulasi Perbandingan Level Kabut BMGD dan Tanpa BMGD.....	66

## ABSTRAK

Ulhaq, Ahmad Ahya. 2020. **Penerapan Metode Box Muller of Gaussian Distribution untuk Menentukan Tingkat Kesulitan pada Game 3D Mitigasi Bencana Alam Berbasis Fitur Kabut.** Skripsi. Jurusan Teknik Informatika Fakultas sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Fresy Nugroho, M.T, (II) Agung Teguh Wibowo Almais, M.T

---

Kata Kunci: *Box Muller, Gaussian Distribution, Tingkat Kesulitan Dinamis, Confusion Matrix*

Skenario level kesulitan pada game merupakan komponen penting dalam pembuatan game. Dalam penelitian ini, penentuan skenario level kesulitan akan dibangun dengan metode *Box Muller of Gaussian Distribution* supaya skenario level kesulitan dapat berjalan secara dinamis atau otomatis, sehingga skenario bisa dipilih secara acak dan tidak bisa diprediksi oleh pemain. Pada penelitian ini skenario tingkat kesulitan digunakan pada *game 3D* mitigasi bencana alam berbasis fitur kabut. Dalam pembuatan sistem level, peneliti menggunakan variabel masukan yaitu variabel *Score, Time Remain* dan *Player Life* dimana variabel *Score* dan *Player Life* didapatkan dari hasil kinerja *Player*. Pada tahap pengujian akan dilakukan perbandingan antara game dengan metode *BMGD* dan tanpa metode. Pengujian dilakukan dengan menggunakan *Confusion Matrix* untuk mengetahui akurasi, *precision, recall* dan *f-measure* pada sistem level dengan menggunakan data hasil level metode *Box Muller of Gaussian Distribution* dan hasil level tanpa metode. Dari tahap pengujian ini didapatkan bahwa sistem level kabut dengan menggunakan metode *BMGD* dapat berjalan secara otomatis dimana penyesuaian dapat berubah secara lebih dinamis dengan hasil terbaik dari pengukuran kinerja metode *BMGD* ditunjukkan pada percobaan pertama dan percobaan kedua dengan nilai yang sama yaitu akurasi sebesar 30%, *precision* sebesar 34,8%, *recall* sebesar 40,3%, dan *f-measure* sebesar 37,3%.

## ABSTRACT

Ulhaq, Ahmad Ahya. 2020. **Application of the Box Muller of Gaussian Distribution Method to Determine the Difficulty Level in 3D Game of Natural Disaster Mitigation Based on Fog Features.** Thesis. Informatics Engineering Department of Science and Technology Faculty Islamic State University Maulana Malik Ibrahim Malang. Supervisor: (I) Fresy Nugroho, M.T, (II) Agung Teguh Wibowo AlMais, M.T

---

Keywords: Box Muller, Gaussian Distribution, Dynamic Difficulty Level, Confusion Matrix

The difficulty level scenario in a game is an important component in making a game. In this study, the determination of the difficulty level scenario will be built with the Box Muller of Gaussian Distribution method so that the difficulty level scenario can run dynamically or automatically, so that the scenario can be chosen randomly and cannot be predicted by the player. In this study the difficulty level scenario is used in the 3D fog mitigation natural disaster mitigation game. In making the system level, researchers used input variables namely Score, Time Remain and Player Life variables where Score and Player Life variables were obtained from the Player's performance results. In the testing phase, a comparison between games with the BMGD method and without a method will be performed. The test is carried out using Confussion Matrix to determine the accuracy, precision, recall and f-measure of the system level by using the level of the Box Muller of Gaussian Distribution method level data and the level results without the method. From this testing phase it was found that the fog level system using the BMGD method can run automatically where adjustments can change more dynamically with the best results from the BMGD method performance measurements shown in the first experiment and the second experiment with the same value that is 30% accuracy, precision at 34.8%, recall at 40.3%, and f-measure at 37.3%.

## الملخص

الحق ، احمد احي. ٢٠٢٠ ، تطبيق طريقة لتحديد مستوى الصعوبة في لعبة تخفيف الكوارث الطبيعية على أساس ميزات الضباب أطروحة. قسم الهندسة المعلوماتية ، كلية العلوم والتكنولوجيا ، جامعة إسلام نجري مولانا مالك إبراهيم مالانج .المشرف : (١) فريسي نوغرها ، الماجستير (٢) أغونغ تعوه وييوروو الماييس . ح. الماجستير

**الكلمات الأساسية:** *Gaussian Distribution* ، *Box Muller* ، مستوى الصعوبة الديناميكي

*Confusion Matrix* ،

يعتبر سيناريو مستوى الصعوبة في اللعبة مكوناً مهماً في صناعة اللعبة .في هذه الدراسة *Box Muller of Gaussian Distribution* سيتم بناء تحديد سيناريو مستوى الصعوبة باستخدام طريقة ، بحيث يمكن تشغيل سيناريو مستوى الصعوبة ديناميكياً أو تلقائياً ، بحيث يمكن اختيار السيناريو بشكل عشوائي ولا يمكن للاعب توقعه .في هذه الدراسة ، يتم استخدام سيناريو مستوى الصعوبة في لعبة تخفيف الكوارث الطبيعية ثلاثية الأبعاد .في صنع مستوى النظام ، استخدم الباحثون متغيرات الإدخال وهي النتيجة ، والوقت المتبقى ومتغيرات حياة اللاعب حيث تم الحصول على متغيرات النتيجة وعمر اللاعب من نتائج *BMGD* أداء اللاعب .في مرحلة الاختبار ، سيتم إجراء مقارنة بين الألعاب باستخدام لتحديد دقة *Confussion Matrixion* وبدون طرق . يتم إجراء الاختبار باستخدام مصفوفة ودقة واستدعاء وقياس مستوى النظام باستخدام مستوى بيانات مستوى طريقة *Box Muller of Gaussian Distribution* ونتائج المستوى بدون الطريقة .من مرحلة الاختبار هذه ، وجد يمكن تشغيله تلقائياً حيث يمكن *BMGD* أن نظام مستوى الضباب الذي يستخدم طريقة أن تغيير التعديلات بشكل أكثر ديناميكية مع أفضل النتائج من قياسات أداء طريقة

كما هو موضح في التجربة الأولى والثانية بنفس القيمة وهي دقة  $30\%$  ، دقة  $BMGD$   $37.3\%$  ، استرجاع  $40.3\%$  ، و - قياس  $34.8\%$ .



## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang Masalah

Dunia teknologi pada saat ini berkembang dengan cepat. Perkembangan teknologi mencakup semua bidang kehidupan manusia seperti kesehatan, pangan, industri, dan lain-lain. Begitu juga dengan perkembangan teknologi dalam bidang *game*. *Game-game* didesain sedemikian rupa agar dapat dimainkan oleh semua kalangan, dari anak-anak sampai orang dewasa. Pada saat ini *game* menjadi menarik dengan tampilan yang bagus dan juga mempunyai *story* sehingga *player* tertarik untuk memainkannya. *Game* bukan hanya untuk *refreshing* saja, tetapi juga dapat menjadi kemoterapi oleh seorang anak yang sedang dalam masa perawatan (Ma'ruf Harsono, 2014).

Kebosanan *player* untuk bermain *game* dipengaruhi oleh beberapa faktor, salah satunya yaitu desain rintangan pada skenario *game* (Thuan et al., 2015). Kemampuan *player* sendiri tentu berbeda dengan kemampuan *player* yang lain. Maka dari itu pembuat *game* berupaya untuk selalu menyesuaikan tingkat kesulitan suatu *game* dengan kemampuan *player*.

Dalam Al-Qur'an sendiri dijelaskan bahwa Allah tidak membebani hamba-Nya melebihi dari kemampuan hamba-Nya yang dijelaskan pada Surat Al-Baqarah Ayat 286 :

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا أَنْكَسَبَتْ رَبَّنَا لَا تُؤَاخِذْنَا إِنْ نَسِينَا أَوْ أَخْطَأْنَا رَبَّنَا وَلَا تَحْمِلْنَا إِصْرًا كَمَا حَمَلْنَا عَلَى الَّذِينَ مِنْ قَبْلِنَا رَبَّنَا وَلَا تُحَمِّلْنَا مَا لَا طَاقَةَ لَنَا بِهِ وَأَعْفُ عَنَا وَأَغْفِرْ لَنَا وَأَرْحَمْنَا أَنْتَ مَوْلَانَا فَانصُرْنَا عَلَى الْقَوْمِ الْكُفَّارِينَ

Artinya :

*“Allah tidak membebani seseorang melainkan sesuai dengan kemampuannya. Ia mendapat pahala (dari kebijakan) yang diusahakannya dan ia mendapat siksa (dari kejahanatan) yang dikerjakannya. (Mereka berdoa): "Ya Tuhan kami, janganlah Engkau hukum kami jika kami lupa atau kami tersalah. Ya Tuhan kami, janganlah Engkau bebankan kepada kami beban yang berat sebagaimana Engkau bebankan kepada orang-orang sebelum kami. Ya Tuhan kami, janganlah Engkau pikulkan kepada kami apa yang tak sanggup kami memikulnya. Beri maaflah kami; ampunilah kami; dan rahmatilah kami. Engkaulah Penolong kami, maka tolonglah kami terhadap kaum yang kafir". (QS. Al-Baqarah: 286)*

Al-Hafidh Ibnu Katsir menjelaskan dalam tafsirnya menjelaskan Surat Al-Baqarah Ayat 286 bahwa Allah SWT tidak membebani hamba-Nya dengan amalan-amalan yang berat, sekalipun hamba-Nya masih mampu untuk melakukannya, sebagaimana yang disyari'atkan pada umat-umat terdahulu dengan dibelenggu dan diikat. Sebagaimana Allah SWT mengutus Nabi Muhammad SAW, Nabi pembawa rahmat, yang telah dijadikan sebagai ciri yang menonjol dalam syariatnya, sebagaimana Allah SWT telah mengutusnya dengan membawa agama yang lurus, yang penuh dengan kemudahan dan toleransi. Maksudnya kemudahan dari beban kewajiban-kewajiban, musibah serta bencana dari perkara yang hamba-Nya tidak sanggup menanggungnya. (Amin, 2013)

Dalam menentukan level kesulitan *game* secara lebih dinamis dapat menggunakan beberapa metode dimana salah satu metodenya yaitu *Box-Muller of Gaussian Distribution*. *Box-Muller of Gaussian Distribution* dalam menentukan tingkat kesulitan *game* akan melakukan operasi pengacakan dengan mengacak dua

variabel (Boiroju & Reddy, 2012). Pada penelitian ini akan berkonsentrasi terhadap penentuan level kesulitan *game* dengan menggunakan metode *Box-Muller of Gaussian Distribution*. Objek yang akan digunakan untuk menentukan tingkat kesulitan *game* yaitu kabut dari Gunung Kelud. Objek kabut ini sendiri diambil dari penelitian sebelumnya yang dibuat dalam bentuk simulasi adegan kabut realistik pada *unity* 3D dengan menggunakan metode *Perlin Noise* (Cahyani, 2019). Sedangkan *gameplay* yang akan digunakan dalam penelitian ini mengacu pada penelitian sebelumnya yang dibuat dalam bentuk simulasi untuk menentukan rekomendasi barang yang diperlukan ketika terjadinya bencana, dimana *gameplay* dari *game* yaitu menjalankan misi dengan mengumpulkan barang yang dibutuhkan untuk mengantisipasi terjadinya bencana alam dengan menerapkan metode *Simple Additive Weighting* (Rohman, 2019). Data perhitungan *Box-Muller of Gaussian Distribution* yang dibutuhkan diperoleh dari penelitian terdahulu yang digunakan untuk menentukan tingkat kesulitan pada *game* pembelajaran mitigasi bencana gunung api (Ihsan, 2019).

## 1.2. Pernyataan Masalah

Berdasarkan latar belakang tersebut, maka identifikasi masalah dari penelitian ini yaitu:

1. Bagaimana kinerja metode *Box-Muller of Gaussian Distribution* berbasis fitur kabut untuk *game* 3D mitigasi bencana alam.
2. Bagaimana perbandingan *game* yang menggunakan metode *Box-Muller of Gaussian Distribution* dengan *game* yang tidak menggunakan metode *Box-Muller of Gaussian Distribution*.

### **1.3. Tujuan Penelitian**

Berdasarkan identifikasi masalah di atas, maka tujuan dari penelitian ini adalah:

1. Mengetahui kinerja metode *Box-Muller of Gaussian Distribution* berbasis fitur kabut untuk *game* 3D mitigasi bencana alam dengan genre *adventure*.
2. Mengetahui perbandingan *game* yang menggunakan metode *Box-Muller of Gaussian Distribution* dengan *game* yang tidak menggunakan metode *Box-Muller of Gaussian Distribution*.

### **1.4. Manfaat Penelitian**

Manfaat yang didapatkan dari penelitian ini adalah :

#### **1.4.1. Untuk desainer game**

1. Penentuan skenario *game* yang dipilih membuat beban dari *desainer game* berkurang.
2. Level kesulitan lebih dinamis membuat *game* lebih sulit ditebak oleh *player*.

#### **1.4.2. Untuk player**

1. Tidak mudah bosan dalam bermain *game*.
2. Dapat memainkan *game* sesuai tingkat kemampuannya.

### **1.5. Batasan Masalah**

Berdasarkan identifikasi masalah di atas, adapun batasan masalah dari penelitian ini adalah :

1. *Software* yang digunakan adalah *unity* 3D.
2. *Game* bergenre *adventure* dimana pemain mengumpulkan barang-barang.

3. Penggunaan *NPC Player* untuk melakukan uji coba *game*.
4. Pengukuran kinerja dari metode *Box-Muller of Gaussian Distribution* menggunakan metode *Confusion Matrix* dengan menghitung nilai akurasi, *precision*, *recall*, dan *f-measure*.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Penelitian Terkait

Banyak *paper* dan referensi pada saat ini yang membahas mengenai penentuan level kesulitan secara dinamis atau otomatis. Yang pertama adalah *paper* pembahasan metode *BMGD* untuk penentuan level kesulitan pada *game* Reog Ponorogo (Sukajaya et al., 2012). Pada penelitian tersebut ada tiga tipe level kesulitan *game* yaitu mudah, sedang, dan sulit.

*Paper* berikutnya membahas penentuan level kesulitan terhadap *game* penentuan lorong yang terbuka dengan penyelesaian pertanyaan matematika (Vega & Mardi, 2011). Pada penelitian tersebut ada tiga kategori pertanyaan matematika yaitu *easy*, *medium*, dan *hard*. Tiga kategori pertanyaan ini yang menentukan *player* untuk meminta suatu petunjuk dan bertemu dewa dengan terbukanya lorong. Tiap kategori pertanyaan akan membuka lorong yang berbeda jika poin yang diperoleh sesuai kriteria yang sudah ditentukan.

Berikutnya penggunaan metode *Box-Muller* yang digunakan untuk penentuan perilaku NPC (Majid & Hariadi, 2010). Pada penelitian tersebut melakukan simulasi perang antar NPC dengan menggunakan *engine* 3D *Game Studio*. Kesimpulan dari penelitian tersebut adalah metode *Distribusi Gaussian* dapat menambah durasi peperangan antar NPC.

#### 2.2. *Game Adventure*

Pada awalnya *game adventure* dikembangkan pada 1970-an dan awal 1980-an berbasis teks, dengan penggunaan teks parser untuk menerjemahkan input *player* menjadi perintah. Ketika perkembangan teknologi semakin maju, jenis *game*

*adventure* menjadi populer. Pada awalnya dengan menambahkan perintah teks *player* dengan grafik, menjadi bergerak menuju antar muka. Kemajuan komputer lebih lanjut menyebabkan *game adventure* dengan grafis yang lebih bagus menggunakan tiga dimensi *real-time* atau *pre-rendered* maupun video *full-motion* yang diambil dari perspektif orang pertama atau ketiga. Popularitas *game* genre *adventure* memuncak selama akhir 1980-an hingga pertengahan 1990-an, genre tersebut banyak menganggapnya sebagai salah satu genre yang paling maju secara teknis, tetapi telah menjadi genre *niche* di awal 2000-an karena popularitasnya (Rollings & Ernest, 2003).

### 2.3. *Non-Player Character* (NPC)

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 berkisar dari yang biasa, yang fantastis, kecil sampai besar, hingga yang tidak manusiawi. NPC adalah entitas yang terlihat seperti pemain dan yang akan berpikir mereka sama jika diberi AI. 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 (Warpefelt, 2016).

### 2.4. Kabut (*fog*)

Kabut adalah uap air yang berada dekat permukaan tanah berkondensasi dan menjadi mirip awan. Hal ini biasanya terbentuk karena hawa dingin membuat uap air berkondensasi dan kadar kelembaban mendekati 100% (Wikipedia). Kabut mirip dengan awan, tetapi awan tidak menyentuh permukaan bumi,

sedangkan kabut menyentuh permukaan bumi. Kabut terbentuk dari uap air yang berasal dari tanah yang lembab, tanaman-tanaman, sungai, danau, dan lautan.

Pada penelitian sebelumnya dilakukan penerapan metode *Perlin noise* yang akan dilaksanakan setelah didapat perhitungan *density* (kedalaman), untuk menghasilkan simulasi lebih nyata. Penelitian ini akan dihitung secara matematis menggunakan Matlab sehingga menghasilkan Citra sintesis kabut heterogen 2D, yang selanjutnya akan dilanjutkan dengan menggunakan unity 3D dengan menghasilkan kabut sintesis heterogen 3D.

**Tabel 2.1** Densitas 12 Citra Uji

Sumber: (Pradanti, 2018)

Waktu	Densitas
06.00 WIB	3,554
07.00 WIB	3,373
08.00 WIB	2,41
09.00 WIB	3,424
10.00 WIB	4,108
11.00 WIB	3,257
12.00 WIB	2,761
13.00 WIB	4,209
14.00 WIB	3,093
15.00 WIB	8,181
16.00 WIB	5,123
17.00 WIB	4,154

Dalam penelitian tersebut digunakan *Estimasi Transmission Map* dimana perkiraan transmisi citra kabut sangat diperlukan untuk menciptakan *scene* kabut. *Transmission map* berguna untuk menunjukkan tingkatan dari transparansi citra, kemudian di dalam penelitiannya menjelaskan perkiraan yang akan digunakan dengan urutan seperti pada tabel 2.2.

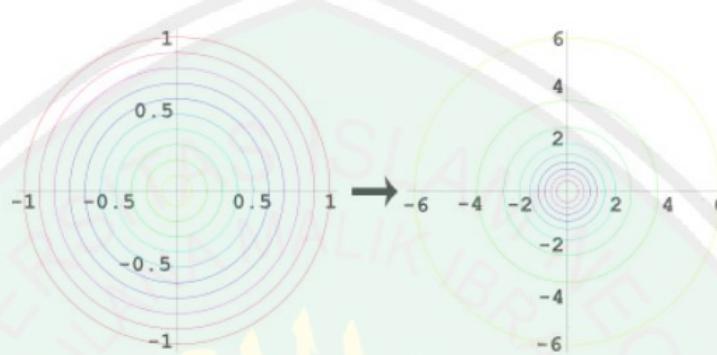
**Tabel 2.2** Fitur Kabut dalam *Game*

Kabut	Ketebalan
1	Extremely clear
2	Sangat jernih
3	Jernih
4	Light haze
5	Haze
6	Thin fog
7	Light fog
8	Moderate fog
9	Thick fog
10	Dense fog

## 2.5. *Box-Muller*

*Box-Muller* adalah metode pengambilan sampel bilangan acak untuk menghasilkan pasangan angka acak yang independen, standar, terdistribusi normal (nol ekspektasi, varian unit), diberikan sumber yang seragam nomor acak terdistribusi. Metode ini sebenarnya pertama kali disebutkan secara eksplisit oleh Raymond E. A. C. Paley dan Norbert Wiener pada tahun 1934. Transformasi Box-Muller umumnya diekspresikan dalam dua bentuk. Bentuk dasar seperti yang

diberikan oleh Box dan Muller mengambil dua sampel dari distribusi yang sama pada interval  $[0, 1]$  dan memetakannya ke dua sampel standar yang didistribusikan secara normal. Bentuk kutub mengambil dua sampel dari interval yang berbeda,  $[-1, +1]$ , dan memetakannya ke dua sampel yang terdistribusi normal tanpa menggunakan fungsi sinus atau kosinus (Thomas et al., 2007).

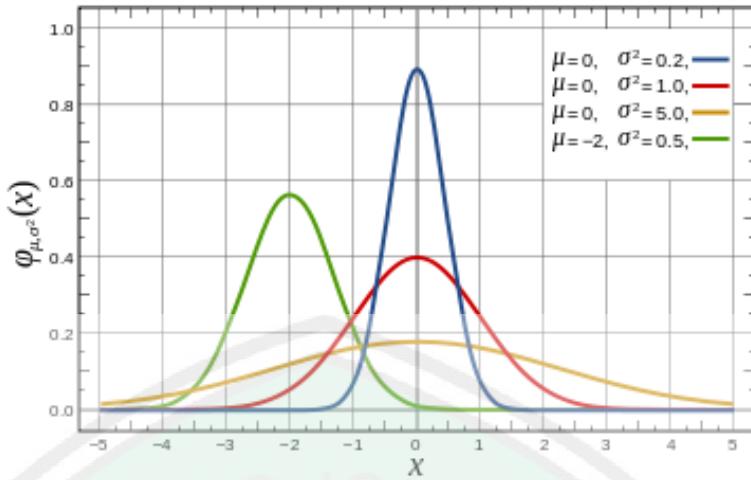


**Gambar 2.1** Sampel Terdistribusi Normal

Sumber: (Majid & Hariadi, 2010)

## 2.6. Gaussian Distribution

Distribusi Gaussian adalah distribusi probabilitas yang paling banyak digunakan dalam berbagai analisis statistika. Distribusi normal banyak digunakan dalam berbagai bidang statistika, misalnya distribusi sampling rata-rata akan mendekati normal, meski distribusi populasi yang diambil tidak berdistribusi normal. Distribusi normal juga banyak digunakan dalam berbagai distribusi dalam statistika, dan kebanyakan pengujian hipotesis mengasumsikan normalitas suatu data. (R et al., 2005).



**Gambar 2.2 Kurva Distribusi Gaussian**

Sumber: (Sukajaya et al., 2012)

### 2.7. Box-Muller of Gaussian Distribution

*Box-Muller of Gaussian Distribution (BMGD)* adalah sebuah metode untuk menentukan nilai *mean* dari Distribusi Gauss dengan menggunakan metode *Box-Muller*. *BMGD* digunakan pada saat awal permainan untuk penyesuaian tingkat kesulitan pada level selanjutnya.

*Pseudocode* algoritma *Box-Muller* (Sukajaya et al., 2012) :

- Hasilkan nomor acak seragam  $u, v$  dalam kisaran  $[-1, 1]$
  - Hitung  $s = u^2 + v^2$
  - Looping* langkah kedua hingga  $s < 1$
  - Temukan angka acak normal  $z_0$  dan  $z_1$
- (2.1)

$$z_0 = u \sqrt{\frac{-2 \ln s}{s}} \quad (2.2)$$

$$z_1 = v \sqrt{\frac{-2 \ln s}{s}} \quad (2.3)$$

Dimana  $z_0$  dan  $z_1$  adalah variabel acak bebas yang terdistribusi normal standar.

Dari pseudocode  $z_0$  dan  $z_1$  akan dimasukkan kedalam rumus berikut:

$$rr = \text{mean} + \text{sd} * z0 \text{ atau } z1 \quad (2.4)$$

Berikutnya *pseudocode* di atas akan diproses ke dalam pembuatan *game* untuk membangkitkan angka *random* yang terdistribusi normal pada pemilihan jenis skenario (Vega & Mardi, 2011).

## 2.8. Rata-rata

*Mean* atau rata-rata diperoleh dengan menjumlahkan skor dibagi dengan banyaknya subjek. Secara sederhana rumusnya adalah (Sudjana, 2013):

$$X = \frac{\sum X}{N} \quad (2.5)$$

Keterangan:

$X$  = rata-rata

$\sum X$  = jumlah seluruh skor

$N$  = banyaknya subjek

## 2.9. Modus

Modus adalah nilai yang sering muncul atau nilai yang frekuensinya banyak dalam distribusi data (Sudjana, 2013).

## 2.10. Matriks Konfusi

Matriks konfusi yaitu suatu matriks yang mengindikasikan tingkat akurasi citra yang telah terkласifikasi terhadap data referensi. Metode ini digunakan untuk menghitung nilai akurasi atau evaluasi model klasifikasi (Utami, 2016).

**Tabel 2.3** Klasifikasi dan Prediksi

Sumber: Utami (2016)

Classification	Predicted Class	
	Class = Yes	Class = No
Class = Yes	a (true positive-TP)	b (false negative-FN)

Class = No	c (false positive-FP)	d (true negative-TN)
------------	-----------------------	----------------------

Dalam Klasifikasi dan Prediksi akan dihitung menggunakan rumus akurasi, *precision*, *recall*, dan *f-measure* sebagai berikut:

#### 2.10.1. Akurasi

$$\text{Akurasi} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (2.6)$$

Perhitungan akurasi dilakukan dengan cara membagi data yang diklasifikasi secara benar dengan total data testing yang diuji.

#### 2.10.2. Precision

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (2.7)$$

Perhitungan *Precision* dilakukan dengan cara membagi jumlah data benar yang bernilai positif (*TP*) dibagi dengan jumlah data benar yang bernilai positif (*TP*) dan data salah yang bernilai positif (*FN*).

#### 2.10.3. Recall

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (2.8)$$

Perhitungan *Recall* dilakukan dengan cara membagi data benar yang bernilai positif (*TP*) dengan hasil penjumlahan dari data benar yang bernilai positif (*TP*) dan data salah yang bernilai negatif (*FN*).

#### 2.10.4. F-Measure

$$\text{F-Measure} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (2.9)$$

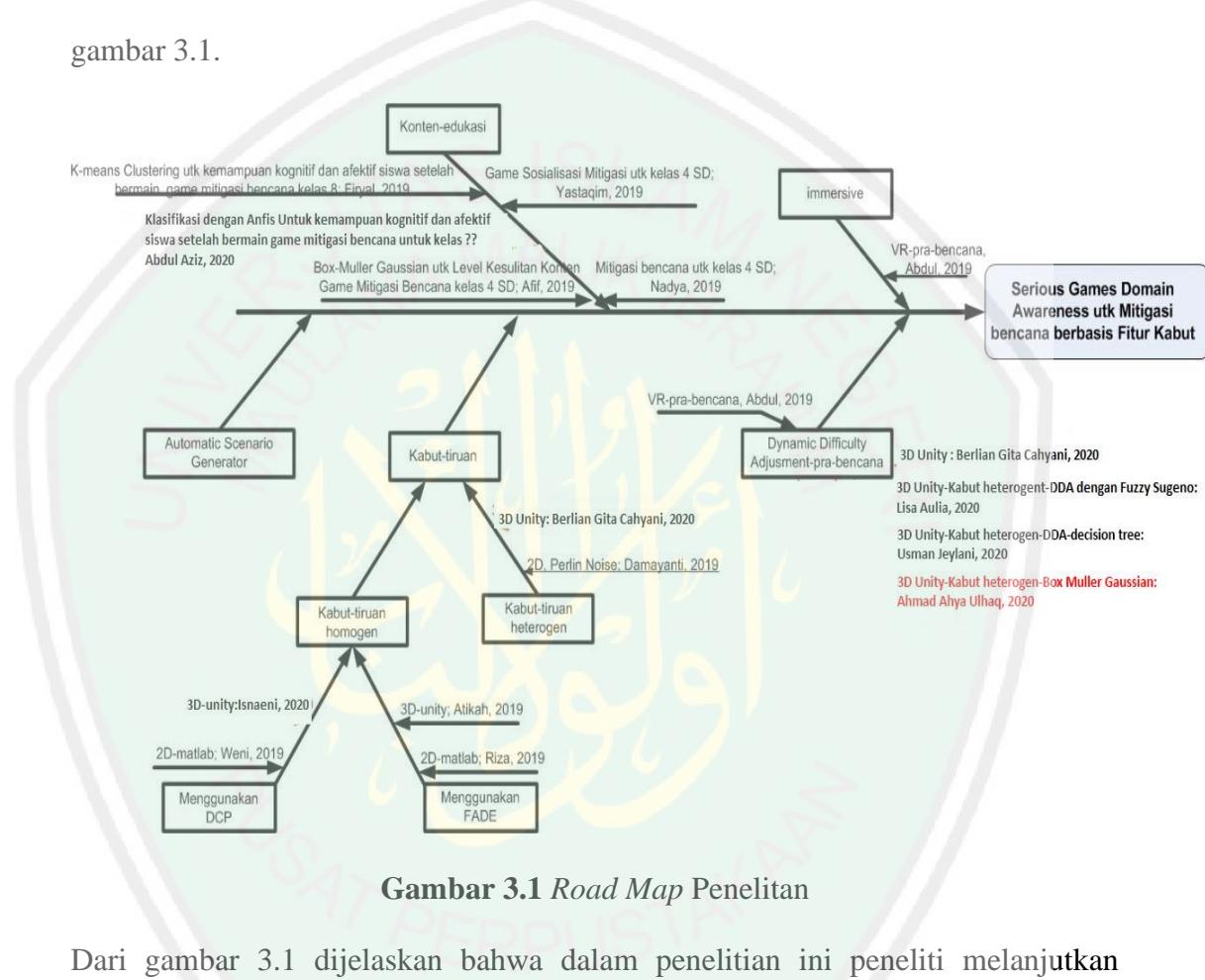
*F-Measure* didapat dari perhitungan pembagian hasil dari perkalian *Precision* dan *Recall* dengan hasil penjumlahan *Precision* dan *Recall*, kemudian dikalikan dua.

## BAB III

### METODE PENELITIAN

#### 3.1. Penelitian Sebelumnya

Pada penelitian ini peneliti melanjutkan hasil penelitian yang telah selesai dilakukan dan sedang dikerjakan sebelumnya. Secara garis besar ditunjukkan pada gambar 3.1.



**Gambar 3.1 Road Map Penelitian**

Dari gambar 3.1 dijelaskan bahwa dalam penelitian ini peneliti melanjutkan penelitian sebelumnya. Peneliti mengumpulkan data ketebalan kabut dan konten *game* dari penelitian sebelumnya untuk diproses menjadi salah satu parameter dalam penelitian ini.

#### 3.2. Desain Sistem Penelitian Sebelumnya

Pada penelitian ini, pembuatan *game* akan dibuat dengan *gameplay* yang mengacu pada penelitian sebelumnya. Pada penelitiannya, peneliti membuat *game*

simulasi untuk menentukan rekomendasi barang yang diperlukan ketika terjadinya bencana, dimana *gameplay* dari *game* yaitu menjalankan misi dengan mengumpulkan barang yang dibutuhkan sebelum terjadinya bencana alam dengan menerapkan metode *Simple Additive Weighting*.

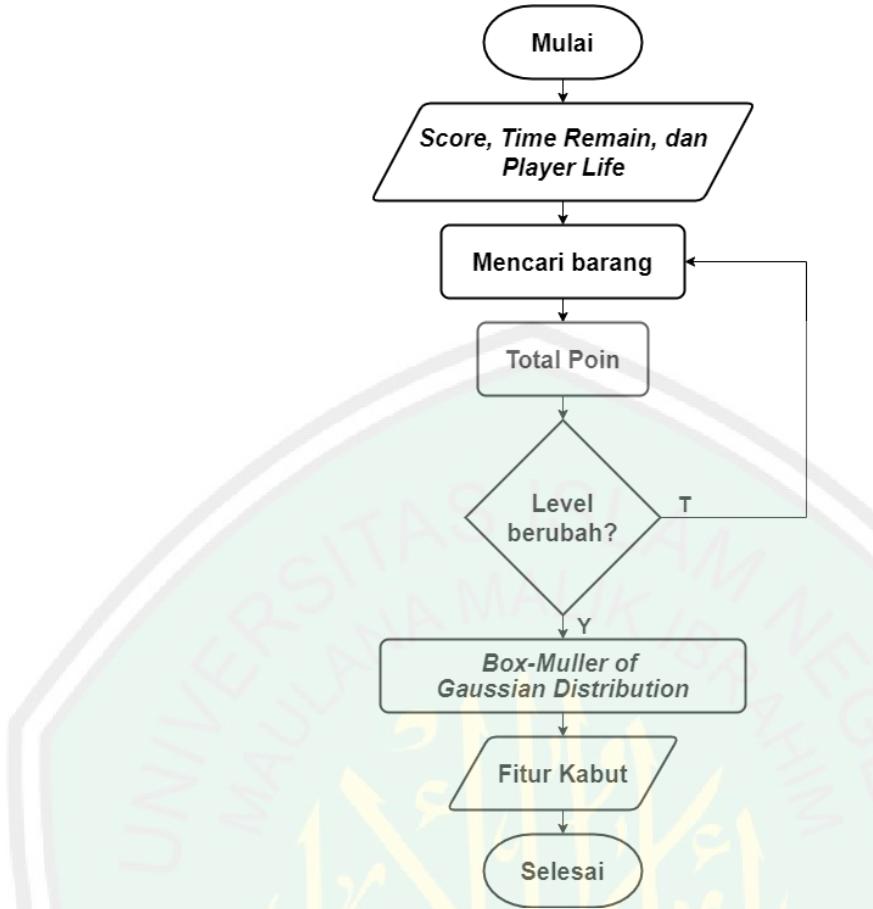


**Gambar 3.2** Desain Sistem *Game* yang digunakan pada Penelitian Sebelumnya  
Sumber: (Rahman, 2019)

Dari gambar 3.2 selanjutnya akan berkonsentrasi terhadap penentuan level kesulitan *game* dengan metode *BMGD* dimana penggunaan kabut yang dihasilkan dari penelitian sebelumnya digunakan sebagai objek untuk menentukan tingkat kesulitan di dalam *game*.

### 3.3. Desain Sistem

Dalam penerapan metode *BMGD* pada *game* akan digunakan fitur kabut dan *gameplay* yang telah dibuat pada penelitian sebelumnya. Peneliti menambahkan variabel *score*, *time remain*, dan *player life* sebagai *inputan* di dalam *game* untuk menentukan nilai variabel total poin yang digunakan sebagai parameter penentuan setiap level. Jika level berubah maka akan memproses *BMGD*. Hasil perhitungan *BMGD* ini selanjutnya digunakan untuk menentukan fitur kabut yang akan ditampilkan.



**Gambar 3.3** Desain Sistem Box-Muller of Gaussian Distribution pada Game

#### 3.4. Skenario Game

Game ini mempunyai *story* yaitu *player* yang akan bersiap menghadapi bencana alam. *Plyaer* harus mencari barang yang diperlukan dan menyelamatkan diri dari bencana alam. Untuk keperluan penelitian, proses berjalanannya *game* tidak dilakukan oleh pemain (*player*), maka fungsi dari *player* dalam *game* akan diganti dengan NPC. NPC harus mencari barang-barang yang perlu dibawa dengan *input* variabel yang telah ditentukan sebelumnya untuk menghasilkan *output* berupa tingkat kesulitan kabut atau level kabut.

Adapun tahapan skenario *game* sebagai berikut:

- Pertama, NPC bergerak mencari barang-barang yang diperlukan dengan *time remain*, *score* dan *player life*.

- Selanjutnya NPC mengumpulkan barang-barang yang diperlukan dalam menghadapi bencana alam.
- NPC menghindari rintangan yang ada di dalam game, dimana dapat mengurangi *player life* yang dapat menyebabkan *game lose*.
- Selanjutnya NPC akan mendapatkan poin *score* sesuai apa yang didapatkan di dalam game.
- Fitur kabut di dalam game berubah-ubah seiring perubahan level dengan menerapkan metode *BMGD*.
- Setelah mendapatkan barang-barang yang dibutuhkan maka permainan selesai dengan *score* mencapai target.
- Permainan dianggap *Win* ketika *score* telah mencapai target dan *time remain* tidak habis
- Permainan dianggap *Lose* ketika *time remain* untuk mengumpulkan barang telah habis dan *player life* habis.

Untuk melihat metode *BMGD* telah berjalan atau tidak maka digunakan *NPC Player*. *NPC Player* akan dibagi menjadi 2 tingkatan, yaitu *NPC Player amateur* dan *advanced*.



**Gambar 3.4** Skenario Game

### 3.5. Perhitungan Manual Box-Muller of Gaussian Distribution

Algoritma perhitungan *BMGD* ditunjukkan pada gambar 3.5.



**Gambar 3.5** Algoritma perhitungan *Box-Muller of Gaussian Distribution*

Data perhitungan *BMGD* yang dibutuhkan diperoleh dari penelitian terdahulu yang digunakan untuk menentukan level kesulitan pada *game* pembelajaran mitigasi bencana gunung api. Tahap pertama perhitungan manualnya yaitu dengan membangkitkan nilai acak *u* dan *v* pada interval [-1,1]. Pada tahap ini akan dilakukan 10 kali percobaan dengan  $s < 1$  pertama, angka yang keluar dari perintah.

$$u = (2.0 * \text{rand}(1.0)) - 1.0;$$

$$v = (2.0 * \text{rand}(1.0)) - 1.0;$$

**Tabel 3.1** Nilai Bilangan Acak U dan V

Sumber: (Ihsan, 2019)

No	U	V
1	0,081003	0,161465

2	0,091804	0,134133
3	0,071796	0,268847
4	0,040067	0,74408
5	0,944833	0,221361
6	0,600565	0,436093
7	0,173561	0,194894
8	0,189121	0,929892
9	0,490544	0,796324
10	0,184799	0,227785

Setelah bilangan acak u dan v sudah ditentukan maka selanjutnya mencari nilai s dengan rumus seperti pada persamaan 2.1. Nilai s harus  $< 1$ .

**Tabel 3.2** Nilai S

No	S
1	0,032632
2	0,02642
3	0,077434
4	0,555261
5	0,94171
6	0,550856
7	0,068107
8	0,900465
9	0,874766
10	0,086036

Setelah nilai s sudah ditentukan maka selanjutnya mencari nilai z0 dan z1 dengan rumus seperti pada persamaan 2.2 dan persamaan 2.3.

**Tabel 3.3** Nilai z0 dan z1

No	z0	z1

1	1,773132	2,541094
2	3,003401	3,46606
3	3,384612	4,440214
4	4,038437	4,713813
5	5,222376	5,052099
6	6,582337	6,422857
7	8,015945	8,140821
8	8,060143	8,295718
9	9,178799	9,290254
10	10,91963	11,13354

Setelah nilai  $z_0$  dan  $z_1$  sudah ditentukan maka tahap selanjutnya adalah menghitung nilai rr dengan rumus seperti pada persamaan 2.4. Pada penelitian ini akan menggunakan tingkat kesulitan 10 level. Maka dibutuhkan  $\mu$  atau *mean* sebanyak 10 untuk menghitung rr. Setelah nilai rr dari setiap mean sudah ditentukan, selanjutnya mencari *between*. *Between* adalah nilai terkecil dan terbesar dari rr setiap *mean*.

**Tabel 3.4** Nilai rr pada  $\mu_1$

$\mu_1$	rr
1	-0,741604717
2	1,596513342
3	1,333949698
4	0,848769644
5	1,091707171
6	1,778290917
7	2,452767867
8	1,482281473
9	2,154153122
10	-0,056244762

*Between* dari  $\mu_1$  adalah -0,741604717 dan 2,452767867.

**Tabel 3.5** Nilai rr pada  $\mu_2$ 

$\mu_2$	rr
1	2,642220375
2	0,633303201
3	1,247825234
4	2,819590668
5	0,838288404
6	2,45564951
7	0,957333401
8	2,310659297
9	2,436749887
10	2,852359488

*Between* dari  $\mu_2$  adalah 0,633303201 dan 2,852359488.

**Tabel 3.6** Nilai rr pada  $\mu_3$ 

$\mu_3$	rr
1	3,057392008
2	3,145706861
3	2,343450128
4	3,676983566
5	3,708575179
6	3,581689224
7	2,239028322
8	5,334629564
9	2,53609027
10	5,200663737

*Between* dari  $\mu_3$  adalah 2,239028322 dan 5,334629564.

**Tabel 3.7** Nilai rr pada  $\mu_4$ 

$\mu_4$	rr

1	4,450504027
2	4,811593616
3	2,023193843
4	4,166551388
5	6,514151758
6	3,500375579
7	2,059684849
8	2,855497499
9	2,733782675
10	4,83262105

Between dari  $\mu_4$  adalah 2,023193843 dan 6,514151758.

**Tabel 3.8** Nilai rr pada  $\mu_5$

$\mu_5$	rr
1	4,90084955
2	4,882245745
3	4,679430651
4	5,010658022
5	4,20513346
6	5,216380074
7	4,230341823
8	4,358991485
9	4,859010801
10	4,225568759

Between dari  $\mu_5$  adalah 4,20513346 dan 5,216380074.

**Tabel 3.9** Nilai rr pada  $\mu_6$

$\mu_6$	rr
1	7,034950537
2	6,160405601

3	7,020710615
4	3,811897579
5	5,831977207
6	5,662341397
7	5,309070442
8	5,89280164
9	3,55318309
10	5,711912727

Between dari  $\mu_6$  adalah 3,55318309 dan 7,034950537.

**Tabel 3.10** Nilai rr pada  $\mu_7$

$\mu_7$	rr
1	8,325148254
2	5,553283628
3	7,113192785
4	5,762317547
5	9,050208728
6	7,420029974
7	7,597932596
8	7,564042494
9	5,545570628
10	6,424871954

Between dari  $\mu_7$  adalah 5,545570628 dan 9,050208728.

**Tabel 3.11** Nilai rr pada  $\mu_8$

$\mu_8$	rr
1	7,493848819
2	10,35065159
3	8,554325282
4	9,511358664

5	9,359853638
6	7,823203525
7	9,042145324
8	9,787498373
9	7,02593393
10	7,832145209

Between dari  $\mu_8$  adalah 7,02593393 dan 10,35065159.

**Tabel 3.12** Nilai rr pada  $\mu_9$

$\mu_9$	rr
1	9,431431286
2	8,497121772
3	7,517033308
4	8,775986285
5	8,713991456
6	8,848125471
7	7,480491113
8	9,606482987
9	9,76094599
10	8,723196188

Between dari  $\mu_9$  adalah 7,480491113 dan 9,76094599.

**Tabel 3.13** Nilai rr pada  $\mu_{10}$

$\mu_{10}$	rr
1	8,836324404
2	9,629473425
3	10,2882695
4	9,610449981
5	8,429914025
6	10,53248004

7	10,12352845
8	9,78983443
9	11,62068526
10	9,547920337

*Between* dari  $\mu_{10}$  adalah 8,836324404 dan 11,62068526.

Setelah nilai rr dan *between* sudah ditentukan maka selanjutnya memasukkan fitur kabut ke dalam setiap *between*.

**Tabel 3.14** Fitur Kabut dalam *between*

Fitur Kabut	<i>mean</i>	<i>between</i>	
		-0,7	0,5
Kabut 1	1	-0,7	0,5
Kabut 2	2	0,6	2,1
Kabut 3	3	2,2	2,6
Kabut 4	4	2,7	4,1
Kabut 5	5	4,2	5,2
Kabut 6	6	5,3	5,4
Kabut 7	7	5,5	6,9
Kabut 8	8	7	7,3
Kabut 9	9	7,4	8,7
Kabut 10	10	8,8	11

### 3.6. Perancangan Pengujian *Box-Muller of Gaussian Distribution*

Pada pengujian metode ini dilakukan dengan menentukan nilai awal *inputan BMGD* dengan variabel *score*, *time remain*, dan *NPC life*. Dalam penyelesaian *game* atau berakhirnya *game*, ditentukan *score* target dengan nilai 60, berikut penentuan nilai setiap variabel yang mempengaruhi berubahnya level kabut:

**Tabel 3.15** Penentuan Nilai Setiap Variabel

Penentuan Level	Score	Time Remain	NPC Life	Fitur Kabut

Level 1	0	300	100	Kabut 1
Level 2	5	280	90	Kabut 2
Level 3	10	260	80	Kabut 3
Level 4	15	240	70	Kabut 4
Level 5	20	220	60	Kabut 5
Level 6	25	200	50	Kabut 6
Level 7	30	180	40	Kabut 7
Level 8	35	160	30	Kabut 8
Level 9	40	140	20	Kabut 9
Level 10	45	120	15	Kabut 10

Berikut adalah detail dari penentuan fitur kabut yang akan ditampilkan sesuai dengan level yang telah ditentukan seperti pada tabel 3.15:

- Level 1 : Kabut 1
- Level 2 : Kabut 2
- Level 3 : Kabut 3
- Level 4 : Kabut 4
- Level 5 : Kabut 5
- Level 6 : Kabut 6
- Level 7 : Kabut 7
- Level 8 : Kabut 8
- Level 9 : Kabut 9
- Level 10 : Kabut 10

**Tabel 3.16** Penentuan Level Kabut dari Total Poin

Penentuan Level	Score	Time Remain	NPC Life	Fitur Kabut	Total Poin
Level 1	0	300	100	Kabut 1	133,3333333

Level 2	5	280	90	Kabut 2	125
Level 3	10	260	80	Kabut 3	116,6666667
Level 4	15	240	70	Kabut 4	108,3333333
Level 5	20	220	60	Kabut 5	100
Level 6	25	200	50	Kabut 6	91,66666667
Level 7	30	180	40	Kabut 7	83,33333333
Level 8	35	160	30	Kabut 8	75
Level 9	40	140	20	Kabut 9	66,66666667
Level 10	45	120	15	Kabut 10	60

Total poin merupakan variabel yang akan digunakan sebagai parameter penentuan setiap level. Total poin merupakan rata-rata yang diambil dari variabel *score*, *time remain*, dan *NPC life* dengan rumus seperti pada persamaan 2.5.

Berikut ini merupakan detail dari desain level yang akan digunakan:

- Level 1 : if  $125 < \text{Total Poin} \leq 133$
- Level 2 : if  $116,6666667 < \text{Total Poin} \leq 125$
- Level 3 : if  $108,3333333 < \text{Total Poin} \leq 116,6666667$
- Level 4 : if  $100 < \text{Total Poin} \leq 108,3333333$
- Level 5 : if  $91,66666667 < \text{Total Poin} \leq 100$
- Level 6 : if  $83,33333333 < \text{Total Poin} \leq 91,66666667$
- Level 7 : if  $75 < \text{Total Poin} \leq 83,33333333$
- Level 8 : if  $66,66666667 < \text{Total Poin} \leq 75$
- Level 9 : if  $60 < \text{Total Poin} \leq 66,66666667$
- Level 10 : if  $\text{Total Poin} \leq 60$

Pada setiap perubahan level akan memproses perhitungan *Box-Muller of Gaussian Distribution* dengan *output* nilai rr. Nilai rr inilah yang akan digunakan untuk menentukan fitur kabut pada setiap level. Sebagai contoh, *NPC Player* telah memperoleh *score* 30 dengan *time remain* masih 220 detik dan *NPC life* 80. Maka total poin yang diperoleh adalah 110 yang dapat diklasifikasikan menjadi Level 3 dengan interval  $108,333333 < \text{Total Poin} \leq 116,666667$ . Selanjutnya akan memproses perhitungan *Box-Muller of Gaussian Distribution*. Jika nilai rr yang keluar adalah 6,6 maka dapat diklasifikasikan menjadi *between* dengan interval 5,5 sampai 6,9. Jadi kabut yang ditampilkan pada level tersebut adalah kabut 7.

Dari perancangan pengujian metode *BMGD* tersebut akan dibandingkan dengan *game* yang tidak menggunakan metode *BMGD*. Objek yang digunakan untuk membandingkan *game* yang menggunakan metode *BMGD* dengan *game* yang tidak menggunakan metode *BMGD* adalah perbandingan level kabut. Level kabut yang ditampilkan pada suatu level yang menggunakan metode *BMGD* akan dibandingkan dengan level kabut yang ditampilkan pada kondisi nilai variabel yang sama tetapi tidak menggunakan metode *BMGD*. Pengujian ini diharapkan dapat menghasilkan sistem level otomatis berbasis fitur kabut yang berbeda untuk meningkatkan daya tarik bagi *player*.

Dari perancangan pengujian metode *BMGD* tersebut akan dibandingkan pada *game* yang tanpa menggunakan metode *BMGD*, dimana pengaruh perubahan level hanya didasari pada variabel *score* dengan ketentuan sebagai berikut:

**Tabel 3.17** Ketentuan Tanpa *BMGD*

NO	Variabel Masukan	
	Tingkatan Level Kabut	Score
1	Level 1	0

2	Level 2	5
3	Level 3	10
4	Level 4	15
5	Level 5	20
6	Level 6	25
7	Level 7	30
8	Level 8	35
9	Level 9	40
10	Level 10	45

Setelah itu, dilakukan pengujian kembali menggunakan *Confusion Matrix* dalam klasifikasi dan prediksi akan dihitung menggunakan rumus akurasi, *precision*, *recall*, dan *f-measure* seperti pada persamaan 2.6, persamaan 2.7, persamaan 2.8, dan persamaan 2.9. Tujuan pengujian menggunakan *Confusion Matrix* adalah untuk membandingkan hasil dari sistem yang menggunakan metode *BMGD* dan sistem yang tidak menggunakan metode *BMGD*. Dari pengujian *Confusion Matrix* tersebut dapat dilihat hasilnya untuk menentukan seberapa berhasil pengujian perbandingan antara *game* yang menggunakan metode *BMGD* dan *game* yang tidak menggunakan metode *BMGD*.

Berikut contoh perhitungan *Confusion Matrix* :

**Tabel 3.18** Contoh Hasil Level Kabut *BMGD* dan Tanpa *BMGD*

Hasil Level Kabut	
<i>BMGD</i>	Tanpa <i>BMGD</i>
1	1
2	2
3	3
4	3

5	3
5	4
5	4
5	5
5	5
6	6
6	6
5	6
7	7
5	7
6	7
8	8
9	9
10	10
8	10
6	10

Dari tabel 3.18 dijelaskan perbedaan level kabut antara *game* yang menggunakan metode *BMGD* dan *game* yang tidak menggunakan metode *BMGD*. Untuk menganalisa perbandingannya maka data dari seluruh percobaan akan dihitung nilai modus.

**Tabel 3.19** Contoh *Confusion Matrix* Hasil Level

- Perhitungan Akurasi

Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai benar.

Diketahui jumlah data uji adalah 20, kemudian data jumlah *True Positif* (*TP*) pada tabel di atas dengan kolom berwarna putih yaitu 8 data dan berjumlah 10.

Formula Akurasi :

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

$$\text{Akurasi} = 10 / 20$$

$$\text{Akurasi} = 0,5$$

- Perhitungan *Precision*

Presisi adalah data yang diambil berdasarkan informasi yang kurang atau salah dan tidak tepat.

Diketahui pada setiap kolom terdapat 8 data *True Positif* (*TP*) dan berjumlah 10, kemudian *False Positif* (*FP*) sebagai berikut:

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

Dengan jumlah *FP* adalah 10, setelah menemukan masing-masing nilai *FP* dari setiap kelas, selanjutnya akan dihitung presisi dari masing-masing kelas

Formula *Precision*:

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

$$\text{P}(1) = 1 / (1+0) = 1$$

$$\text{P}(2) = 1 / (1+0) = 1$$

$$\text{P}(3) = 1 / (1+2) = 0,3333333333$$

$$P(4) = 0/(0+2) = 0$$

$$P(5) = 2/(2+0) = 1$$

$$P(6) = 2/(2+2) = 0,5$$

$$P(7) = 0/(0+2) = 0$$

$$P(8) = 1/(1+0) = 1$$

$$P(9) = 1/(1+0) = 1$$

$$P(10) = 1/(1+2) = 0,3333333333$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

All Precision = P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9)  
+ P(10) / Jumlah kelas

$$Precision = (1 + 1 + 0,3333333333 + 0 + 1 + 0,5 + 0 + 1 + 1 + 0,3333333333) / 10$$

*Precision = 0,616666667*

- Perhitungan *Recall*

*Recall* Adalah data yang tidak mampu diprediksi dengan benar.

Diketahui pada setiap kolom terdapat 8 data *True Positif* (*TP*) dan berjumlah

10, kemudian False Negatif (FN) dapat dilihat pada tabel sebagai berikut:

**Tabel 3.20** *Confusion Matrix (FN)*

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

Setelah menemukan masing-masing nilai  $FN$ , proses perhitungan *Recall* dilakukan pada masing-masing kelasnya.

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

$$R(1) = 1 / (1+0) = 1$$

$$R(2) = 1 / (1+0) = 1$$

$$R(3) = 1 / (1+0) = 1$$

$$R(4) = 0 / (0+1) = 0$$

$$R(5) = 2 / (2+5) = 0,2857142857$$

$$R(6) = 2 / (2+2) = 0,5$$

$$R(7) = 0 / (0+1) = 0$$

$$R(8) = 1 / (1+1) = 0,5$$

$$R(9) = 1 / (1+0) = 1$$

$$R(10) = 1 / (1+0) = 1$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

$$Recall = (1 + 1 + 1 + 0 + 0,2857142857 + 0,5 + 0 + 0,5 + 1 + 1) / 10$$

$$Recall = 0,6285714286$$

- Perhitungan *F-measure*

*F-Measure* adalah perhitungan pembagian hasil dari perkalian *Precision* dan *Recall* dengan hasil penjumlahan *Precision* dan *Recall*, kemudian dikalikan dua.

Formula *F-measure*

$$F\text{-measure} = (2 * \text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$

$$F\text{-measure} = (2 * 0,616666667 * 0,6285714286) / (0,616666667 + 0,6285714286) = 0,7752380953 / 1,2452380953$$

***F-Measure = 0,6225621415***



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Implementasi

Pada bab ini akan menjelaskan tentang perencanaan serta pengujian hasil implementasi metode *Box-Muller of Gaussian Distribution* yang telah diajukan terhadap *game* untuk mengetahui level kesulitan kabut dengan menggunakan *NPC Player* sebagai *player*.

#### 4.2. Implementasi Antarmuka *Game*

Berikut merupakan tampilan antarmuka untuk *player*:

##### 4.2.1. Tampilan Awal *Game*



Gambar 4.1 Tampilan *Scenes* Awal *Game*

Pada *scene* awal terdapat menu utama untuk memulai bermain dengan pembuatan menggunakan *canvas* yang terdiri dari *Button* Mulai BMGD, *Button* Mulai tanpa BMGD dan *Button* Keluar.

#### 4.2.2. Tampilan Tahap 1



**Gambar 4.2** Tampilan Tahap 1

Pada gambar 4.2, *player* memasuki arena *game* dengan tujuan mencari barang-barang yang ada di dalam *game*. Pada tahap ini peneliti menggunakan *NPC Player* untuk melakukan pengujian sistem level terhadap kabut. Pada tampilan awal memasuki arena terdapat *Score*, *Time Remain*, *Score Target*, *Difficulty Level Npc's Life*, dan *Button Pause*.

#### 4.2.3. Tampilan Tahap 2



**Gambar 4.3** Tampilan Tahap 2

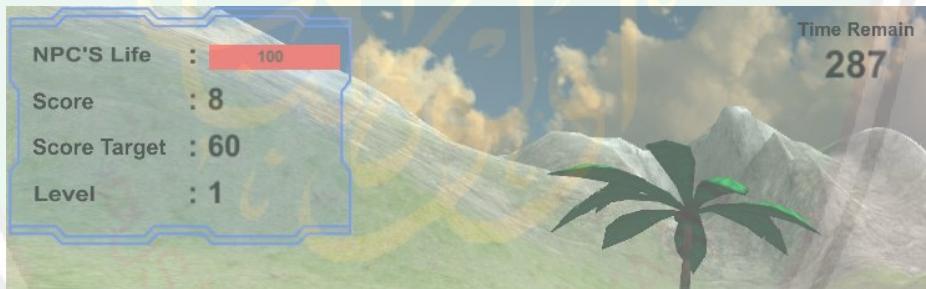
Pada gambar 4.3, setelah memasuki arena *game*, *player* bergerak mencari dan mengumpulkan barang-barang di dalam *game* dengan batas *Time Remain* 300

detik, *Score* target 60 *point* dan *Player Life* dengan jumlah 100. Pada *game* terdapat rintangan hewan-hewan yang harus dihindari agar tidak mengurangi *player life* sehingga mengakibatkan kekalahan *Game*.



**Gambar 4.4** Rintangan di dalam *Game*

Setelah memasuki arena *game*, *player* bergerak mencari dan mengumpulkan barang-barang di dalam *game* dan menghindari rintangan seperti pada gambar 4.4.



**Gambar 4.5** Panel Variabel di dalam *Game*

Pada gambar 4.5, merupakan panel yang berisi variabel di dalam *game* yang meliputi *Score*, *Player Life*, dan *Time Remain*.

#### 4.2.4. Tampilan Tahap 3

Setelah *player* mendapatkan barang-barang, *Score* pada *game* terisi sesuai jumlah barang di dalam *game*, untuk *point* barang. Peneliti menentukan setiap barang dengan 2 *point* untuk makanan dan 4 *point* untuk barang-barang yang

dibutuhkan dalam menghadapi bencana alam, kemudian jumlah keseluruhan barang yaitu 25.



**Gambar 4.6** Panel *Score* terisi di dalam *Game*

Pada gambar 4.6, merupakan panel yang berisi variabel *score* di dalam *game* yang sudah mendapatkan *score* 8.

#### 4.2.5. Tampilan Tahap 4

Pada saat nilai variabel *BMGD* terisi, kemudian kondisi *BMGD* terpenuhi maka proses sistem level kabut berjalan otomatis sesuai kemampuan *player* di dalam *game*.



#### **Gambar 4.7 Perubahan Level Kabut di dalam Game**

Pada gambar 4.7 perubahan level kabut dilakukan dengan mengolah data variabel didalam *game* yaitu *Time Remain*, *Score*, dan *Player Life*. Selanjutnya dihasilkan jumlah setiap variabel sehingga mempengaruhi terjadinya perubahan otomatis sesuai apa yang telah didapatkan *player*.

#### **4.2.6. Tampilan Game Win**

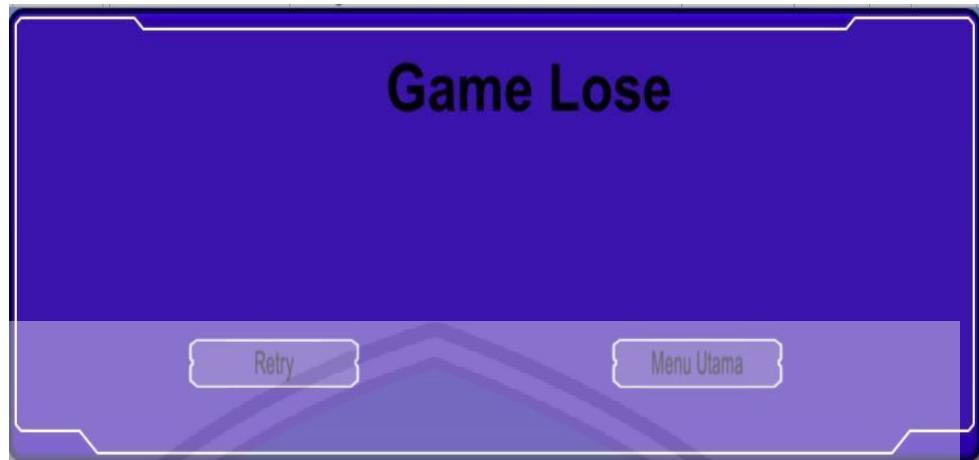
Setelah target *score* tercapai *game* akan memunculkan panel *Game Win*, dimana *game* telah berakhir dengan tampilan seperti pada gambar 4.8.



**Gambar 4.8 Panel Game Win**

#### **4.2.7. Tampilan Game Lose**

*Game Lose* terjadi ketika, ketika kondisi *player* telah melewati batas dimana, *Player Life* dan *Time Remain* telah habis dengan tampilan seperti pada gambar 4.9.



Gambar 4.9 Panel Game Lose

#### 4.2.8. Tampilan Input Nilai Variabel



Gambar 4.10 Panel Input Nilai Variabel BMGD

Pada gambar 4.10 untuk melihat atau mengecek level kabut, *User* dapat memasukkan input nilai variabel sehingga dihasilkan kabut di dalam *game*. Sebelum melakukan proses penentuan level, dilakukan terlebih dahulu perhitungan di dalam sistem *game*.

### 4.3. Tahap Pengujian

Pada pengujian sistem level kabut menggunakan *BMGD*, hal yang diamati adalah apakah sistem ini telah bekerja sesuai dengan tujuan yang telah dituliskan pada Bab 1. Selanjutnya pengujian akan dilakukan dengan memasukan nilai setiap variabel. Sebelum memasuki tahap pengujian sistem level kabut, peneliti akan melakukan beberapa tahap persiapan.

#### 4.3.1. Pembuatan Sistem Level

Pada tahap ini peneliti membuat *Script C#* dengan nama *Sistem\_Game* dimana *script* ini untuk mengatur sistem level didalam *game*.

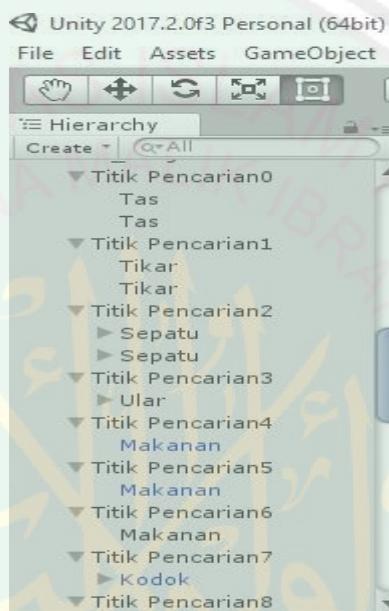


Gambar 4.11 Panel *Script* Sistem Level *BMGD* pada *Unity*

Pada gambar 4.11 terdapat “*Score Text*” dimana menampilkan *score* yang telah didapatkan melalui *point* setiap barang, “*Text\_Time Remain*” menampilkan *Time Remain* dengan awal waktu 300 detik waktu habis, “*Healthbar*” dan “*RatioText*” sebagai angka atau bar sisa *player life*, kemudian “*Text\_Level\_1*” hingga “*Text\_Level\_10*” sebagai tampilan Level setiap Kabut, terakhir “*InputScore*”, “*Input Time Remain*” dan “*Input Player Life*” dengan menggunakan *InputField* untuk memudahkan pengecekan kabut setiap level.

### 4.3.2. Objek Pendukung Sistem Level

Untuk mempengaruhi berubahnya level kabut secara *real time*, terdapat objek barang dan rintangan dimana objek barang mempengaruhi *score* yang didapatkan serta menambah *player life*, kemudian objek rintangan untuk mengurangi *player life* di dalam *game*. Berikut adalah gambar 4.12 yang menampilkan objek pendukung sistem level *game*.



Gambar 4.12 Objek Pendukung Sistem Level *Game*

### 4.4. Hasil Pengujian

Pada pengujian sistem level kabut menggunakan *BMGD* akan dibandingkan pada sistem level kabut tanpa *BMGD* dengan nilai masukan variabel yang sama dalam beberapa percobaan. Pada setiap percobaan dilakukan pengujian kembali menggunakan *Confusion Matrix* dalam klasifikasi dan prediksi akan dihitung menggunakan rumus akurasi, *precision*, *recall*, dan *f-measure* yang bertujuan untuk membandingkan hasil dari sistem yang menggunakan metode *BMGD* dan sistem yang tidak menggunakan metode *BMGD*. Dari pengujian *Confusion Matrix* tersebut dapat dilihat hasilnya untuk menentukan seberapa berhasil pengujian perbandingan

antara *game* yang menggunakan metode *BMGD* dan *game* yang tidak menggunakan metode *BMGD*.sebagai berikut:

#### 4.4.1. Percobaan 1

Alur yang sama, *BMGD* dan tanpa *BMGD* di dalam *Game*

**Tabel 4.1** Hasil Percobaan 1

No.	Variabel Masukan			Level Kabut		
	Score	Time Remain	Player Life	Desain	Tanpa BMGD	BMGD
1	0	300	100	1	1	1
2	8	288	100	1	2	1
3	12	278	66	2	3	2
4	16	268	70	2	4	2
5	18	254	72	3	4	3
6	18	247	38	4	4	4
7	22	232	42	5	5	5
8	22	222	22	6	5	6
9	26	214	22	6	6	6
10	26	198	22	7	6	7
11	32	182	24	7	7	7
12	38	174	26	7	8	7
13	42	162	46	6	9	6
14	46	150	50	7	10	7
15	48	142	52	7	10	7
16	50	134	23	8	10	8
17	50	128	6	9	10	9
18	50	118	6	10	10	10
19	54	105	22	9	10	9
20	54	88	44	9	10	9

Dari hasil level yang telah didapatkan akan dilakukan pengujian kembali dengan menggunakan *Confusion Matrix* dimana sumbu Y merepresentasikan *True Positive Rate* dan sumbu X merepresentasikan *False Positive Rate* sebagai berikut:

**Tabel 4.2** *Confusion Matrix* Percobaan 1

		Matrix									
X	1	2	3	4	5	6	7	8	9	10	
1	1	1	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	
4	0	0	0	1	0	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	1	0	
7	0	0	0	0	0	1	1	1	0	2	
8	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	0	0	0	0	0	0	3	
10	0	0	0	0	0	0	0	0	0	1	

- Perhitungan Akurasi

Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai benar.

Diketahui jumlah data uji adalah 20, kemudian data jumlah *True Positif* (*TP*) pada tabel di atas dengan kolom berwarna putih yaitu 6 data dan berjumlah 6.

Formula Akurasi :

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

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

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

- Perhitungan *Precision*

Presisi adalah data yang diambil berdasarkan informasi yang kurang atau salah dan tidak tepat.

Diketahui pada setiap kolom terdapat 6 data *True Positif* (*TP*) dan berjumlah 6, kemudian *False Positif* (*FP*) sebagai berikut:

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

$FP(8)=1; FP(9)=1; FP(10)=6$

Dengan jumlah  $FP$  adalah 14, setelah menemukan masing-masing nilai  $FP$  dari setiap kelas, selanjutnya akan dihitung presisi dari masing-masing kelas

Formula *Precision*:

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

$$P(1) = 1 / (1+0) = 1$$

$$P(2) = 0 / (0+1) = 0$$

$$P(3) = 0 / (0+1) = 0$$

$$P(4) = 1 / (1+2) = 0,33333$$

$$P(5) = 1 / (1+1) = 0,5$$

$$P(6) = 1 / (1+1) = 0,5$$

$$P(7) = 1 / (1+0) = 1$$

$$P(8) = 0 / (0+1) = 0$$

$$P(9) = 0 / (0+1) = 0$$

$$P(10) = 1 / (1+6) = 0,14285$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

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

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

$$Precision = (1 + 0 + 0 + 0,33333 + 0,5 + 0,5 + 1 + 0 + 0 + 0,14285) / 10$$

$$\textbf{Precision} = \mathbf{0,347618}$$

- Perhitungan *Recall*

*Recall* Adalah data yang tidak mampu diprediksi dengan benar.

Diketahui pada setiap kolom terdapat 6 data *True Positif* (*TP*) dan berjumlah 6, kemudian False Negatif (*FN*) dapat dilihat pada tabel sebagai berikut:

**Tabel 4.3 Confusion Matrix (*FN*) Percobaan 1**

x	Matrix									
	1	2	3	4	5	6	7	8	9	10
1	1	1	0	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	1	1	0	0	1	0
7	0	0	0	0	0	1	1	1	0	2
8	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	3
10	0	0	0	0	0	0	0	0	0	1

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

Setelah menemukan masing-masing nilai *FN*, proses perhitungan *Recall* dilakukan pada masing-masing kelasnya.

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

$$R(1) = 1/(1+1) = 0,5$$

$$R(2) = 0/(0+2) = 0$$

$$R(3) = 0/(0+1) = 0$$

$$R(4) = 1/(1+0) = 1$$

$$R(5) = 1/(1+0) = 1$$

$$R(6) = 1/(1+2) = 0,33333$$

$$R(7) = 1/(1+4) = 0,2$$

$$R(8) = 0/(0+1) = 0$$

$$R(9) = 0/(0+3) = 0$$

$$R(10) = 1/(1+0) = 1$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

$$\text{Recall} = (0,5 + 0 + 0 + 1 + 1 + 0,33333 + 0,2 + 0 + 0 + 1) / 10$$

$$\text{Recall} = \mathbf{0,403333}$$

- Perhitungan *F-measure*

*F-Measure* adalah perhitungan pembagian hasil dari perkalian *Precision* dan *Recall* dengan hasil penjumlahan *Precision* dan *Recall*, kemudian dikalikan dua.

Formula *F-measure*

$$\text{F-measure} = (2 * \text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$

$$\begin{aligned} \text{F-measure} &= (2 * 0,347618 * 0,403333) / (0,347618 + 0,403333) = \\ &0,280412 / 0,750951 \end{aligned}$$

$$\text{F-measure} = \mathbf{0,373409}$$

#### 4.4.2. Percobaan 2

Alur yang sama, *BMGD* dan tanpa *BMGD* di dalam *Game*

**Tabel 4.4** Hasil Percobaan 2

No.	Variabel Masukan			Level Kabut		
	Score	Time Remain	Player Life	Desain	Tanpa BMGD	BMGD
1	0	299	100	1	1	1
2	8	287	100	1	2	1
3	12	277	66	2	3	2
4	16	267	70	2	4	2
5	18	253	72	3	4	3
6	18	246	38	4	4	4
7	22	233	42	5	5	5

8	22	223	22	6	5	6
9	26	215	22	6	6	6
10	26	197	22	7	6	7
11	32	181	24	7	7	7
12	38	173	26	7	8	7
13	42	161	46	6	9	6
14	46	148	50	7	10	7
15	50	140	54	7	10	7
16	50	133	23	8	10	8
17	50	129	6	9	10	9
18	50	117	6	10	10	10
19	54	104	22	9	10	9
20	54	87	44	9	10	9

Dari hasil level yang telah didapatkan akan dilakukan pengujian kembali dengan menggunakan *Confusion Matrix* dimana sumbu Y merepresentasikan *True Positive Rate* dan sumbu X merepresentasikan *False Positive Rate* sebagai berikut:

**Tabel 4.5 Confusion Matrix Percobaan 2**

		Matrix									
X	1	2	3	4	5	6	7	8	9	10	
1	1	1	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	
4	0	0	0	1	0	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	1	0	
7	0	0	0	0	0	1	1	1	0	2	
8	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	0	0	0	0	0	0	3	
10	0	0	0	0	0	0	0	0	0	1	

- Perhitungan Akurasi

Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai benar.

Diketahui jumlah data uji adalah 20, kemudian data jumlah *True Positif* (*TP*) pada tabel di atas dengan kolom berwarna putih yaitu 6 data dan berjumlah 6.

Formula Akurasi :

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

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

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

- Perhitungan *Precision*

Presisi adalah data yang diambil berdasarkan informasi yang kurang atau salah dan tidak tepat.

Diketahui pada setiap kolom terdapat 6 data *True Positif* (*TP*) dan berjumlah 6, kemudian *False Positif* (*FP*) sebagai berikut:

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

Dengan jumlah *FP* adalah 14, setelah menemukan masing-masing nilai *FP* dari setiap kelas, selanjutnya akan dihitung presisi dari masing-masing kelas

Formula *Precision*:

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

$$\text{P}(1) = 1/(1+0) = 1$$

$$\text{P}(2) = 0/(0+1) = 0$$

$$\text{P}(3) = 0/(0+1) = 0$$

$$\text{P}(4) = 1/(1+2) = 0,33333$$

$$\text{P}(5) = 1/(1+1) = 0,5$$

$$\text{P}(6) = 1/(1+1) = 0,5$$

$$P(7) = 1/(1+0) = 1$$

$$P(8) = 0/(0+1) = 0$$

$$P(9) = 0/(0+1) = 0$$

$$P(10) = 1/(1+6) = 0,14285$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

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

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

$$\text{Precision} = (1 + 0 + 0 + 0,33333 + 0,5 + 0,5 + 1 + 0 + 0 + 0,14285) / 10$$

$$\text{Precision} = 0,347618$$

- Perhitungan *Recall*

*Recall* Adalah data yang tidak mampu diprediksi dengan benar.

Diketahui pada setiap kolom terdapat 6 data *True Positif* (*TP*) dan berjumlah 6, kemudian False Negatif (*FN*) dapat dilihat pada tabel sebagai berikut:

**Tabel 4.6 Confusion Matrix (*FN*) Percobaan 2**

		Matrix									
x	1	2	3	4	5	6	7	8	9	10	
1	1	1	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	
4	0	0	0	1	0	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	1	0	
7	0	0	0	0	0	1	1	1	0	2	
8	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	0	0	0	0	0	0	3	
10	0	0	0	0	0	0	0	0	0	1	

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

$$FN(8)=1; FN(9)=3; FN(10)=0$$

Setelah menemukan masing-masing nilai *FN*, proses perhitungan *Recall* dilakukan pada masing-masing kelasnya.

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

$$R(1) = 1/(1+1) = 0,5$$

$$R(2) = 0/(0+2) = 0$$

$$R(3) = 0/(0+1) = 0$$

$$R(4) = 1/(1+0) = 1$$

$$R(5) = 1/(1+0) = 1$$

$$R(6) = 1/(1+2) = 0,33333$$

$$R(7) = 1/(1+4) = 0,2$$

$$R(8) = 0/(0+1) = 0$$

$$R(9) = 0/(0+3) = 0$$

$$R(10) = 1/(1+0) = 1$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

$$\text{Recall} = (0,5 + 0 + 0 + 1 + 1 + 0,33333 + 0,2 + 0 + 0 + 1) / 10$$

$$\text{Recall} = 0,403333$$

- Perhitungan *F-measure*

*F-Measure* adalah perhitungan pembagian hasil dari perkalian *Precision* dan *Recall* dengan hasil penjumlahan *Precision* dan *Recall*, kemudian dikalikan dua.

Formula *F-measure*

$$\text{F-measure} = (2 * \text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$

$$F\text{-measure} = (2 * 0,347618 * 0,403333) / (0,347618 + 0,403333) =$$

$$0,280412 / 0,750951$$

$$\mathbf{F\text{-}measure} = \mathbf{0,373409}$$

#### 4.4.3. Percobaan 3

Alur yang sama, *BMGD* dan tanpa *BMGD* di dalam *Game*

**Tabel 4.7** Hasil Percobaan 3

No.	Variabel Masukan			Level Kabut		
	Score	Time Remain	Player Life	Desain	Tanpa BMGD	BMGD
1	0	298	100	1	1	1
2	8	286	100	1	2	1
3	12	276	66	2	3	2
4	16	266	70	2	4	2
5	18	252	72	3	4	3
6	20	244	40	4	5	4
7	22	234	42	5	5	5
8	22	224	22	6	5	6
9	26	216	22	6	6	6
10	26	196	22	7	6	7
11	32	183	24	7	7	7
12	38	172	26	7	8	7
13	42	163	46	6	9	6
14	46	149	50	7	10	7
15	48	141	52	7	10	7
16	50	132	23	8	10	8
17	50	127	6	9	10	9
18	50	116	6	10	10	10
19	52	106	20	10	10	10
20	54	89	44	9	10	9

Dari hasil level yang telah didapatkan akan dilakukan pengujian kembali dengan menggunakan *Confusion Matrix* dimana sumbu Y merepresentasikan *True Positive Rate* dan sumbu X merepresentasikan *False Positive Rate* sebagai berikut:

**Tabel 4.8** *Confusion Matrix* Percobaan 3

		Matrix									
X	1	2	3	4	5	6	7	8	9	10	
1	1	1	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	
4	0	0	0	0	1	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	1	0	
7	0	0	0	0	0	1	1	1	0	2	
8	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	0	0	0	0	0	0	2	
10	0	0	0	0	0	0	0	0	0	2	

- Perhitungan Akurasi

Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai benar.

Diketahui jumlah data uji adalah 20, kemudian data jumlah *True Positif* (*TP*) pada tabel di atas dengan kolom berwarna putih yaitu 5 data dan berjumlah 6.

Formula Akurasi :

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

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

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

- Perhitungan *Precision*

Presisi adalah data yang diambil berdasarkan informasi yang kurang atau salah dan tidak tepat.

Diketahui pada setiap kolom terdapat 5 data *True Positif* (*TP*) dan berjumlah 6, kemudian *False Positif* (*FP*) sebagai berikut:

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

$FP(8)=1; FP(9)=1; FP(10)=5$

Dengan jumlah  $FP$  adalah 14, setelah menemukan masing-masing nilai  $FP$  dari setiap kelas, selanjutnya akan dihitung presisi dari masing-masing kelas

Formula *Precision*:

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

$$P(1) = 1 / (1+0) = 1$$

$$P(2) = 0 / (0+1) = 0$$

$$P(3) = 0 / (0+1) = 0$$

$$P(4) = 0 / (0+2) = 0$$

$$P(5) = 1 / (1+2) = 0,33333$$

$$P(6) = 1 / (1+1) = 0,5$$

$$P(7) = 1 / (1+0) = 1$$

$$P(8) = 0 / (0+1) = 0$$

$$P(9) = 0 / (0+1) = 0$$

$$P(10) = 2 / (2+5) = 0,28571$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

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

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

$$Precision = (1 + 0 + 0 + 0 + 0,33333 + 0,5 + 1 + 0 + 0 + 0,28571) / 10$$

$$\textbf{Precision} = \mathbf{0,311904}$$

- Perhitungan *Recall*

*Recall* Adalah data yang tidak mampu diprediksi dengan benar.

Diketahui pada setiap kolom terdapat 5 data *True Positif* (*TP*) dan berjumlah 6, kemudian False Negatif (*FN*) dapat dilihat pada tabel sebagai berikut:

**Tabel 4.9 Confusion Matrix (*FN*) Percobaan 3**

x	Matrix									
	1	2	3	4	5	6	7	8	9	10
1	1	1	0	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	1	1	0	0	1	0
7	0	0	0	0	0	1	1	1	0	2
8	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	2
10	0	0	0	0	0	0	0	0	0	2

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

Setelah menemukan masing-masing nilai *FN*, proses perhitungan *recall* dilakukan pada masing-masing kelasnya.

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

$$R(1) = 1/(1+1) = 0,5$$

$$R(2) = 0/(0+2) = 0$$

$$R(3) = 0/(0+1) = 0$$

$$R(4) = 0/(0+0) = 0$$

$$R(5) = 1/(1+0) = 1$$

$$R(6) = 1/(1+2) = 0,33333$$

$$R(7) = 1/(1+4) = 0,2$$

$$R(8) = 0/(0+1) = 0$$

$$R(9) = 0/(0+2) = 0$$

$$R(10) = 2/(2+0) = 1$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

$$\text{Recall} = (0,5 + 0 + 0 + 0 + 1 + 0,33333 + 0,2 + 0 + 0 + 1) / 10$$

$$\text{Recall} = \mathbf{0,303333}$$

- Perhitungan *F-measure*

*F-Measure* adalah perhitungan pembagian hasil dari perkalian *Precision* dan *Recall* dengan hasil penjumlahan *Precision* dan *Recall*, kemudian dikalikan dua.

Formula *F-measure*

$$\text{F-measure} = (2 * \text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$

$$\begin{aligned} \text{F-measure} &= (2 * 0,311904 * 0,303333) / (0,311904 + 0,303333) = \\ &0,189222 / 0,615237 \end{aligned}$$

$$\text{F-measure} = \mathbf{0,307559}$$

#### 4.4.4. Percobaan 4

Alur yang sama, *BMGD* dan tanpa *BMGD* di dalam *Game*

**Tabel 4.10** Hasil Percobaan 4

No.	Variabel Masukan			Level Kabut		
	Score	Time Remain	Player Life	Desain	Tanpa BMGD	BMGD
1	0	297	100	1	1	1
2	8	285	100	1	2	1
3	12	275	66	2	3	2
4	16	263	70	3	4	3
5	18	260	72	2	4	2
6	20	245	40	4	5	4
7	22	235	42	5	5	5

8	22	225	22	6	5	6
9	26	217	22	6	6	6
10	28	195	24	7	6	7
11	32	184	24	7	7	7
12	42	170	30	7	9	7
13	42	160	46	7	9	7
14	46	147	50	7	10	7
15	50	138	23	8	10	8
16	50	129	6	9	10	9
17	50	122	6	10	10	10
18	50	114	18	9	10	9
19	52	107	20	10	10	10
20	54	86	44	9	10	9

Dari hasil level yang telah didapatkan akan dilakukan pengujian kembali dengan menggunakan *Confusion Matrix* dimana sumbu Y merepresentasikan *True Positive Rate* dan sumbu X merepresentasikan *False Positive Rate* sebagai berikut:

**Tabel 4.11** *Confusion Matrix* Percobaan 4

Matrix										
X	1	2	3	4	5	6	7	8	9	10
1	1	1	0	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	1	1	0	0	0	0
7	0	0	0	0	0	1	1	0	2	1
8	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	3
10	0	0	0	0	0	0	0	0	0	2

- Perhitungan Akurasi

Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai benar.

Diketahui jumlah data uji adalah 20, kemudian data jumlah *True Positif* (*TP*) pada tabel di atas dengan kolom berwarna putih yaitu 5 data dan berjumlah 6.

Formula Akurasi :

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

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

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

- Perhitungan *Precision*

Presisi adalah data yang diambil berdasarkan informasi yang kurang atau salah dan tidak tepat.

Diketahui pada setiap kolom terdapat 5 data *True Positif* (*TP*) dan berjumlah 6, kemudian *False Positif* (*FP*) sebagai berikut:

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

Dengan jumlah *FP* adalah 14, setelah menemukan masing-masing nilai *FP* dari setiap kelas, selanjutnya akan dihitung presisi dari masing-masing kelas

Formula *Precision*:

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

$$\text{P}(1) = 1 / (1+0) = 1$$

$$\text{P}(2) = 0 / (0+1) = 0$$

$$\text{P}(3) = 0 / (0+1) = 0$$

$$\text{P}(4) = 0 / (0+2) = 0$$

$$\text{P}(5) = 1 / (1+2) = 0,33333$$

$$\text{P}(6) = 1 / (1+1) = 0,5$$

$$P(7) = 1/(1+0) = 1$$

$$P(8) = 0/(0+0) = 0$$

$$P(9) = 0/(0+2) = 0$$

$$P(10) = 2/(2+5) = 0,28571$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

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

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

$$Precision = (1 + 0 + 0 + 0 + 0,33333 + 0,5 + 1 + 0 + 0 + 0,28571) / 10$$

$$Precision = 0,311904$$

- Perhitungan *Recall*

*Recall* Adalah data yang tidak mampu diprediksi dengan benar.

Diketahui pada setiap kolom terdapat 5 data *True Positif* (*TP*) dan berjumlah 6, kemudian *False Negatif* (*FN*) dapat dilihat pada tabel sebagai berikut:

**Tabel 4.12 Confusion Matrix (*FN*) Percobaan 4**

		Matrix									
x	1	2	3	4	5	6	7	8	9	10	
1	1	1	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	
4	0	0	0	0	1	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	0	0	
7	0	0	0	0	0	1	1	0	2	1	
8	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	0	0	0	0	0	0	3	
10	0	0	0	0	0	0	0	0	0	2	

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

$$FN(8)=1; FN(9)=3; FN(10)=0$$

Setelah menemukan masing-masing nilai *FN*, proses perhitungan *recall* dilakukan pada masing-masing kelasnya.

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

$$R(1) = 1/(1+1) = 0,5$$

$$R(2) = 0/(0+2) = 0$$

$$R(3) = 0/(0+1) = 0$$

$$R(4) = 0/(0+1) = 0$$

$$R(5) = 1/(1+0) = 1$$

$$R(6) = 1/(1+1) = 0,5$$

$$R(7) = 1/(1+4) = 0,2$$

$$R(8) = 0/(0+1) = 0$$

$$R(9) = 0/(0+3) = 0$$

$$R(10) = 2/(2+0) = 1$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

$$\text{Recall} = (0,5 + 0 + 0 + 0 + 1 + 0,5 + 0,2 + 0 + 0 + 1) / 10$$

$$\text{Recall} = 0,32$$

- Perhitungan *F-measure*

*F-Measure* adalah perhitungan pembagian hasil dari perkalian *Precision* dan *Recall* dengan hasil penjumlahan *Precision* dan *Recall*, kemudian dikalikan dua.

Formula *F-measure*

$$F\text{-measure} = (2 * \text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$

$$F\text{-measure} = (2 * 0,311904 * 0,32) / (0,311904 + 0,32)$$

$$= 0,199618 / 0,63119$$

$$\textbf{F-measure} = \mathbf{0,316257}$$

#### 4.4.5. Percobaan 5

Alur yang sama, *BMGD* dan tanpa *BMGD* di dalam *Game*

**Tabel 4.13** Hasil Percobaan 5

No.	Variabel Masukan			Level Kabut		
	Score	Time Remain	Player Life	Desain	Tanpa BMGD	BMGD
1	4	292	100	1	1	1
2	12	284	100	1	3	1
3	14	274	68	2	3	2
4	16	262	70	3	4	3
5	18	261	72	2	4	2
6	20	243	40	4	5	4
7	22	236	42	4	5	4
8	22	226	22	6	5	6
9	26	218	22	6	6	6
10	28	194	24	7	6	7
11	32	185	24	7	7	7
12	42	171	30	7	9	7
13	42	159	46	7	9	7
14	48	146	52	7	10	7
15	50	137	23	8	10	8
16	50	130	6	9	10	9
17	50	121	6	10	10	10
18	50	113	18	9	10	9
19	54	102	22	10	10	10
20	54	84	44	9	10	9

Dari hasil level yang telah didapatkan akan dilakukan pengujian kembali dengan menggunakan *Confusion Matrix* dimana sumbu Y merepresentasikan *True Positive Rate* dan sumbu X merepresentasikan *False Positive Rate* sebagai berikut:

**Tabel 4.14 Confusion Matrix** Percobaan 5

		Matrix									
X	1	2	3	4	5	6	7	8	9	10	
1	1	0	1	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	
4	0	0	0	0	2	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	0	0	
7	0	0	0	0	0	1	1	0	2	1	
8	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	0	0	0	0	0	0	3	
10	0	0	0	0	0	0	0	0	0	2	

- Perhitungan Akurasi

Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai benar.

Diketahui jumlah data uji adalah 20, kemudian data jumlah *True Positif* (*TP*) pada tabel di atas dengan kolom berwarna putih yaitu 4 data dan berjumlah 5.

Formula Akurasi :

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

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

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

- Perhitungan *Precision*

Presisi adalah data yang diambil berdasarkan informasi yang kurang atau salah dan tidak tepat.

Diketahui pada setiap kolom terdapat 5 data *True Positif* (*TP*) dan berjumlah 6, kemudian *False Positif* (*FP*) sebagai berikut:

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

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

Dengan jumlah  $FP$  adalah 15, setelah menemukan masing-masing nilai  $FP$  dari setiap kelas, selanjutnya akan dihitung presisi dari masing-masing kelas

Formula *Precision*:

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

$$P(1) = 1 / (1+0) = 1$$

$$P(2) = 0 / (0+0) = 0$$

$$P(3) = 0 / (0+2) = 0$$

$$P(4) = 0 / (0+2) = 0$$

$$P(5) = 0 / (0+3) = 0$$

$$P(6) = 1 / (1+1) = 0,5$$

$$P(7) = 1 / (1+0) = 1$$

$$P(8) = 0 / (0+0) = 0$$

$$P(9) = 0 / (0+2) = 0$$

$$P(10) = 2 / (2+5) = 0,28571$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

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

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

$$Precision = (1 + 0 + 0 + 0 + 0 + 0,5 + 1 + 0 + 0 + 0,28571) / 10$$

$$\textbf{Precision} = \mathbf{0,27857}$$

- Perhitungan *Recall*

*Recall* Adalah data yang tidak mampu diprediksi dengan benar.

Diketahui pada setiap kolom terdapat 4 data *True Positif* (*TP*) dan berjumlah 5, kemudian False Negatif (*FN*) dapat dilihat pada tabel sebagai berikut:

**Tabel 4.15 Confusion Matrix (*FN*) Percobaan 5**

		Matrix									
x	1	2	3	4	5	6	7	8	9	10	
1	1	0	1	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	
4	0	0	0	0	2	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	0	0	
7	0	0	0	0	0	1	1	0	2	1	
8	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	0	0	0	0	0	0	3	
10	0	0	0	0	0	0	0	0	0	2	

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

Setelah menemukan masing-masing nilai *FN*, proses perhitungan *recall* dilakukan pada masing-masing kelasnya.

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

$$R(1) = 1/(1+1) = 0,5$$

$$R(2) = 0/(0+2) = 0$$

$$R(3) = 0/(0+1) = 0$$

$$R(4) = 0/(0+2) = 0$$

$$R(5) = 1/(1+0) = 1$$

$$R(6) = 1/(1+1) = 0,5$$

$$R(7) = 1/(1+4) = 0,2$$

$$R(8) = 0/(0+1) = 0$$

$$R(9) = 0/(0+3) = 0$$

$$R(10) = 2/(2+0) = 1$$

Dari jumlah semua perhitungan di atas, maka dijumlahkan lagi sebagai berikut:

$$\text{Recall} = (0,5 + 0 + 0 + 0 + 1 + 0,5 + 0,2 + 0 + 0 + 1) / 10$$

$$\text{Recall} = 0,32$$

- Perhitungan *F-measure*

*F-Measure* adalah perhitungan pembagian hasil dari perkalian *Precision* dan *Recall* dengan hasil penjumlahan *Precision* dan *Recall*, kemudian dikalikan dua.

Formula *F-measure*

$$\text{F-measure} = (2 * \text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$

$$\text{F-measure} = (2 * 0,27857 * 0,32) / (0,27857 + 0,32)$$

$$= 0,178284 / 0,59857$$

$$\text{F-measure} = 0,29785$$

Berikut rekapitulasi dari percobaan yang telah dilakukan:

**Tabel 4.16** Rekapitulasi Perhitungan *Confusion Matrix*

Uraian	Akurasi	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>
Percobaan 1	0,3	0,348	0,403	0,373
Percobaan 2	0,3	0,348	0,403	0,373
Percobaan 3	0,3	0,312	0,303	0,308
Percobaan 4	0,3	0,312	0,32	0,316
Percobaan 5	0,25	0,279	0,32	0,298
Total	1,45	1,599	1,749	1,668
Rata-rata	0,29	0,319	0,349	0,334

Dari pengujian menggunakan *Confusion Matrix* yang telah dilakukan terhadap beberapa percobaan, dapat dilihat menjadi satu pada tabel 4.16. Dari tabel

4.16 dijelaskan nilai dari akurasi, *precision*, *recall*, dan *f-measure* dari semua percobaan yang telah dilakukan.

Pada percobaan dan pengujian yang telah dilakukan, nilai akurasi tertinggi diperoleh pada percobaan pertama, kedua, ketiga, dan keempat dengan nilai akurasi 0,3 atau sebesar 30%. Akurasi tersebut lebih tinggi apabila dibandingkan dengan percobaan kelima yang mempunyai jumlah *True Positif* lebih sedikit. Nilai akurasi pada percobaan ini menandakan seberapa dekat prediksi yang dihasilkan oleh sistem yang menggunakan metode *BMGD* terhadap sistem yang tidak menggunakan metode *BMGD* dari keseluruhan data uji.

Kemudian pada nilai *precision* menunjukkan seberapa banyak sistem memprediksi kelas positif dari data yang memang positif (aktual). Dalam hal ini *precision* sangat dipengaruhi oleh *False Positif*, sehingga semakin banyak jumlah *False Positif* maka nilai *precision* akan semakin rendah. Oleh karena itu *precision* menunjukkan berapa persen prediksi dari sistem yang tepat sasaran.

Selanjutnya pada nilai *recall* berguna untuk mengukur seberapa berhasil sistem dapat mengenali kembali sampel positif pada data uji. Nilai *recall* dipengaruhi oleh *False Negatif* dan *True Positif*, sehingga semakin banyak jumlah *False Negatif* maka nilai *recall* akan semakin rendah.

Setelah itu diperoleh nilai *f-measure* sebagai penyeimbang antara nilai *precision* dan *recall* karena kedua nilai tersebut sama-sama penting. Pada pengujian ini *f-measure* dapat membantu menentukan percobaan mana yang lebih baik.

**Tabel 4.17** Rekapitulasi Perbandingan Level Kabut BMGD dan Tanpa BMGD

No.	Level Kabut	
	Tanpa BMGD	BMGD

1	1	1
2	2	1
3	3	2
4	4	2
5	4	3
6	5	4
7	5	5
8	5	6
9	6	6
10	6	7
11	7	7
12	8	7
13	9	6
14	10	7
15	10	7
16	10	8
17	10	9
18	10	10
19	10	10
20	10	9

Dari hasil percobaan yang telah dilakukan dapat diperoleh nilai modus yang dijelaskan pada tabel 4.17. *Game* yang tidak menggunakan metode *BMGD* lebih statis dengan perubahan level yang bertambah terus, sedangkan *game* yang menggunakan metode *BMGD* lebih dinamis dalam menentukan level yaitu sesuai dengan kemampuan player. Sebagaimana pada Al-Qur'an Surat Al-Baqarah Ayat 286:

لَا يُكَافِئُ اللَّهُ تَفْسِيْأً إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْسَبَتْ رَبَّنَا لَا تُؤَاخِذنَا إِنَّنَا أُوْخَطَانَا  
 رَبَّنَا وَلَا تَحْمِلْنَا عَلَيْنَا إِصْرًا كَمَا حَمَلْنَاهُ عَلَى الَّذِينَ مِنْ قَبْلِنَا رَبَّنَا وَلَا تُحَمِّلْنَا مَا لَا طَاقَةَ لَنَا بِهِ وَاعْفُ  
 عَنَا وَاعْفُرْ لَنَا وَارْحَمْنَا أَنْتَ مَوْلَانَا فَانْصُرْنَا عَلَى الْقَوْمِ الْكُفَّارِينَ

Artinya :

*"Allah tidak membebani seseorang melainkan sesuai dengan kemampuannya. Ia mendapat pahala (dari kebaikan) yang diusahakannya dan ia mendapat siksa (dari kejahanatan) yang dikerjakannya. (Mereka berdoa): "Ya Tuhan kami, janganlah Engkau hukum kami jika kami lupa atau kami tersalah. Ya Tuhan kami, janganlah Engkau bebankan kepada kami beban yang berat sebagaimana Engkau bebankan kepada orang-orang sebelum kami. Ya Tuhan kami, janganlah Engkau pikulkan kepada kami apa yang tak sanggup kami memikulnya. Beri maaflah kami; ampunilah kami; dan rahmatilah kami. Engkaulah Penolong kami, maka tolonglah kami terhadap kaum yang kafir". (QS. Al-Baqarah: 286)*

Al-Hafidh Ibnu Katsir menjelaskan dalam tafsirnya menjelaskan Surat Al-Baqarah Ayat 286 bahwa Allah SWT tidak membebani hamba-Nya dengan amalan-amalan yang berat, sekalipun hamba-Nya masih mampu untuk melakukannya, sebagaimana yang disyari'atkan pada umat-umat terdahulu dengan dibelenggu dan diikat. Sebagaimana Allah SWT mengutus Nabi Muhammad SAW, Nabi pembawa rahmat, yang telah dijadikan sebagai ciri yang menonjol dalam syari'atnya, sebagaimana Allah SWT telah mengutusnya dengan membawa agama yang lurus, yang penuh dengan kemudahan dan toleransi. Maksudnya kemudahan dari beban kewajiban-kewajiban, musibah serta bencana dari perkara yang hamba-Nya tidak sanggup menanggungnya. (Amin, 2013)

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan beberapa hal pada penerapan metode *Box Muller of Gaussian Distribution* untuk menentukan level kesulitan pada *game* 3D mitigasi bencana alam berbasis kabut, yaitu sebagai berikut:

1. Hasil terbaik dari pengukuran kinerja metode *Box Muller Of Gaussian Distribution* ditunjukkan pada percobaan pertama dan percobaan kedua dengan nilai yang sama yaitu akurasi sebesar 30%, *precision* sebesar 34,8%, *recall* sebesar 40,3%, dan *f-measure* sebesar 37,3%.
2. Dari hasil perbandingan *game* yang menggunakan metode *BMGD* dengan *game* yang tidak menggunakan metode *BMGD* didapatkan bahwa sistem level kabut dengan menggunakan metode *BMGD* dapat berjalan secara otomatis dimana penyesuaian dapat berubah secara lebih dinamis dengan melihat kinerja *player*.

#### 5.2. Saran

Setelah peneliti melakukan penelitian ini, peneliti mencoba memberikan saran kepada beberapa pihak. Berikut beberapa saran yang dapat diajukan antara lain:

1. Menambah objek yang berguna lebih banyak lagi untuk memberikan pengaruh dan perubahan yang lebih variatif pada nilai variabel untuk penentuan level.

2. Diperlukan pengembangan *game* untuk *player* yang memainkan *game* secara langsung sehingga *player* selain *refreshing* juga mendapatkan pengalaman bermain yang lebih baik.



## DAFTAR PUSTAKA

- Amin, S. (2013). *Tafsir Penutup Surat al-Baqarah*.
- Boiroju, N. K., & Reddy, M. K. (2012). Generation of standard normal random numbers. *Interstat, May*, 1–14.  
<http://interstat.statjournals.net/INDEX/May12.html>
- Ma'ruf Harsono. (2014). Pengaruh Bermain Game Terhadap Perkembangan Remaja. *Surya University*, 5.  
<https://doi.org/10.1017/CBO9781107415324.004>
- Majid, N. K., & Hariadi, M. (2010). Distribusi Gaussian Perilaku Tarung NPC Prajurit Pada Game Peperangan Menggunakan Metode Box-Muller. *Jaringan Cerdas Multimedia (Game Technology)*.
- R, K., Agus, R., Ofyar, T., & Ade, S. (2005). Penggunaan Distribusi Normal Dalam Memodelkan Sebaran Persepsi Biaya Perjalanan Dan Transportasi Box-Muller Pada Pengambilan Sampel Acak Model Pemilihan Rute Dan Pembebanan Stokastik. *Jurnal Transportasi*.
- Rollings, A., & Ernest, A. (2003). Andrew Rollings and Ernest Adams on Game Design. *Carcinogenesis*. <https://doi.org/10.1093/carcin/bgs054>
- Sudjana, Nana. (2013). Dasar-Dasar Proses Belajar Mengajar. Sinar Baru Algensindo.
- Sukajaya, I. N., Vitianingsih, A. V., Mardi, S. N. S., Purnama, K. E., Hariadi, M., & Purnomo, M. H. (2012). Multi-parameter dynamic difficulty game's scenario using Box-Muller of Gaussian distribution. *ICCSE 2012 - Proceedings of 2012 7th International Conference on Computer Science and Education*. <https://doi.org/10.1109/ICCSE.2012.6295384>
- Thomas, D. B., Luk, W., Leong, P. H. W., & Villasenor, J. D. (2007). Gaussian random number generators. *ACM Computing Surveys*.  
<https://doi.org/10.1145/1287620.1287622>
- Thuan, H., Saurik, T., Yuniarno, E. M., Susiki, S. M., Elektro, J. T., & Industri, F. T. (2015). *Kepuasan pemain terhadap desain rintangan pada skenario game*

- petualang.* 2015(November), 156–164.
- Vega, A., & Mardi, S. (2011). Proceeding on The 12th Seminar on Intelligent Technology and Its-Istitut Teknologi Sepuluh Nopember (ITS) Surabaya.  
*DISTRIBUSI GAUSSIAN UNTUK DINAMIK SKENARIO GAME MENGGUNAKAN METODE BOX MULLER.*
- Warpefelt, H. (2016). *The Non-Player Character: Exploring the believability of NPC presentation and behavior* (Issue 16).

