

**IMPLEMENTASI *PATHFINDING* DENGAN ALGORITMA
FLOYD-WARSHALL PADA NPC UNTUK GAME
MITIGASI GUNUNG MELETUS**

SKRIPSI

Oleh :
FIRHAN ADE ACHMAD PRAYOGA
NIM. 15650004



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

**IMPLEMENTASI *PATHFINDING* DENGAN ALGORITMA
FLOYD-WARSHALL PADA *NPC* UNTUK *GAME*
MITIGASI GUNUNG MELETUS**

SKRIPSI

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

**Oleh:
FIRHAN ADE ACHMAD PRAYOGA
NIM. 15650004**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

LEMBAR PERSETUJUAN

**IMPLEMENTASI PATHFINDING DENGAN ALGORITMA
FLOYD-WARSHALL PADA NPC UNTUK GAME
MITIGASI GUNUNG MELETUS**

SKRIPSI

Oleh :
FIRHAN ADE ACHMAD PRAYOGA
NIM. 15650004

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : 20 Desember 2019

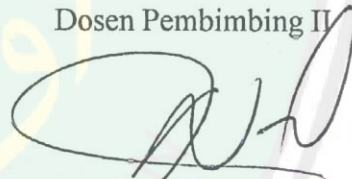
Dosen Pembimbing I



Hani Nurhayati, M.T.

NIP. 19780625 200801 2 006

Dosen Pembimbing II



Fresy Nugroho, M.T.

NIP. 19710722 201101 1 001

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 *PATHFINDING* DENGAN ALGORITMA
FLOYD-WARSHALL PADA *NPC* UNTUK *GAME*
MITIGASI GUNUNG MELETUS**

SKRIPSI

Oleh:
FIRHAN ADE ACHMAD PRAYOGA
NIM. 15650004

Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal 20 Desember 2019

Susunan Dewan Penguji

- | | |
|-----------------------|--|
| 1. Penguji Utama | : <u>Khadijah F.H. Holle, M.Kom</u>
NIDT. 19900626 20160801 2 077 |
| 2. Ketua Penguji | : <u>Yunifa Miftachul Arif, M.T</u>
NIP. 19830616 201101 1 004 |
| 3. Sekretaris Penguji | : <u>Hani Nurhayati, M.T</u>
NIP. 19780625 200801 2 006 |
| 4. Anggota Penguji | : <u>Fresy Nugroho, M.T</u>
NIP. 19710722 201101 1 001 |

Tanda tangan

(*[Signature]*)
(*[Signature]*)
(*[Signature]*)
(*[Signature]*)

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



Dr. Cahyo Crysdiyan
NIP. 196740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Firhan Ade Achmad Prayoga
NIM : 15650004
Fakultas/Jurusan : Sains dan Teknologi/Teknik Informatika
Judul Skripsi : Implementasi *Pathfinding* Dengan Algoritma *Floyd-Warshall*
Pada *NPC* Untuk *Game* Mitigasi Gunung Meletus

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 dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 Desember 2019
Yang membuat pernyataan,



Firhan Ade Achmad Prayoga
NIM. 15650004

HALAMAN MOTTO

*“Bakat bisa membawamu jauh,
Tapi kerja keras bisa membawamu kemana saja”*



HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Atas kehadiran Allah Subhanahu wa ta'ala, dengan mengucap syukur alhamdulillah penulis mempersembahkan sebuah karya untuk orang – orang yang sangat berarti

Terima kasih diucapkan kepada kedua orang tua saya yang selalu memberikan dukungan, motivasi dan juga doa. Dan selalu mengajarkan serta mendidik berbagai nilai – nilai dalam kehidupan. Serta teruntuk adik saya sebagai saudara kandung yang telah memberikan bantuan dan motivasi kepada saya.

Terima kasih pula diucapkan untuk ibu dan bapak pembimbing yang telah membimbing dalam melakukan penelitian ini dan memberikan motivasi serta dorongan hingga penelitian terselesaikan dengan lancar.

Tak lupa juga diucapkan terima kasih kepada sahabat – sahabat saya yang telah mendukung serta mendoakan selama saya masuk perkuliahan di UIN Maulana Malik Ibrahim Malang. Dan juga keluarga besar Teknik Informatika Interface 2015 yang telah memberikan semangat dan doa – doa yang dikirimkan.

Terima kasih untuk orang – orang yang tidak bisa disebutkan satu per satu yang telah mendoakan serta mendukung sehingga dapat terselesaikan skripsi ini.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji dan syukur penulis panjatkan ke hadirat Allah subhanahu wa ta'ala yang telah melimpahkan rahmat dan hidayahNya kepada kita, sehingga penulis bisa menyelesaikan skripsi dengan tepat waktu, yang kami beri judul “Implementasi *Pathfinding* Dengan Algoritma *Floyd-warshall* Pada *NPC* Untuk *Game* Mitigasi Gunung Meletus”. Tujuan dari penyusunan skripsi ini guna memenuhi salah satu syarat untuk bisa menempuh ujian sarjana komputer pada Fakultas Sains dan Teknologi (FSAINTEK) Program Studi Teknik Informatika di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Didalam pengerjaan skripsi ini telah melibatkan banyak pihak yang sangat membantu dalam banyak hal. Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. Abdul Haris, M.Ag selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan, Selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Hani Nurhayati, M.T, selaku Dosen Pembimbing I yang telah membimbing dalam penyusunan skripsi ini hingga selesai.
5. Fresy Nugroho, M.T, selaku Dosen Pembimbing II yang telah membimbing dalam penyusunan skripsi ini hingga selesai.

6. Prof. Dr. Suhartono, S.Si., M.Kom. selaku Dosen Wali yang senantiasa memberikan banyak motivasi dan saran untuk kebaikan penulis.
7. Kedua orang tua tercinta Bapak Achmad Baidowi dan Ibu Sri Anik Mumpuni yang telah banyak memberikan doa dan dukungan kepada penulis secara moril maupun materil hingga skripsi ini dapat terselesaikan.
8. Adik tercinta Tabah Rizki juga anggota keluarga dan kerabat yang senantiasa memberikan doa dan dukungan semangat kepada penulis.
9. Sahabat – sahabat seperjuangan yaitu Afif Nuril Ihsan, Yulfian Akbar Kharismawan, Yastaqim Muqorrobin, Nadya Putri Harfianti, Ahmad Zaky Rozini, Shovan Fanny Mahmud, Muhammad Firyal Alfarisi, Abdul Aziz Iswahyudi yang tiada henti memberi dukungan dan motivasi kepada penulis serta target bersama untuk lulus skripsi dan wisuda bersama.
10. Sahabat – sahabat sepermainan yaitu Andrias Rico, Mutiarani Esti, Ayu Sukma, Annissa Kusumawardhani, Putri Ikrimah, Sofie Achmad, Fadhil Al Amal, Farrah Nurmalia Sari, Berlian Gita Cahyani, dan Evi Susanti.
11. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak bisa penulis sebutkan semuanya.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Malang, 20 Desember 2019

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
ABSTRAK	xiii
ABSTRACT	xiv
المخلص	xv
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penelitian.....	5
BAB 2	6
LANDASAN TEORI	6
2.1 Penelitian Terkait.....	6
2.2 <i>Game</i>	8
2.3 Gunung Meletus.....	14
2.4 <i>Unity</i>	22
2.5 Algoritma <i>Floyd-Warshall</i>	26
BAB 3	34
METODOLOGI PENELITIAN	34
3.1 Deskripsi <i>Game</i>	34
3.2 <i>Storyline</i>	34

3.3	Desain <i>Storyboard</i>	36
3.4	<i>User Interface</i>	39
3.5	Deskripsi Karakter dan Objek.....	42
3.6	<i>Finite State Machine</i> (FSM)	43
3.7	Perhitungan <i>Floyd-Warshall</i>	44
BAB 4	48
HASIL DAN PEMBAHASAN	48
4.1	Implementasi.....	48
4.1.1	Implementasi Perangkat Keras	48
4.1.2	Implementasi Perangkat Lunak	49
4.2	Implementasi <i>Game</i>	49
4.2.1	Antarmuka <i>Main Menu</i>	49
4.2.2	Antarmuka <i>How to play</i>	50
4.2.3	Antarmuka <i>Option</i>	50
4.2.4	Konten <i>Game</i>	51
4.2.5	Misi <i>Game</i>	52
4.2.6	<i>Mission Complete</i>	52
4.2.7	<i>Game Over</i>	53
4.2.8	Karakter dan Objek yang Digunakan	53
4.3	Implementasi Algoritma <i>Floyd-Warshall</i>	55
4.4	Integrasi dengan Islam	62
BAB 5	65
PENUTUP	65
5.1	Kesimpulan.....	65
5.2	Saran.....	65
DAFTAR PUSTAKA	66

DAFTAR GAMBAR

Gambar 2. 1 Game Action	10
Gambar 2. 2 Game Shooter.....	10
Gambar 2. 3 Game Racing.....	11
Gambar 2. 4 Game Sport	12
Gambar 2. 5 Game Adventure	12
Gambar 2. 6 Game Strategy	13
Gambar 2. 7 Game RPG	14
Gambar 2. 8 Script Writer.....	23
Gambar 2. 9 Platform Unity.....	24
Gambar 2. 10 Asset Store	25
Gambar 2. 11 Contoh Gambar Graf Berbobot.....	28
Gambar 3. 1 Desain Main Menu.....	39
Gambar 3. 2 Desain Menu Option	40
Gambar 3. 3 Desain Menu How to Play	40
Gambar 3. 4 Tampilan Gameplay	41
Gambar 3. 5 Kondisi Berbicara.....	43
Gambar 3. 6 Kondisi Waypoint	43
Gambar 3. 7 Titik Pada Denah Game	45
Gambar 3. 8 Pseudocode algoritma floyd-warshall.....	47
Gambar 4. 1 MainMenu	50
Gambar 4. 2 Panel How To Play	50
Gambar 4. 3 Panel Option.....	51
Gambar 4. 4 Tampilan gameplay.....	51
Gambar 4. 5 Tampilan Misi	52
Gambar 4. 6 Mission Complete	52
Gambar 4. 7 Game Over	53
Gambar 4. 8 Pengujian Pertama.....	55
Gambar 4. 9 Pemilihan Jalur.....	56
Gambar 4. 10 Hubungan tiap point.....	57
Gambar 4. 11 Hasil Pengujian pertama	58
Gambar 4. 12 Waypoint pada jalur	59

DAFTAR TABEL

Tabel 2. 1 Daftar Gunung Aktif Indonesia	20
Tabel 2. 2 Matriks W_0	28
Tabel 2. 3 Matriks W_1	29
Tabel 2. 4 Matriks W_2	30
Tabel 2. 5 Matriks W_3	31
Tabel 2. 6 Matriks W_4	32
Tabel 3. 1 Desain Storyboard.....	36
Tabel 4. 1 Tabel Spesifikasi Perangkat Keras.....	48
Tabel 4. 2 Tabel Spesifikasi Perangkat Lunak.....	49
Tabel 4. 3 Karakter dan Objek.....	53
Tabel 4. 4 Pengujian dari titik awal ke titik akhir.....	60

ABSTRAK

Prayoga, Firhan Ade Achmad. 2019. *Implementasi pathfinding dengan algoritma floyd-warshall pada npc untuk game mitigasi gunung meletus*. Skripsi. Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Hani Nurhayati, M.T. (II) Fresy Nugroho, M.T.

Kata Kunci : *Pathfinding, Floyd-warshall, Mitigasi Gunung Meletus, Game Edukasi.*

Gunung Meletus merupakan peristiwa yang telah terjadi akibat endapan-endapan magma yang ada di dalam perut bumi dan didorong keluar oleh gas yang bertekanan tinggi. Saat ini Indonesia telah memiliki 127 gunung api yang aktif. Sebanyak 68 gunung diantaranya terus didapatkan pantauan yang intensif karena terdapat aktivitas magma didalamnya. Game edukasi merupakan sebuah permainan yang dibuat sebagai media pembelajaran untuk pemain sehingga dapat memberikan pemahaman yang lebih baik melalui game tersebut. Dari masalah gunung meletus yang sering terjadi di indonesia maka penting untuk dibuat sebuah game edukasi. Game edukasi yang dibuat lebih dikhususkan untuk siswa sekolah dasar. Algoritma Floyd-Warshall adalah algoritma untuk pencarian jalur terpendek pada teori graf, ini berkaitan dengan pengoptimuman suatu masalah sehingga dapat meminimumkan biaya serta meningkatkan efisiensi waktu. Algoritma ini diterapkan di game pada saat proses pencarian jalur menuju ke titik kumpul oleh NPC. Pengujian algoritma floyd-warshall pada NPC dilakukan sebanyak 30 kali dan dari jalur yang berbeda-beda. Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan implementasi algoritma floyd-warshall pada game mitigasi gunung meletus. Didapatkan hasil penelitian dari 30 pengujian dengan tingkat keberhasilan 100% pada game yang dibuat.

ABSTRACT

Prayoga, Firhan Ade Achmad. 2019. *Implementation of pathfinding with floyd-warshall algorithm on the npc for the volcano eruption mitigation game*. Essay. Department of Informatics Engineering, Faculty of Science and Technology, Islamic State University of Maulana Malik Ibrahim of Malang. Counselor : (I) Hani Nurhayati, M.T. (II) Fresy Nugroho, M.T.

Kata Kunci : *Pathfinding, Floyd-warshall*, volcanic eruption mitigation, Education Game.

Mount Eruption is an event that occurs due to magma sediments in the bowels of the earth and pushed out by high-pressure gas. A total of 68 mountains of them continue to get intensive monitoring because there is magma activity in it. Educational game is a game created as a learning media for players so that they can provide a better understanding through the game. From the eruption of mountains that often occur in Indonesia, it is important to make an educational game. Educational game made more specifically for elementary school students. The Floyd-Warshall algorithm is an algorithm for finding the shortest path in graph theory, this is related to optimizing a problem so that it can minimize costs and increase time efficiency. This algorithm is applied in the game during the process of finding a path to the gathering point by the NPC. Floyd-warshall algorithm testing on NPC was done 30 times and from different paths. This test was conducted to find out the level of success of the implementation of the floyd-warshall algorithm in the volcano eruption mitigation game. The research results obtained from 30 times tested with a 100% success rate on the game created.

الملخص

فاريوفا ، فرحان أدي أحمد. ٢٠١٩. تنفيذ العثور على الطريق مع خوارزمية فلويد وارسال على NPC لعبة الإغاثة بركان. قسم هندسة المعلوماتية لكلية العلوم والتكنولوجيا في جامعة مولانا مالك إبراهيم الإسلامية الحكومية بمالانق. المشرف : (١) حاني نورحاياتي، الماجستير. (٢) د. فريسي نوغراها، الماجستير.

الكلمات الرئيسية : العثور على الطريق, فلويد وارسال, التخفيف من ثوران البركان ، لعبة التعليم.

ثار بركان هو حدث يحدث بسبب رواسب الصهارة في أحشاء الأرض ويتم دفعه بالغاز عالي الضغط. ما مجموعه ثمانية وستون جبال منهم لا تزال تحصل على رصد مكثف لأن هناك نشاط الصهارة في ذلك. اللعبة التعليمية هي لعبة تم إنشاؤها كوسيلة تعليمية للاعبين حتى يتمكنوا من توفير فهم أفضل من خلال اللعبة. من ثوران الجبال التي تحدث غالبًا في إندونيسيا ، من المهم إنشاء لعبة تعليمية. لعبة تعليمية جعلت أكثر تحديدا لطلاب المدارس الابتدائية. تعد خوارزمية فلويد وارسال خوارزمية لإيجاد أقصر مسار في نظرية الرسوم البيانية ، وهذا يرتبط بتحسين المشكلة بحيث يمكن تقليل التكاليف وزيادة كفاءة الوقت. يتم تطبيق هذه الخوارزمية في اللعبة أثناء عملية البحث عن مسار إلى نقطة التجمع بواسطة NPC. تم إجراء اختبار خوارزمية فلويد وارسال على NPC ثلاثين مرة ومن مسارات مختلفة. تم إجراء هذا الاختبار لمعرفة مستوى نجاح تنفيذ خوارزمية فلويد وارسال في لعبة تخفيف ثوران البركان. نتائج البحث التي تم الحصول عليها من ثلاثين مرة تم اختبارها بمعدل نجاح مائة في اللعبة التي تم إنشاؤها.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara kepulauan terbesar di dunia dengan jumlah pulau yang dimiliki mencapai 17.504 baik itu pulau yang besar dan kecil (Prasetya, 2017). Terdapat sekitar 6100 pulau diantaranya belum berpenghuni dan menyebar disekitar khatulistiwa. Oleh sebab itu lah Indonesia menjadi negara dengan iklim tropis karena berada pada garis khatulistiwa. Keseluruhan luas wilayah negara Indonesia melebihi dua juta kilo meter. Diantara puluhan ribu pulau di Indonesia terdapat 5 pulau terbesar, antara lain Pulau Jawa, Sumatra, Papua, Kalimantan dan juga Sulawesi. Posisi Indonesia tercatat terletak pada koordinat $6^{\circ}\text{LU} - 11^{\circ}08'\text{LS}$ dan dari $95^{\circ}\text{BT} - 141^{\circ}45'\text{BT}$ serta terletak di antara dua benua yaitu benua Australia atau Oseania dan benua Asia. Selain itu secara geografis Indonesia terletak di antara dua samudera, yaitu Samudera Hindia dan Samudera Pasifik dimana kedua samudera tersebut dilewati oleh garis khatulistiwa (Zakky, 2018).

Garis Khatulistiwa merupakan suatu garis khayal membentang dan memutar bumi serta memberikan tanda bahwa bumi terdapat 2 belahan yaitu pada bagian utara yang ditandai dengan kutub utara dan bagian selatan pada kutub selatan. Garis khatulistiwa yang juga biasa di sebut garis equator ini berada pada lintang nol derajat bumi dengan memiliki panjang lebih dari 40,065 km. diantaranya 78,70% nya adalah air serta 21,30% adalah daratan. Sebanyak hampir 14 negara berada pada bentangan garis khayal ini, dan satu dari beberapa negara tersebut adalah Indonesia. Wilayah yang terlintasi garis khatulistiwa ini dikenal sebagai wilayah rawan bencana, baik bencana dari alam yang terjadi secara alamiah dan sudah

mustahil untuk di hindari ataupun bencana yang beasal atau akibat dari faktor manusia atau yang sebenarnya bisa dikatakan dapat antisipasi. Pada segi geografisnya, Indonesia memiliki letak atau posisi yang memang tergolong berpotensi atau rawan terhadap bencana, khususnya bencana gempa bumi (Maarif, A. Kinseng, & Pramono, 2016). Serta Indonesia terdapat pada jalur cincin api pasifik ataupun bisa dikatakan jalur ring of fire dengan kata lain wilayah ini terkenal sebagai jalur gempa teraktif dan rawan terjadi adanya gunung berapi yang erupsi atau meletus. Maka dari hal tersebut, Indonesia menjadi salahsatu negara yang memiliki tingkat kebencanaan yang tergolong besar dan khususnya pada bencana gunung meletus, gempa bumi dan tsunami. Bencana tersebut sering terjadi di beberapa daerah di Indonesia yaitu pada provinsi-provinsi yang dilalui jalur khatulistiwa dan jalur cincin api pasifik yaitu, Aceh, Sumatera Barat, Sumatera Utara, Banten, Seluruh Jawa, Sulawesi, Bali Nusa Tenggara dan sekitarnya dan Papua. Daerah-daerah tersebut yang termasuk dalam wilayah dengan potensi bencana tinggi yang salah satunya adalah bencana gunung erupsi atau meletus. Gunung meletus atau erupsi gunung sendiri adalah fenomena alam yang terjadi karena endapan - endapan dari magma-magma yang telah ada di dalamnya perut bumi dan akan dikeluarkan karena ada pacuan gas-gas yang ber tekanan besar. Saat ini Indonesia telah memiliki 127 gunung api yang aktif (Pratama, Laila N, & Putra W, 2014). Sebanyak 68 gunung gunung diantaranya terus didapatkan pantauan yang intensif karena terdapat aktivitas magma didalamnya. Magma merupakan cairan panas atau pijar dan berada pada bagian perut dalam bumi yang memiliki suhu yang tergolong ekstra panas. Kira-kira panasnya mencapai 1000°C. Cairan pijar/magma dikeluarkan dari bagian dalam bumi dapat menembus suhu 700 -1200°C. Dari

letusan atau erupsi gunung berapi akan membawa abu serta batu dan akan menyemburkan hingga radius melebihi 17km atau tidak bisa terprediksi jauh letusnya. Untuk cairan Lava nya dapat meluber pada permukaan hingga radius 91km atau lebih. Namun status gunung berapi tidak selalu akan meletus terus menerus, gunung api yang mengalami erupsi berulang kali atau sering dikategorikan dengan gunung api yang aktif. Maka dari itu, dengan adanya kategori aktif tersebut pemerintah akan lebih mudah dalam mewaspadaai karena dampaknya yang dapat merusak lingkungan serta membunuh makhluk hidup yang ada disekitarnya.

Hal ini membuat penelitian ini penting untuk dilakukan. Salah satunya dengan membuat sebuah *game* edukasi. *Game* edukasi atau *Education Game* merupakan sebuah permainan yang dibuat sebagai media pembelajaran untuk *player* atau pemain sehingga dapat memberikan pemahaman yang lebih baik melalui *game* tersebut. *Game* edukasi pada penelitian ini memuat tentang mitigasi bencana gunung meletus. Membangun materi berbentuk *game* seputar tata cara atau langkah untuk meminimalisir kerugian atau dampak dari gunung meletus

Game edukasi 3D berbasis desktop yang akan dibuat didalam penelitian ini akan menggunakan algoritma *Floyd-warshall* sebagai penentu jarak terdekat yang diterapkan didalam *Non player character (NPC)*. Dalam pencarian jalan yang terpendek, algoritma *Floyd-warshall* terbukti lebih unggul dari sisi kesederhanaan dan solusi *optimum* dibandingkan dengan algoritma lain seperti Algoritma *Dijkstra* (Muzdalifah, Oktafianto, & Mustika, 2018). Penelitian ini bertujuan untuk mengetahui tingkat keberhasilan algoritma *floyd-warshall* serta meningkatkan kemauan dan minat anak tentang pembelajaran mitigasi bencana gunung meletus

sehingga dengan penelitian ini dapat mengurangi resiko dari bencana gunung meletus tersebut.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang dipaparkan di atas identifikasi masalah yang dapat dirumuskan adalah seberapa berhasil implementasi *pathfinding* dengan algoritma *Floyd-warshall* pada *NPC* untuk *game* mitigasi gunung meletus

1.3 Tujuan Penelitian

Mengukur tingkat keberhasilan implementasi *pathfinding* dengan algoritma *Floyd-warshall* pada *NPC* untuk *game* mitigasi gunung meletus.

1.4 Batasan Masalah

Beberapa batasan masalah terkait penelitian ini antara lain:

- a. *Game* ini berbasis 3D
- b. *Game* level 1 dan level 2
- c. *Game* ini berbasis *desktop*
- d. *Game* ini ditujukan untuk anak sekolah dasar (SD)

1.5 Manfaat Penelitian

Manfaat dari penelitian yang dibuat diantaranya adalah:

1. Memberikan edukasi mitigasi bencana gunung meletus kepada anak SD dengan memainkan *game* seperti yang biasa mereka mainkan.
2. Bagi peneliti, diharapkan penelitian ini dapat dijadikan acuan maupun referensi pada penelitian selanjutnya.

1.6 Sistematika Penelitian

Penerapan *pathfinding* dengan algoritma *Floyd-warshall* untuk pencarian korban bencana pada *game* mitigasi Gunung Meletus ini tersusun dengan sistematika adalah sebagai berikut.

BAB I Pendahuluan

Pendahuluan pada penelitian ini berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Manfaat Penelitian, dan Sistematika Penulisan.

BAB II Landasan Teori

Landasan teori pada penelitian ini berisi tinjauan pustaka yang digunakan sebagai referensi dalam pembuatan *game* tersebut.

BAB III Metodologi Penelitian

Metodologi penelitian pada penelitian ini berisi tentang seputar perancangan *game* dan metode yang akan dibuat.

BAB IV Hasil dan Pembahasan

Hasil dan Pembahasan pada penelitian ini berisi tentang hasil analisis dari *game* yang dibuat dengan penjelasan atau pembahasan dalam bentuk teks maupun gambar, serta penerapan algoritma pada karakter yang dibuat.

BAB V Penutup

Penutup pada penelitian ini berisi tentang kesimpulan dan saran dari penelitian yang telah dibuat.

BAB 2

LANDASAN TEORI

2.1 Penelitian Terkait

Beberapa penelitian yang terkait dengan penelitian ini adalah sebagai berikut.

1. Pada penelitian pertama yang berjudul “Simulasi Jaringan Jalan di Kota Semarang Berbasis Algoritma *Floyd-warshall* Untuk Menangani Masalah Lintasan Terpendek” menjelaskan tentang masalah pencarian lintasan terpendek pada teori graf berkaitan dengan masalah pengoptimuman, yaitu meminimumkan biaya dan efisiensi waktu. Algoritma *Floyd-warshall* merupakan bagian dari program dinamik yang dapat mencari semua lintasan terpendek masing-masing antar tiap kemungkinan pasang tempat berbeda dan juga sangat efektif digunakan dalam mengatasi masalah lintasan optimum. Hasil dari penelitian ini menjelaskan bahwa jika dilakukan perbandingan antara hasil solusi pencarian lintasan terpendek menggunakan simulasi dengan cara manual ternyata mempunyai hasil yang sama. Jadi, simulasi algoritma *Floyd-warshall* dalam menangani masalah lintasan terpendek pada graf dapat diimplementasikan dengan berbagai macam bentuk graf yang dapat direpresentasikan ke dalam simulasi (Suyitno, 2016).
2. Pada penelitian kedua yang berjudul “Penerapan Algoritma *Dijkstra* Dan *Floyd-warshall* Untuk Menentukan Rute Terpendek Tempat Wisata Di Batang” menjelaskan tentang Algoritma *Floyd-warshall* yang bekerja berdasarkan formulasi *dynamic programming*, di mana setiap langkahnya akan menghitung bobot terkecil dari semua rute yang menghubungkan

sebuah pasang titik, dan melakukannya sekaligus untuk semua pasangan titik. Dengan kata lain pada saat perhitungan rute terpendek, terlebih dahulu menghitung semua kemungkinan rute yang akan dilalui. Kemudian baru mencari rute terpendek dengan cara membandingkan tiap pasang rute dan dicari rute yang lebih pendek. Hasil dari penelitian ini menjelaskan berdasarkan rute terpendek dari penerapan algoritma *Floyd-warshall* dalam menentukan rute terpendek menuju ke tempat wisata batang adalah 27 rute terpendek dimana 25 rute adalah sama dan terdapat 2 rute yang berbeda (Marlina & Suyitno, 2017).

3. Pada penelitian ketiga yang berjudul “Model Jaringan Distribusi Beras Optimal Menggunakan Algoritma *Floyd-warshall*” menjelaskan tentang Algoritma *Floyd-warshall* yang terbukti lebih unggul dibandingkan dengan algoritma Dijkstra, karena algoritma *Floyd-warshall* memiliki *recovery time* yang lebih cepat. Algoritma *Floyd-warshall* dengan pemrograman yang dinamis juga lebih menjamin penemuan solusi *optimum* untuk kasus penentuan rute terpendek. Algoritma *Floyd-warshall* banyak diterapkan pada masalah penentuan rute terpendek, seperti masalah tentang pengiriman barang, evakuasi korban bencana alam, dan lain lain. Hasil dari penelitian ini menjelaskan bahwa rute terpendek distribusi beras dari 12 petani ke konsumen akhir telah didapatkan secara akurat dan cukup hanya dilakukan dengan satu kali running program. Akan tetapi, algoritma ini tidak dapat mengidentifikasi solusi alternatif secara langsung (Muzdalifah et al., 2018).
4. Pada penelitian keempat yang berjudul “*Floyd-warshall algorithm to determine the shortest path based on android*” menjelaskan bahwa

algoritma *Floyd-warshall* adalah salah satu varian dari pemrograman dinamis yang memecahkan masalah dengan melihat solusi yang akan diperoleh sebagai keputusan yang saling berkaitan. Ini artinya sebuah solusi terbentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan terdapat lebih dari satu solusi. Algoritma *Floyd-warshall* membandingkan setiap variasi rute dengan jarak dari setiap titik ke titik tujuan dan mengakumulasi jarak antara *node* (titik persimpangan) yang dilewati. Hasil dari penelitian ini menjelaskan bahwa hasil pengujian manual yang telah dilakukan sesuai dengan *ouput* dari program pengambilan donasi dengan menggunakan algoritma *Floyd-warshall* dan menghasilkan rute penjemputan donasi ke 20 marka donor yang tersebar di wilayah Samarinda dengan total jarak 29.314 meter mengikuti jalur rute yang telah ditunjukkan (Ramadiani, Bukhori, Azainil, & Dengen, 2018).

2.2 Game

Game atau permainan, menurut Kamus Besar Bahasa Indonesia atau (KBBI) adalah sesuatu yang di gunakan untuk bermain. *Game* merupakan sebuah permainan atau sebuah media hiburan yangmana untuk melakukan atau memainkan *game* tersebut harus terdapat seorang *player* atau pemain. *Player* atau pemain ini bisa dari bermacam macam kalangan, dari mulai yang masih kecil sampai orang orang remaja maupun dewasa. Dalam sebuah *game* atau permainan selalu memiliki hasil akhir atau biasa disebut dengan skor. Terdapat tiga kemungkinan hasil akhir dalam sebuah *game* atau permainan, yaitu kalah, seri dan menang. Konflik dan peraturan juga merupakan bagian dari sebuah *game*. Untuk menyelesaikan *game*,

player akan diberi beberapa konflik untuk dilewati dengan mengikuti aturan-aturan yang telah ditentukan dalam *game* tersebut.

Semakin berkembangnya teknologi, *game* menjadi media permainan yang semakin praktis. Kini *game* dapat dimainkan melalui sebuah Komputer, Laptop, maupun *Smartphone*. *Game* dapat membantu pola pikir orang untuk memusatkan dan mempertahankan perhatian pada berbagai situasi. Selain itu *game* juga dapat membantu dalam meningkatkan *visuospatial*, yaitu kemampuan untuk menempatkan suatu atau sebuah benda, objek, atau gambar di dalam sebuah ruangan.

Pengertian dari *genre game*, merupakan klasifikasi *game* yang di dasari interaksi *player* atau pemain nya. Yang menjadi ukuran dari klasifikasi *genre* ini adalah visualisasi. Sehingga variasi jenis format dan *genre game* akan lebih banyak. *genre game* di bagi menjadi beberapa jenis, diantaranya yaitu:

a. *Action*

Merupakan jenis *game* atau permainan yang banyak membutuhkan kecepatan tangan, refleks, akurasi, kecepatan mata, dan juga timing dalam menghadapi suatu rintangan. Permainan jenis ini biasa dihubungkan dengan pertarungan fisik. Contoh *game* ini adalah *Naruto Shippuden*, *Bleach*, *Mortal Combat*.



Gambar 2. 1 Game Action

Sumber : <https://www.gamestop.com/videogames/playstation-4/games/products/naruto-shippuden-ultimate-ninja-storm-4/10119794.html>

b. *Shooter*

Merupakan jenis *game* yang hampir sama dengan jenis *game action*. Namun banyak pemain atau *player* yang menganggap *game shooter* merupakan *genre* yang berdiri sendiri dikarenakan *game* jenis ini lebih menitikberatkan pada tingkat akurasi pemain dalam menembak musuh. Contoh *game* ini adalah *PUBG*, *Free Fire*, *Point blank*.

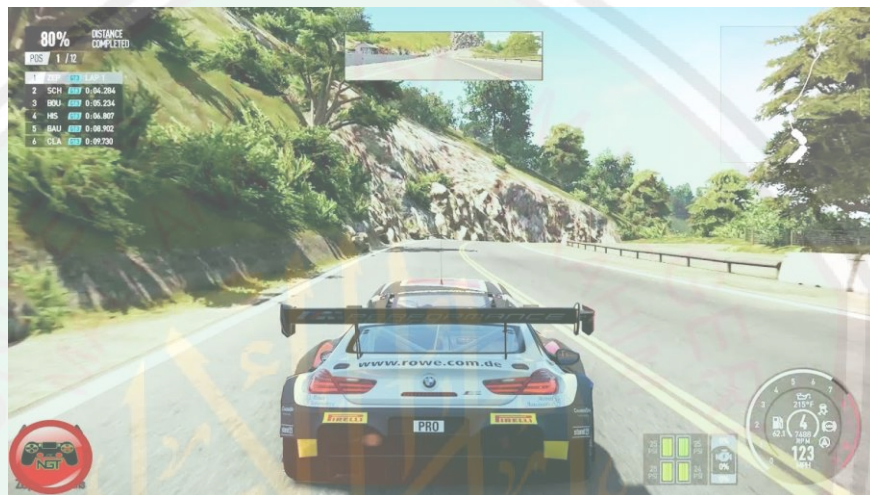


Gambar 2. 2 Game Shooter

Sumber : <https://apkpure.com/gun-shot-fire-war/com.idoing.linefirefree>

c. *Racing*

Merupakan jenis *game* yang menuntut pemain untuk terampil dalam mengemudikan sebuah kendaraan yang dimainkan. Tujuan dari *game* ini yaitu untuk melewati musuh-musuhnya dan menjadi yang paling depan. Jenis *game* ini biasanya menggunakan mobil atau motor sebagai media kompetisinya. Contoh *game* ini adalah *NFS*, *Ashpalt*, *Burnout*.



Gambar 2. 3 Game Racing

Sumber : https://i.ytimg.com/vi/7xD2uvazG_8/maxresdefault.jpg

d. *Sport*

Merupakan jenis permainan yang akan mengharuskan para *player* untuk melakukan kompetisi atau permainan olah raga dengan secara tampilan *virtual*. Semakin berkembangnya *game sport* maka telah bermunculan seperti *game* badminton, basket, voli, atau sepak bola. Contoh dari macam *game* ini diantaranya adalah *FIFA*, *Winningeleven*, dan *PES*.



Gambar 2. 4 Game Sport

Sumber : <http://testphoneapps.com/top-10-best-sports-games-for-android/>

e. *Adventure*

Merupakan jenis *game* yang biasanya menggabungkan unsur *action* dan *adventure* di dalamnya. *Game* jenis ini biasanya memiliki banyak rintangan dan berjangka panjang. *Player* akan disuruh untuk menyelesaikan misi dengan menggunakan *item-item* tertentu. Dalam *game* ini terdapat banyak misi atau chapter yang harus diselesaikan. Contoh *game* ini adalah *Assasins Creed*, *Firewatch*, *Disney's Hercules*.



Gambar 2. 5 Game Adventure

Sumber : <https://www.gamesradar.com/pokemon-sword-shield-review/>

f. *Strategy*

Merupakan jenis *game* yang membutuhkan kemampuan otak dalam memikirkan strategi untuk memenangkan sebuah permainan. *Game* jenis ini melibatkan logika, taktik, dan strategi. Contoh dari *game* ini diantaranya adalah *Clash of Clan*, *Clash Royale*, *Chess*.



Gambar 2. 6 *Game Strategy*

Sumber : <http://legoninjagoonline.com/strategy-games-tips/>

g. *RPG*

Merupakan jenis *game* yang memainkan tokoh atau pemeran utama dalam permainannya. Dalam beberapa keadaan tokoh atau pemeran utama dapat berganti sesuai dengan jalan cerita *game* tersebut. *Game RPG* terdapat dua pilihan, yaitu *RPG* dengan animasi 2D dan *RPG* dengan animasi 3D. Contoh *game* ini adalah *Final fantasy*, *Suikoden*, *Ragnarok*.



Gambar 2. 7 Game RPG

Sumber : <https://keepo.me/techno/5-game-rpg-rekomendasi-terbaik-di-tahun-2019/>

2.3 Gunung Meletus

Gunung meletus atau berapi, merupakan lobang kepundan, yang juga bisa dikatakan rekahan yang berada pada dalam kerak pada bumi. Merupakan jalur menyemburkan cairan pijar atau magma, gas, dan material lain keluar dari perut bumi. Meterial-material semburan letusan dari perut bumi berbeda-beda, yang biasanya berbentuk kerucut-terpancung. Letusan gunung berapi dapat dikategorikan pada 4 sumber letusan atau erupsi antara lain:

- a. Erupsi Pusat, yaitu letusan yang melewati kawah utamanya
- b. Erupsi Samping, yaitu erupsi atau letusan keluar melalui bagian lereng gunung.
- c. Erupsi Celah, yaitu letusan yang timbul dari bagian retakkan atau sesar. Retakan-retakan ini nantinya bisa me manjang hingga puluhan kilo meter
- d. Erupsi Eksentrik, yaitu letusan samping, namun nantinya magma akan dikeluarkan tidak berasal dari bagian kepundan utama yang me nyimpang sebelah samping, namun akan langsung berasal dari bagian dapur magma.

Pembentukan gunung api sudah lama terjadi bahkan dari jutaan tahun sudah muncul dan hingga saat ini menjadi besar karena perkembangan waktu. Ilmu yang membahas mengenai gunung berapi awalnya terbentuk berdasarkan kehidupan manusia sendiri yaitu pada perilakunya. Selain itu juga dilihat dari konteks manusia pada jaman purbakala yang sudah menggambarkan bahwa kehidupan mereka tidak jauh dengan adanya gunung berapi. Perihal fenomena tersebut dapat dibuktikan dengan telah ditemukan jejak-jejak sejarah dalam bentuk fosil manusia terletak didalam endapan-vulkanik dan mayoritas bentuk fosilnya telah diidentifikasi berasal dari sekitar Indonesia hingga Afrika yang berwujud kumpulan tulang-belulang manusia berada didalam endapan-endapan vulkanic sesuai Survey Vulcanological of Indonesia. Pengertian atau makna lain untuk gunung api dapat dijabarkan bahwa, gunung berapi adalah proses keluarnya lava yang memiliki suhu tinggi melalui sebuah sistem saluran fluida yang dapat memanjang dari kedalaman 10km bahkan lebih pada bagian bawah permukaan bumi hingga menuju atas permukaan bumi, yang didalamnya berupa endapan-endapan semua campuran material yang pada erupsi dikeluarkan disaat terjadi ledakan/letusan pada gunung berapi.

Busur vulkanik merupakan serangkaian gunung berapi yang muncul pada lempeng subduksi, serta mempunyai letak berwujudkan busur saat dipantau atau diperhatikan dari ketinggian. Pada gunung berapi lepas pantai dapat membentuk gugusan pulau, lalu akan melahirkan busur kepulauan-vulkanik. Berdasarkan hasil penelitian *Vulcanological Survey of Indonesia*, telah dijelaskan terdapat empat busur gunung api antara lain adalah:

- a. Busur Tengah Benua, dengan adanya pemekaran pada kerak benua maka terbentuklah busur ini.
- b. Busur Tepi Benua, yaitu adanya subduksi atau penunjaman kerak samudera kearah kerak-benua.
- c. Busur Tengah Samudra, yaitu terbentuk dengan adanya pemekaran kerak pada Samudra.
- d. Busur Dasar Samudra, yaitu terbentuk karena penempisan kerak Samudra yang berasal dari banyaknya lelehan magma basa.

Wujud gunung berapi sangatlah variative dari masa kemasa. Status gunung api tidaklah mutlak namun sewaktu-waktu atau dalam periode tertentu dapat berubah, mulai dari aktif, paruh aktif, lalu jadi pasif, hingga pada riwayatnya bisa menjadi status gunung mati/tidak aktif. Namun selang waktu yang dapat merubah status gunung berapi ini terolong sangat lama yaitu berkisar lebih dari 6000 tahun lamanya, baik perubahan status aktif ke mati atau sebaliknya. Maka, sangatlah sulit untuk menyebut kondisi yang sebenarnya dari suatu status gunung api, gunung tersebut aktif atau mati.

Pada Negara Indonesia khususnya bagian kepulauan Jawa dan sekitaran Sumatra, adanya gunung berapi merupakan efek dari adanya tumbukan kerak Samudra Hindia yang langsung bertemu dengan kerak pada Benua Asia. Pertemuan kerak yang mengakibatkan tumbukan antara dua kerak di daerah Sumatra lebih keras dan menghasilkan bagian akresi timbul keluar dari laut dan membentuk gugusan kepulauan dipesisir wilayah Sumatra seperti Kepulauan Mentawai dan Nias. Untuk efek dari letusan gunung berapi, telah diindikasikan ada empat belas efek bahaya letusan. Bahaya letusan gunung berapi yang ini mampu mempengaruhi secara

langsung (primer) dan tidak langsung (sekunder) yang nantinya akan berdampak langsung pada dunia manusia atau masyarakat dan dapat dikatakan menimbulkan bencana. Berikut merupakan bahaya yang dikategorikan langsung dari adanya erupsi gunung berapi antara lain yaitu :

a. Leleran lava

Leleran lava atau molten lava, yaitu berbentuk cairan lava yang kental/pekat dan bersuhu tinggi yang bisa menghancurkan semua objek yang dilewatinya. Aliran lava dapat dihitung kecepatannya tergantung dari tinggi rendahnya kepekatan dan keenceran magmanya, sehingga makin rendah kepekatan, berakibat akan lebih panjang lagi radiusnya aliran magma tersebut. Panas pada tererupsi adalah berada pada kisaran 8000 – 12000 derajat C. di Indonesia biasanya, leleran lava yang meluber dari gunung api, material magmanya berada pada kisaran tengah yang menjadikan kecepatannya tergolong pelan oleh karena itu antisipasi masih dapat dilakukan untuk mengurangi banyaknya korban.

b. Aliran Piroklastik (awan panas)

Awan panas ini muncul karena adanya runtuhannya pada tiang asap erupsi plinian, untuk erupsinya hanya pada satu arah secara langsung, guguran dari lidahlava atau kubahlava, dan aliran dipermukaan tanah (*surge*). Adanya gravitasi begitu mempengaruhi pergerakan awan panas ini dan akan menuju lembah bagian bawah atau langsung ke daerah yang lebih rendah yaitu lembah. Pergerakan yang tinggi dari awan panas ini dipengaruhi oleh dorongan gas yang berasal dari magma panas ataupun lava panas atau dari udara pada saat mengalir. Pergerakan awan panas bisa mencapai antara 150,0 sampai dengan 250,0

kilometer per jam dan radius alirannya dapat melebihi puluhan kilometer walau pun pergerakannya berada di atas lautan.

c. Jatuhan piroklastik

Jatuhan piroklastik itu diakibatkan oleh erupsi yang mana melahirkan tiang-tiang asap serta tergolong tinggi. Material dalam bentuk abu letusan akan menyebar tergantung pada pergerakan angin dan akan jatuh ke permukaan bumi pada saat energinya telah hilang. Bisa dikatakan bahwa ini adalah fenomena hujan abu, namun hal ini tidak dikategorikan bahaya langsung terhadap manusia namun, endapan – endapan dari adanya hujan abu akan berakibat pada tumbuhan atau pohon yang daunnya akan mati serta rontok, merugikan para petani dan merusak perumahan khususnya pada atap. Hujan abu ini akan berakibat pada kondisi cuaca yang bisa menutupi sinar matahari dan pastinya dunia penerbangan akan terganggu secara langsung.

d. Lahar letusan

Lahar letusan, biasanya akan timbul bilamana terdapat danau dalam kawah gunung. Jika jumlah air pada danau tergolong banyak, hal ini merupakan bahaya langsung karena saat terjadi ledakan atau erupsi air dalam danau akan menjadi lumpuh dengan suhu panas.

e. Gas vulkanik beracun

Umumnya gas beracun yang ada di gunung berapi aktif seperti CO_2 , CO , HCN , SO_2 , H_2S , dan lainlain. Dan apabila konsentrasinya telah melewati batas normal dapat memusnahkan makhluk hidup.

Sedangkan bahaya-bahaya sekunder dari kejadian setelah ataupun saat gunung berapi aktif antara lain:

a. Longsoran vulkanik

Longsoran vulkanik dipicu dari erupsi gunung berapi, ledakan uap air didalam tubuh gunung api sehingga dapat merubah struktur menjadi rapuh, atau terdampak dari adanya getaran gempa yang tergolong kuat. Longsoran vulkanik ini secara umum kecil peluangnya untuk muncul di gunung api oleh karena itu di dalam peta atau maps wilayah potensi bencana tinggi jarang tercantum bahaya dari Longsoran vulkanik.

b. Lahar hujan

Jika hujan tiba dan mengangkut sedimen material setelah dari efek letusan gunungberapi yang di endapkan dipuncak maupun lereng, maka bahaya lahar ini akan muncul karena hujan akan membawa material itu. Lumpur yang kental/pekat adalah lahar yang dimaksudkan dan lumpur ini berisi berbagai jenis material seperti sebuah batu dan memiliki ukuran yang sangat besar yang nantinya dapat terapung di permukaan sungai. Lahar ini sangat merusak dan bisa merubah topografi pada aliran sungai yang dilaluinya serta mampu merusak fasilitas dan infrastruktur sepanjang aliran lahar ini.

c. Banjir bandang

Dengan adanya sebaran material erupsi dan bekas dari longsoran material-material vulkanik lama disekitaran lereng gunung api dan adanya curah hujan yang lebat akan menimbulkan bajir paska erupsi yaitu banjir bandang. Aliran Lumpur tergolong sudah cair dan tidak kental ataupun berbentuk seperti lahar, akan tetapi tetap menjadi ancaman untuk masyarakat yang berada disekitan lairan banjir ini.

Berikut merupakan daftar gunung api aktif di Indonesia berdasarkan data dari aplikasi Magma Indonesia milik Pusat Vulkanologi dan Mitigasi Bencana Geologi Kementerian Energi dan Sumber Daya Mineral (ESDM) (Vebri, 2018).

Tabel 2. 1 Daftar Gunung Aktif Indonesia

No	Nama Gunung	Letak	Tinggi
1	Gunung Agung	Karangasem, Bali	3142 m
2	Gunung Ambang	Bolaang, Mongondow, Sulawesi Utara	1795 m
3	Gunung Anak Krakatau	lampung selatan, Lampung	338 m
4	Gunung Anak Ranakah	Manggarai, NTT	2350 m
5	Gunung Arjuno Welirang	Malang, Mojokerto, Pasuruan, Jatim	3339 m
6	Gunung Awu	Kepulauan Sangihe, Sulawesi Utara	1320 m
7	Gunung Banda Api	Maluku Tengah, Maluku	641 m
8	Gunung Batur	Bangli, Bali	1717 m
9	Gunung Batutara	Lembata, NTT	748 m
10	Gunung Bromo	Probolinggo, Jatim	2329 m
11	Gunung Bur Ni Telong	Aceh Tengah, Aceh	2623 m
12	Gunung Ciremai	Cirebon, Kuningan, Majalengka, Jabar	3078 m
13	Gunung Colo	Tojo Una-Una, Sulawesi Tengah	404 m
14	Gunung Dempo	Lahat, Sumatera Selatan	3173 m
15	Gunung Dieng	Banjarnegara, Wonosobo, Batang, Jateng	2565 m
16	Gunung Dukono	Halmahera Utara, Maluku Utara	1229 m
17	Gunung Ebulobo	Nagekeo, NTT	2124 m
18	Gunung Egon	Sikka, NTT	1661 m
19	Gunung Galunggung	Tasikmalaya, Garut, Jabar	2168 m
20	Gunung Gamalama	Ternate, Maluku Utara	1715 m
21	Gunung Gamkonora	Halmahera Barat, Maluku Utara	1635 m
22	Gunung Gede	Cianjur, Bogor, Sukabumi, Jawa Barat	2958 m
23	Gunung Guntur	Garut, Jawa Barat	2249 m
24	Gunung Ibu	Halmahera Barat, Maluku Utara	1325 m
25	Gunung Ijen	Banyuwangi, Bondowoso, Jatim	2386 m
26	Gunung Ile Werung	Lembata, NTT	1018 m
27	Gunung Ili Boleng	Flores Timur, NTT	1659 m
28	Gunung Ili Lewotolok	Lembata, NTT	1423 m
29	Gunung Inielika	Ngada, NTT	1559 m
30	Gunung Inierie	Ngada, NTT	2245 m
31	Gunung Iya	Ende, NTT	637 m
32	Gunung Kaba	Rejang Lebong, Bengkulu	1952 m
33	Gunung Karangetang	Siau Tagulandang Biaro, Sulawesi Utara	1784 m
34	Gunung Kelimutu	Ende, NTT	1639 m

No	Nama Gunung	Letak	Tinggi
35	Gunung Kelud	Kediri, Blitar, Malang, Jatim	1731 m
36	Gunung Kerinci	Kerinci, Solok Selatan, Jambi	3805 m
37	Gunung Kie Besi	Halmahera Selatan, Maluku Utara	1357 m
38	Gunung Lamongan	Lumajang, Jatim	1651 m
39	Gunung Lereboleng	Flores Timur, NTT	1095 m
40	Gunung Lewotobi Laki-Laki	Flores Timur, NTT	1584 m
41	Gunung Lewotobi Perempuan	Flores Timur, NTT	1703 m
42	Gunung Lokon	Tomohon, Sulawesi Utara	1580 m
43	Gunung Mahuwu	Tomohon, Sulawesi Utara	1299 m
44	Gunung Marapi	Agam, Batusangkar, Sumatera Barat	1891 m
45	Gunung Merapi	Magelang, Boyolali, Klaten, Jawa Tengah	2968 m
46	Gunung Papandayan	Garut, Jawa Barat	2665 m
47	Gunung Peut Segue	Pidie Meriah, DI Aceh	2801 m
48	Gunung Raung	Banyuwangi, Bondowoso, Jatim	3332 m
49	Gunung Rinjani	Lombok Timur, NTB	3726 m
50	Gunung Rokatenda	Sikka, NTT	875 m
51	Gunung Ruang	Siau Tagulandang Biaro, Sulawesi Utara	725 m
52	Gunung Salak	Sukabumi, Bogor, Jabar	2211 m
53	Gunung Sangeangapi	Bima, NTB	1949 m
54	Gunung Semeru	Lumajang, Malang, Jatim	3676 m
55	Gunung Seulawah Agam	Aceh Besar, DI Aceh	1810 m
56	Gunung Sinabung	Karo, Sumatera Utara	2460 m
57	Gunung Sirung	Alor, NTT	862 m
58	Gunung Slamet	Pemalang, Banyumas, Brebes, Tegal, Purbalingga	3428 m
59	Gunung Soputan	Minahasa Tenggara, Sulawesi Utara	1809 m
60	Gunung Sorikmarapi	Mandailing Natal, Sumut	2145 m
61	Gunung Sumbing	Magelang, Temanggung, Wonosobo	
62	Gunung Sindoro	Temanggung, Wonosobo, Jateng	3160 m
63	Gunung Talang	Solok, Sumatera Barat	2597 m
64	Gunung Tamboro	Dompu, Bima, Nusa Tenggara Barat	2850 m
65	Gunung Tandikat	Padang Pariaman, Agam, Sumbar	2438 m
66	Gunung Tangkoko	Bitung, Sulawesi Utara	1334 m
67	Gunung Tangkuban Parahu	Subang, Bandung, Jawa Barat	2084 m
68	Gunung Wurlali	Maluku Tenggara Barat, Maluku	868 m

Dari tabel diatas terlihat bahwa banyak gunung berapi yang tersebar dari aceh sampai NTT.

2.4 Unity

Unity Technologies, di bangun pada kisaran tahun 2004, oleh David Helgasson (CEO), Nicholass Francis (CCO), dan Joachim Ante (CTO) di Copenhagen, Denmark. Mereka berencana untuk membuatnya lebih terjangkau. Sehingga pada akhirnya *Unity Technologies* tersebut mendapat suatu bantuan dana-dana dari *WestSummit Capitals, iGlobe Partner, & Sequoia Capitals*.

Pada kisaran tahun 2009, pada akhirnya *Unity* telah meluncurkan berbagai produk-produk mereka dengan gratis. Setelah digratiskannya produk-produk mereka, jumlah dari developers yang mendaftarpun meningkat dengan cepat. Pada tahun 2012 bulan april, *Unity* pun telah mencapai popularitas dengan sangat tinggi dengan lebihdari 1juta developers yang telah menggunakan produk-produk mereka. Dengan berbagai kemampuan, dan pengetahuan mereka yang telah dapat melihat berbagai maca peluang & perubahan, pada akhirnya *Unity* pun telah menjadi sebagai sebuah *game engine* yang termurah dan paling banyak di gunakan oleh para *developers game*.

Beberapa fitur dari *Unity* adalah sebagai berikut.

1. *Rendering*

Graphic engine yang dipakai adalah *OpenGL (Windows, Mac, PS3, Linux), Direct3D (Xbox 360, Windows), proprietary, APIs (Wii), dan OpenGL ES (Android dan iOS)*. Ada jua kemampuan-kemampuan dalam *reflection mapping, bump mapping, screen space ambient occlusion (SSAO), parallax mapping, render-to-texture, full-screen post-processing effects, dan dynamic shadows using shadow maps*.

Selain itu *unity* juga dapat mengambil beberapa format design dari *3ds Maya* , *Max*, *Blender*, *Softimage*, *ZBrush*, *modo*, *Cheetah3D*, *Cinema 4D*, *Adobe Fireworks*, *Adobe Photoshop*, dan *Allegorithmic Substance*. Dan juga *Asset* tersebut nantinya dapat di tambahkan ke dalam *gameproject* lalu dilakukan pengaturan melalui *graphical user interface Unity*.

2. Scripting

Script dari *game engine* ini di buat dengan menggunakan *Mono v 2.6*, suatu implementasi *opensource* yangmana dari *NET Framework* dan *Programmers* dapat memakai *Unity-Script* (bahasa *tercustom* yang dalam bentuk *C#*, *JavaScript* , atau *Boo* (terinspirasi dari *syntax* bahasa pemrograman *python*) dan juga terinspirasi oleh *syntax ECMA Script*). Di mulai dengan di rilisnya v 3.0, *Unity* menyertakan versi dari *MonoDevelop* yang *tercustom* untuk *debug script*. Selain itu ada juga *Visual Studio* seperti pada gambar.



Gambar 2. 8 *Script Writer*

Sumber : <https://windowsreport.com/visual-studio-2019-version-16-2/>

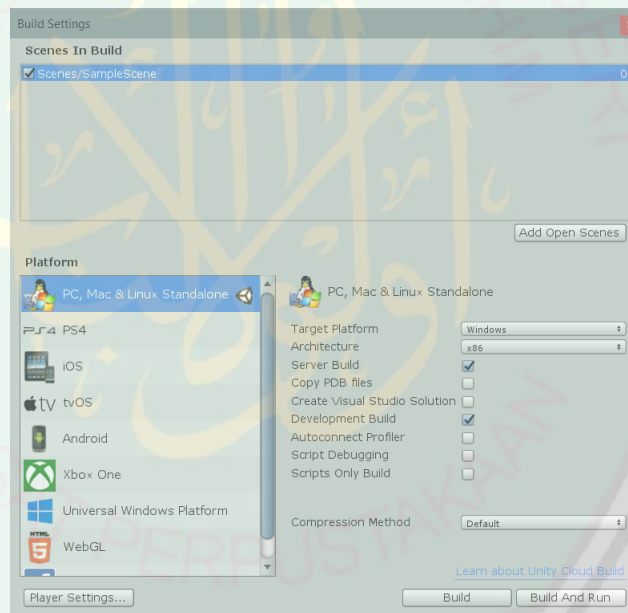
3. *Asset Tracking*

Server Unity Asset telah disediakan pada *unity*, yaitu sebuah solusi-solusi yang terkontroll untuk developers *script* dan *game asset*. *Server* yang digunakan adalah *PostgreSQL* yaitu sebagai *backend*. Lalu juga *audio system* yang di buat

memakai *FMOD library*, lalu juga *video playback* menggunakan *built-in light mapping*, *Theora codec*, dan juga *globals illuminations with Beast*. *Multiplayer-networking* dengan menggunakan *Rak Net*, dan juga *pathfinding built-in* atau pencari jalur (*navigation mesh*).

4. Platforms

Beberapa *platform* yangmana didukung oleh *Unity* adalah *Android*, *Mac*, *Windows*, *iOS*, *Linux*, *Adobe Flash*, *UnityWebPlayer*, *PlayStation3*, *Blackberry 10*, *Xbox360*, *Wii*, dan *Wii U*. Walaupun bukan semuanya yang ter-konfirmasi dengan resmi, Walau begitu platform yang didukung oleh *Unity* akan terus berkembang dan menjadi lebih *compatible* di berbagai *platform*.



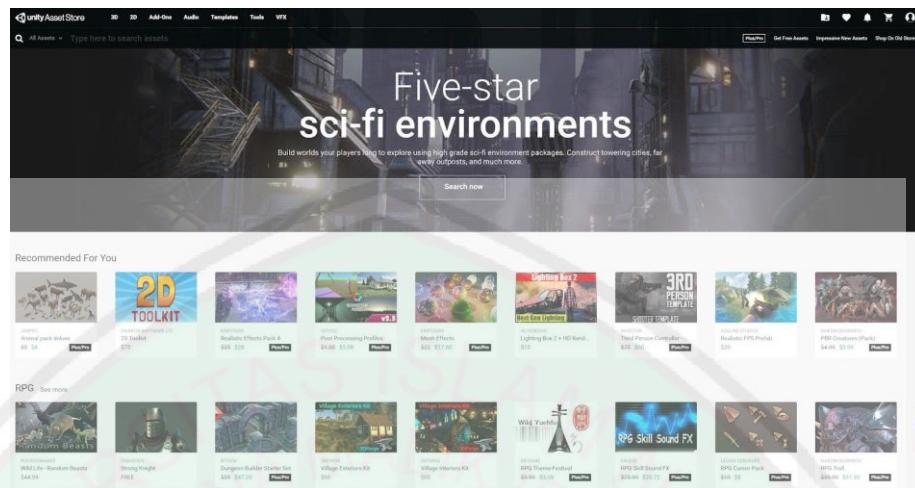
Gambar 2. 9 Platform Unity

Sumber : <http://docs.unity3d.com/2018.3/Documentation/Manual/BuildSettings.html>

5. Asset Store

Diluncurkan pada bulan November 2010, *Unity editor* telah dihadiri oleh sebuah *resource* dari *Unity asset store*. *Asset store* juga terdiri dari banyak koleksi-koleksi yaitu *3D models*, *asset packages*, *materials*, dan *textures*,

system particles, musik dan sound effect, tutorial dan projek, scripting package, online service dan editor extension.



Gambar 2. 10 Asset Store

Sumber : <http://unity3d.com/quick-guide-to-unity-asset-store>

6. *Physics*

Untuk PhysX physics engine, support built in juga telah dimiliki oleh Unity sejak Unity v 3.0 oleh Nvidia yang sebelumnya disebut ageia, dengan menambahkan kemampuan-kemampuan dalam simulasi *realtime cloth* pada *skinnedmeshes*, dan *arbitrary, collisionlayers, dan thick ray cast*.

7. *UI (User Interface)*

Merupakan fitur baru yang terdapat pada unity versi 4.6. Pada fitur UI ini sangat meminimalkan sebuah penggunaan *code*. Sehingga apa yang dibuat dan dilihat pada *canvas unity* adalah hasil yang akan didapatkan ketika saat *game* *dibuild*.

Di *menu game object* kita dapat membuat *object ui*, lalu klik atau pilih pada UI.

8. *Navigation and Pathfinding*

Merupakan fitur yang digunakan untuk membuat objek mengikuti jalur yang telah dilalui oleh objek lain dan pencarian jalur terdekat yang dapat ditempuh suatu objek.

2.5 Algoritma *Floyd-Warshall*

Algoritma *Floyd-Warshall*, adalah algoritma untuk mencari jalur-jalur yang paling pendek pada teori graff, hal tersebut berkaitan dengan pengoptimuman suatu masalah sehingga dapat meminimumkan biaya serta meningkatkan efisiensi waktu. Algoritma *Floyd-warshall* merupakan salahsatu varian dari program dinamik yang dapat mencari semua lintasan terpendek masing-masing antara tiap kemungkinan pasang tempat yang berbeda (*All pairs Shortest Path Problems*) dan sangat efektif digunakan dalam menangani masalah lintasan optimum (Suyitno, 2016).

Algoritma ini juga memiliki lebih dari satu kemungkinan eksekusi. Tiap eksekusi hanya memiliki satu solusi atau jalan untuk diproses, jika tidak ada solusi lagi maka algoritma akan dianggap selesai. Dalam membuat keputusan atau memproses perhitungan akan selalu berkaitan dengan solusi sebelumnya.

Selain algoritma *Floyd-warshall* ada juga Algoritma *Dijkstra* untuk memecahkan masalah pencarian jarak terdekat. Algoritma *Dijkstra* hanya menghitung dan menentukan rute terpendek dari titik asal dengan titik lainnya dalam graf berarah yang berbobot. Berbeda dengan Algoritma *Floyd-warshall* yang dapat menentukan masing-masing rute yang terpendek diantara dua titik kedalam graff berarah berbobot. Selain itu dalam hal *recovery time*, algoritma *floyd-warshall* terbukti lebih unggul dibanding algoritma *dijkstra* karena lebih cepat *recovery time*-nya.

Algoritma untuk mencari rute terpendek dengan algoritma *floyd-warshall* adalah dengan proses, dimisalkan W_0 adalah matriks ketetangaan awal graf berarah berbobot. W^* adalah matriks ketetangaan berbobot terpendek dengan W_{ij} sama dengan path terpendek dari titik V_i ke V_j (Sani, Tastrawati, & Dwipayana, 2013).

- 1) $W = W_0$
- 2) Untuk $k = 1$ hingga n , lakukan:
 Untuk $i = 1$ hingga n , lakukan:
 Untuk $j = 1$ hingga n , lakukan:
- 3) Jika $W[i,j] > W[i,k] + W[k,j]$ maka tukar $W[i,j]$ dengan $W[i,k] + W[k,j]$
- 4) $W^* = W$

Keterangan :

W : Matriks

W_0 : Matriks awal

W^* : Matriks akhir

i : Titik awal dari V_i

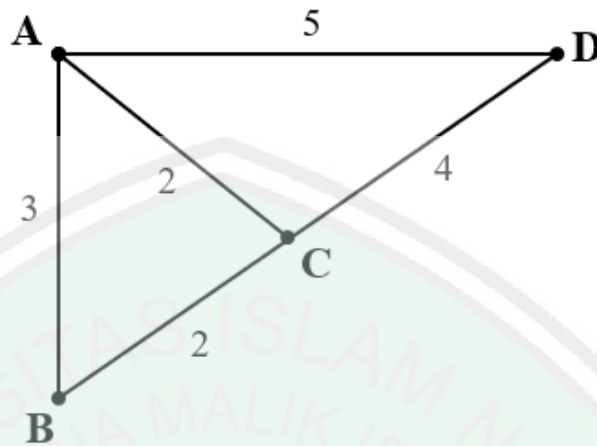
j : Titik awal dari V_j

k : Iterasi dari 1 sampai dengan n

Input : Matriks hubung graf berarah berlabel.

Output : Jalur terpendek dari semua pasang titik.

Salah satu contoh perhitungan manual Algoritma *Floyd-warshall* adalah :



Gambar 2. 11 Contoh Gambar Graf Berbobot

Dari gambar di atas didapat:

Matriks W^0 dan Matriks Hubung Graf $K=0$

$k = 0,1,2,3,4$

$i = A,B,C,D$

$j = A,B,C,D$

Lalu bentuk matriks nya sebagai berikut.

Tabel 2. 2 Matriks W_0

	A	B	C	D
A	0	3	2	5
B	3	0	2	∞
C	2	2	0	4
D	5	∞	4	0

Pada tabel 2.2 matriks diisi sesuai dengan jarak antar titik, untuk jarak antar titik yang sama diisi dengan 0. Jika jarak antar titik dengan titik lain melewati lebih dari satu jalur, maka diisi dengan ∞ . Selanjutnya membuat matriks W_1 .

Rumus : $W[i,j] > W[i,k] + W[k,j]$

Baris 1 dan Kolom 1 = Tetap

Tabel 2. 3 Matriks W_1

	A	B	C	D
A	0	3	2	5
B	3	0	2	8
C	2	2	0	4
D	5	8	4	0

$$W^0[B,C] (\dots) W^0[B,A] + W^0[A,C] = 2 > 3 + 2$$

$$W^0[B,D] (\dots) W^0[B,A] + W^0[A,D] = \infty > 3 + 5$$

$$W^0[C,B] (\dots) W^0[C,A] + W^0[A,B] = 2 < 2 + 3$$

$$W^0[C,D] (\dots) W^0[C,A] + W^0[A,D] = 4 < 2 + 5$$

$$W^0[D,B] (\dots) W^0[D,A] + W^0[A,B] = \infty > 5 + 3$$

$$W^0[D,C] (\dots) W^0[D,A] + W^0[A,C] = 4 < 5 + 2$$

Pada tabel 2.3 pada matriks W_1 , kolom pertama baris pertama diisi sama dengan kolom pertama baris pertama pada matriks W_0 . Setelah itu masukkan rumus $W[i,j] > W[i,k] + W[k,j]$ untuk menghitung matrik yang belum terisi.

Rumus : $W[i,j] > W[i,k] + W[k,j]$

Baris 2 dan Kolom 2 = Tetap

Tabel 2. 4 Matriks W_2

	A	B	C	D
A	0	3	2	5
B	3	0	2	8
C	2	2	0	4
D	5	8	4	0

$$W^1 [A,C] (\dots) W^1 [A,B] + W^1 [B,C] = 2 < 3 + 2$$

$$W^1 [A,D] (\dots) W^1 [A,B] + W^1 [B,D] = 5 < 3 + 8$$

$$W^1 [C,A] (\dots) W^1 [C,B] + W^1 [B,A] = 2 < 2 + 3$$

$$W^1 [C,D] (\dots) W^1 [C,B] + W^1 [B,D] = 4 < 2 + 8$$

$$W^1 [D,A] (\dots) W^1 [D,B] + W^1 [B,A] = 5 < 8 + 3$$

$$W^1 [D,C] (\dots) W^1 [D,B] + W^1 [B,C] = 4 < 8 + 2$$

Pada tabel 2.4 baris kedua kolom kedua pada matriks W_2 diisi sama dengan baris kedua kolom kedua pada matriks W_1 . Setelah itu masukkan rumus $W[i,j] > W[i,k] + W[k,j]$ untuk menghitung matrik yang belum terisi.

Rumus : $W[i,j] > W[i,k] + W[k,j]$

Baris 3 dan Kolom 3 = Tetap

Tabel 2. 5 Matriks W_3

	A	B	C	D
A	0	3	2	5
B	3	0	2	6
C	2	2	0	4
D	5	6	4	0

$$W^2[A,B] (\dots) W^2[A,C] + W^2[C,B] = 3 < 2 + 2$$

$$W^2[A,D] (\dots) W^2[A,C] + W^2[C,D] = 5 < 2 + 4$$

$$W^2[B,A] (\dots) W^2[B,C] + W^2[C,A] = 3 < 2 + 2$$

$$W^2[B,D] (\dots) W^2[B,C] + W^2[C,D] = 8 > 2 + 4$$

$$W^2[D,A] (\dots) W^2[D,C] + W^2[C,A] = 5 < 4 + 2$$

$$W^2[D,B] (\dots) W^2[D,C] + W^2[C,B] = 8 > 4 + 2$$

Pada tabel 2.5 baris ketiga kolom ketiga pada matriks W_3 diisi sama dengan baris ketiga kolom ketiga pada matriks W_2 . Setelah itu masukkan rumus $W[i,j] > W[i,k] + W[k,j]$ untuk menghitung matrik yang belum terisi.

Rumus : $W[i,j] > W[i,k] + W[k,j]$

Baris 4 dan Kolom 4 = Tetap

Tabel 2. 6 Matriks W_4

	A	B	C	D
A	0	3	2	5
B	3	0	2	6
C	2	2	0	4
D	5	6	4	0

$$W^3[A,B] (\dots) W^3[A,D] + W^3[D,B] = 3 < 5 + 6$$

$$W^3[A,C] (\dots) W^3[A,D] + W^3[D,C] = 2 < 5 + 4$$

$$W^3[B,A] (\dots) W^3[B,D] + W^3[D,A] = 3 < 6 + 5$$

$$W^3[B,C] (\dots) W^3[B,D] + W^3[D,C] = 2 < 6 + 4$$

$$W^3[C,A] (\dots) W^3[C,D] + W^3[D,A] = 2 < 4 + 5$$

$$W^3[C,B] (\dots) W^3[C,D] + W^3[D,B] = 2 < 4 + 6$$

Pada tabel 2.6 baris keempat kolom keempat pada matriks W_4 diisi sama dengan baris keempat kolom keempat pada matriks W_3 . Setelah itu masukkan rumus $W[i,j] > W[i,k] + W[k,j]$ untuk menghitung matrik yang belum terisi. Ini merupakan langkah atau iterasi terakhir sesuai dengan banyak node atau titik yang telah ditentukan pada awal perhitungan.

Dari tabel dapat diketahui jalur terpendek pada setiap titik dalam graf berbobot. Sehingga dapat disimpulkan dengan melihat baris dan kolom pada hasil matriks. Hasil jarak dari node atau titik A ke B adalah 3, jarak dari node atau titik A ke C adalah 2, jarak dari node atau titik A ke D adalah 5 dan seterusnya.



BAB 3

METODOLOGI PENELITIAN

3.1 Deskripsi *Game*

Game yang dibuat pada penelitian ini, merupakan *game* yang ber *genre* strategi dan edukasi. *Game* ini dibuat dalam *platform desktop*. *Game* ini bercerita tentang seorang pemuda desa yang terjebak dalam situasi bencana gunung meletus. Pemuda ini memiliki misi untuk menyelamatkan atau mengevakuasi masyarakat di desanya.

Game dengan kisah seorang pria atau pemuda (*main character*) yang berusaha menyelesaikan misi evakuasi bencana letusan gunung berapi. Konsep permainannya adalah tentang apa yang harus dilakukan oleh (*main character*) saat akan terjadi letusan gunung berapi. Pada bagian misi atau *quest* nya adalah tentang bagaimana cara melaporkan bahwa telah terjadi bencana, apa saja barang yang dibutuhkan saat terjadi bencana, dan kemana kita harus mengevakuasi warga di desa tersebut. Pada *game* ini pemain (*main character* harus mengikuti instruksi untuk menyelesaikan misi).

3.2 *Storyline*

Game ini menceritakan tentang seorang pemuda yang tengah berada pada sebuah desa. Pemuda ini tinggal di sebuah desa yang dekat dengan kaki gunung merapi. Lalu kejadian tanda-tanda gunung meletus muncul. Seorang Ayah memberitahukan bahwa asap dari gunung, suhu sekitar kaki gunung yang meningkat, hewan yang berlarian menjadi tanda gunung akan meletus

Pemain diminta untuk menyelesaikan misi yang diberikan. Jika misi pertama telah terpenuhi, maka misi selanjutnya akan muncul. Setiap misi yang diberikan memberi sebuah pelajaran dan langkah langkah tentang apa apa saja yang harus

dilakukan saat terjadi bencana alam. Hadiah yang diberikan setelah menyelesaikan misi adalah diizinkan untuk meneruskan ke misi setelahnya.

Bagian pertama adalah *introduction* atau pengenalan. *Player* sebagai pemuda desa berada pada suatu desa lalu melakukan serangkaian misi yang mana misi tersebut memperkenalkan tata cara bermain di *game* tersebut. Tata cara seputar tombol berjalan, berlari, mengambil *item*, berdialog, dan sebagainya. Pada level pertama *player* harus menyelesaikan misi *introduction* untuk lanjut ke level berikutnya. Di level ini tidak diberikan sebuah tantangan agar *player* dapat memahami tata cara bermain di *game* ini tanpa mengkhawatirkan kekalahan.

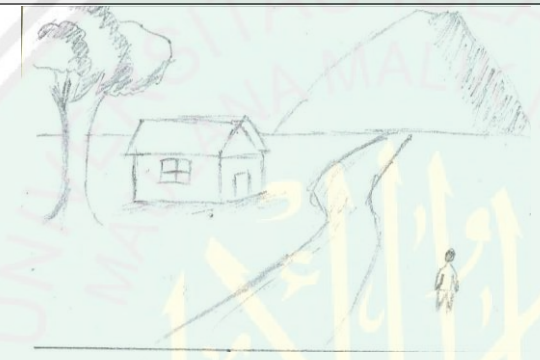

Bagian kedua adalah mengajak warga. Pada level ini *player* di beri sebuah misi yang mana misi tersebut adalah untuk mencari beberapa *item* yang penting. Selain itu *player* juga diberi misi untuk mencari serta mengajak warga terdekat untuk menjauh dari area kaki gunung merapi. Di level ini *player* harus bermain cepat dalam mencari dan mengajak warga karena terdapat tantangan di dalam level ini. Waktu akan dihitung mundur sampai akhirnya gunung merapi meletus sehingga *player* harus bergegas dalam mengajak warga menjauh dari area kaki gunung.

Bagian ketiga adalah selamatkan warga. Pada level ini *player* beralih menjadi tim SAR. Di level ini tim SAR memiliki misi untuk menemukan warga yang diperkirakan masih tertinggal di desa dekat area kaki gunung merapi. Akan ada beberapa *NPC* yang akan di letakkan di desa untuk di tolong. Pada misi terakhir ini akan diberikan tantangan health point yang terus menurun dikarenakan asap atau abu vulkanik dari gunung merapi. Level ketiga akan selesai jika *player* dapat membawa *NPC* warga keluar dari desa.

3.3 Desain Storyboard

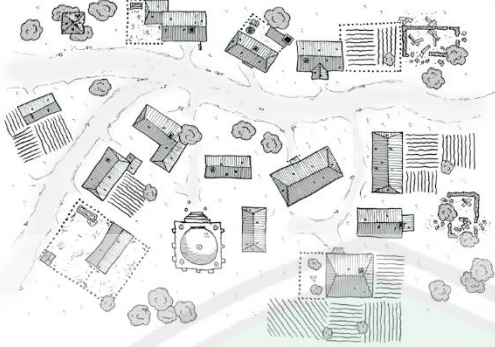
Pembuatan *storyboard* diperlukan sebagai rancangan dan juga acuan cerita dari *game* yang akan dibuat. *Storyboard* dapat membuat pembaca lebih paham akan gambaran dari cerita yang dibuat. Pada *storyboard* ini terdapat tiga level permainan. Diantaranya adalah seperti yang ada pada tabel berikut ini.

Tabel 3. 1 Desain *Storyboard*

No	Gambar	Detail
1		Pada level satu, yaitu <i>introduction</i> , <i>player</i> berada pada suatu desa dimana desa tersebut bertempat di dekat kaki gunung berapi.
2		Sesuai dengan tema level satu yaitu <i>introduction</i> , misi yang dilakukan oleh <i>player</i> adalah pengenalan seputar tata cara memainkan <i>game</i> ini, salah satunya adalah pengenalan fungsi keyboard.

No	Gambar	Detail
3		<p>Misi selanjutnya adalah dengan mencari item tas lalu mencoba untuk mengambilnya. Misi ini akan menjadi misi <i>introduction</i> terakhir pada level satu. Akan ada panduan dalam bentuk teks untuk mengambil sebuah item</p>
4		<p>Pada level dua, yaitu mengajak warga. Tanda tanda gunung akan meletus telah muncul. Misi pertama adalah masuk ke rumah untuk mencari beberapa item penting untuk dibawa. Yang membedakan di level dua ini adalah ditambahkan rintangan time untuk membatasi waktu permainan</p>
5		<p>Setelah berhasil masuk ke rumah, misi berikutnya adalah mencari beberapa item seperti pada gambar. Barang tersebut sangat penting untuk dibawa ketika bencana.</p>

No	Gambar	Detail
6		<p>Misi selanjutnya adalah mengajak warga terdekat untuk keluar dari daerah rawan bencana. Setelah misi ini selesai akan masuk pada level berikutnya yaitu level tiga.</p>
7		<p>Pada level tiga, yaitu mengevakuasi warga. Player akan berganti karakter menjadi tim SAR. Lalu mengetahui bahwa masih ada beberapa warga yang tertinggal maka tim SAR atau player diberi misi untuk kembali ke desa dan mencari warga yang tertinggal. Pada level ini terdapat abu vulkanik yang menutup jarak pandang player.</p>

No	Gambar	Detail
8		<p>Beberapa NPC warga yang tertinggal akan diletakkan di beberapa titik di sekitar desa. Setelah penyelamatan berhasil maka misi akan sukses.</p>

3.4 User Interface

Pengertian dari *User Interface* adalah serangkaian tampilan-tampilan grafis yang nantinya bisa mudah di mengerti oleh user ataupun pemakai *computer* & di program dengan berbagai macam bentuk sehingga nantinya bisa dibaca oleh OS *computer* dan ber operasi dengan sesuai, dengan kata lain UI ini fungsinya adalah agar sistem dan *user* dapat terhubung. Desain *User Interface* pada *game* ini diantaranya yaitu:

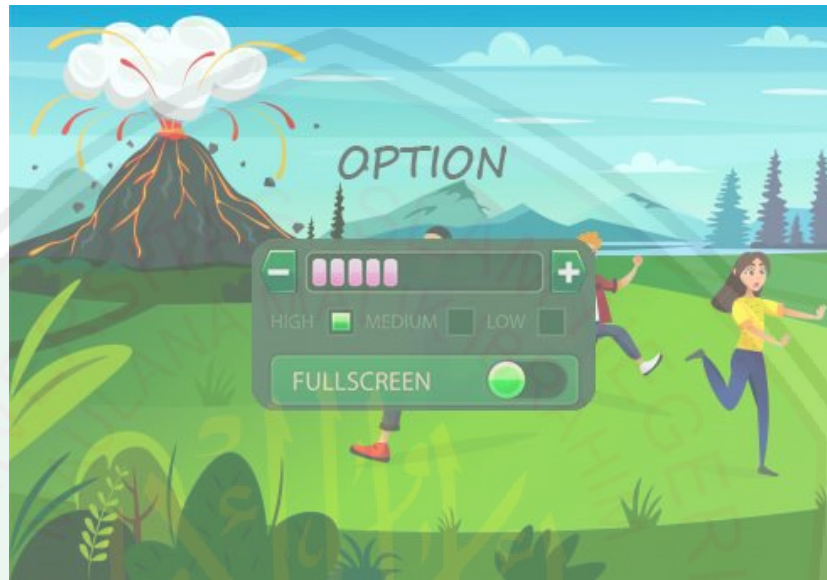
1. Main Menu



Gambar 3. 1 Desain *Main Menu*

Merupakan layar utama atau menu utama dari *game* yang dibuat. Pada main menu terdapat beberapa tombol diantaranya tombol *play*, *how to play*, *option*, dan *exit*. Tiap-tiap tombol atau *button* memiliki fungsi-fungsi yang berbeda.

2. Menu *Option*



Gambar 3. 2 Desain Menu Option

Merupakan fungsi dari tombol *option*. Terdapat pengaturan suara pada panel option. Dengan pengaturan suara tersebut pemain dapat menentukan besar kecilnya suara yang diinginkan.

3. Menu *How To Play*



Gambar 3. 3 Desain Menu How to Play

Merupakan fungsi dari tombol *how to play*. Terdapat *tutorial* atau tata cara seputar *game* ini. Diantaranya fungsi *keyboard*, dan fungsi *mouse*. Hal ini dibuat agar pemain dapat mengerti fungsi dari setiap tombol *keyboard* maupun *mouse*. Pada layar ini *tutorial* atau tata cara akan dijelaskan dalam bentuk teks sehingga untuk mengetahui fungsi tombol *keyboard* maupun *mouse* dapat diketahui hanya dengan membaca teks tersebut.

4. *Gameplay*



Gambar 3. 4 Tampilan *Gameplay*

Gambaran dari *game* yang nanti akan di buat adalah Seperti yang ada pada gambar di atas. Tampilan awal bermain *game* akan berada pada sebuah desa dekat kaki gunung merapi. Lalu awal permainan atau *gameplay* akan diberikan misi seputar pengenalan tombol tombol keyboard agar *player* atau pemain dapat mengetahui fungsi keyboard dalam *game* tanpa mengkhawatirkan kekalahan dalam memainkan *game* ini.

3.5 Deskripsi Karakter dan Objek

Beberapa macam karakter serta objek pada *game* ini adalah sebagai berikut.

1. Pemuda

Merupakan karakter utama pada *game* ini. Karakter pemuda ini akan dimainkan langsung oleh *player* atau user. Karakter utama ini akan digunakan pada bagian pertama dan kedua untuk menjalankan serangkaian misi sehingga dapat menuju ke level berikutnya.

2. Warga Desa

Merupakan *NPC* (*non player character*) yang akan dievakuasi oleh *player* sehingga *NPC* dapat terselamatkan dari bencana. *NPC* inilah yang nantinya akan diberikan sebuah algoritma *floyd-warshall* untuk mencari jalur terpendek ke titik tertentu.

3. Tim SAR

Merupakan karakter yang akan digunakan pada bagian ketiga. Misinya adalah untuk mencari *NPC* warga desa yang tertinggal. Rintangan abu vulkanik akan diberikan pada karakter ini.

4. Barang penting

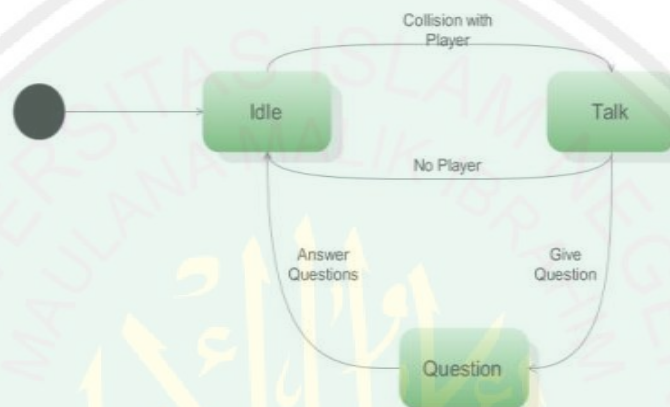
Merupakan barang atau *item* yang digunakan untuk menyelesaikan misi dikarenakan barang tersebut penting untuk dibawa saat keadaan pra bencana. Barang-barang tersebut seperti tas, pakaian, makanan kering, obat-obatan, masker, dan sebagainya. Barang ini akan disebar di area rumah untuk selanjutnya dicari oleh *player*.

3.6 Finite State Machine (FSM)

Beberapa FSM pada game ini diantaranya adalah sebagai berikut.

1. Kondisi Berbicara

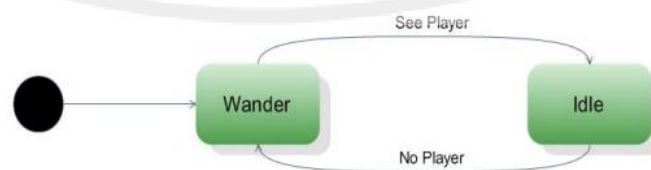
Disini saat NPC bertemu dengan player akan berbincang bincang dengan player, jika perbincangan selesai maka NPC akan berjalan seperti biasa kembali. Saat bertemu NPC lain, NPC akan tetap berjalan seperti biasa



Gambar 3. 5 Kondisi Berbicara

2. Kondisi Waypoint

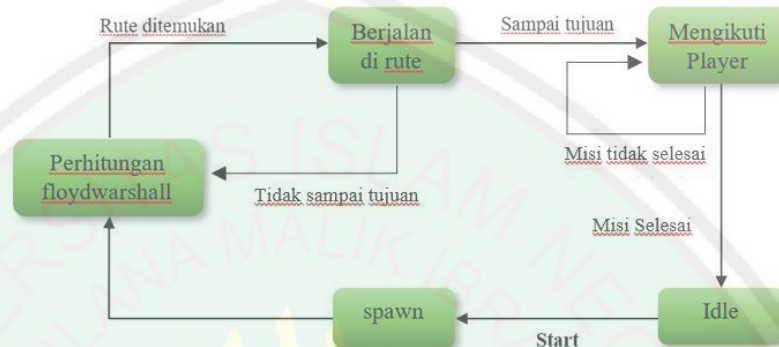
Disini saat NPC bertemu dengan player dalam jarak beberapa meter, maka NPC akan berhenti. Berbeda dengan saat bertemu NPC lain, maka NPC akan tetap berjalan seperti biasa.



Gambar 3. 6 Kondisi Waypoint

3. Kondisi *Pathfinding*

Disini NPC ditugaskan untuk mencari jalur terpendek yang telah ditentukan. Tujuannya adalah untuk menuju ke titik kumpul dan selanjutnya mengikuti player untuk keluar menuju titik evakuasi.



Gambar 3. 7 Kondisi *Pathfinding*

3.7 Perhitungan *Floyd-Warshall*

Pada *game* ini, algoritma *floyd-warshall* akan diterapkan pada *NPC* yang mengikuti *player*. Sehingga *NPC* dapat mengetahui jarak tercepat untuk menuju ke arah *player*. Algoritma *floyd-warshall* ini akan diterapkan dalam bahasa pemrograman *C#* (*C-Sharp*) yang ada pada *Unity*.

Algoritma penemuan dari warshall ini merupakan algoritma yang optimal dan sederhana untuk digunakan sebagai pencarian jalur terpendek. Algoritma ini memiliki masukan dalam bentuk matriks graf berarah dan memiliki keluaran jalur terpendek yang dari semua node atau titik ke semua node atau titik.

Proses perhitungan algoritma *floyd-warshall* sehingga mendapatkan keluaran jalur terpendek adalah sebagai berikut.

1. Pada iterasi pertama, dilakukan pengecekan pada setiap sel matriks. apakah jarak dari dua titik mula-mula lebih besar dari penjumlahan jarak titik awal ke

titik tujuan (titik tujuan=Iterasi pertama) dengan jarak titik awal (titik awal=Iterasi pertama) ke titik tujuan. Singkatnya mengecek apakah $W[i,j] > W[i,k] + W[k,j]$.

2. Dari hasil proses pertama, jika jawaban “iya” maka jarak dari dua titik mula-mula diganti dengan hasil penjumlahan jarak titik awal ke titik tujuan (titik tujuan=Iterasi pertama) dengan jarak titik awal (titik awal=Iterasi pertama) ke titik tujuan. $W[i,j]$ diganti dengan $W[i,k] + W[k,j]$.
3. Dari hasil proses pertama, jika jawaban “tidak” maka selanjutnya jarak dari kedua titik awal atau mula-mula akan digunakan pada iterasi berikutnya (tidak diganti).
4. Proses iterasi akan terus dilakukan sampai dengan iterasi yang terakhir. Banyak iterasi adalah banyak titik pada graf. Dapat di lihat pada gambar di bawah:



Gambar 3. 8 Titik Pada Denah Game

Rumusnya adalah sebagai berikut.

$$W = W_0$$

Untuk $k = 1$ hingga n , lakukan:

Untuk $i = 1$ hingga n , lakukan:

Untuk $j = 1$ hingga n , lakukan:

Jika $W[i,j] > W[i,k] + W[k,j]$ maka tukar $W[i,j]$ dengan $W[i,k] + W[k,j]$

$$W^* = W$$

Dalam proses iterasinya, untuk mencari jalur yang terdekat, sesuai dengan *iterasi-k* maka *floyd-warshall* akan membentuk n matriks. Hal ini yang menyebabkan proses dari algoritma ini dapat lebih lambat dibanding algoritma yang lainnya. Terutama untuk n yang besar.

Di sisi lain kelebihanannya adalah kesederhanaan algoritmanya, serta hasil yang optimal. Dalam setiap langkahnya, melalui v_i dan v_j nantinya algoritma *floyd-warshall* akan memeriksa jalur, apakah dengan melalui jalur v_i-v_k & v_k-v_j akan lebih pendek. Berikut pseudocode algoritma *floyd-warshall*.

```
function fw(int[1..n,1..n] graph){
  // Inisialisasi
  var int[1..n,1..n] jarak := graph
  var int[1..n,1..n] sebelum
  for i from 1 to n
    for j from 1 to n
      if jarak[i,j] < Tak-hingga
        sebelum[i,j] := i
  // Perulangan utama pada algoritma
  for k from 1 to n
    for i from 1 to n
      for j from 1 to n
        if jarak[i,j] > jarak[i,k] + jarak[k,j]
          jarak[i,j] = jarak[i,k] + jarak[k,j]
          sebelum[i,j] = sebelum[k,j]
  return jarak
}
```

Gambar 3. 9 Pseudocode algoritma floyd-warshall

BAB 4

HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada bab ini membahas tentang hasil seputar implementasi algoritma yang digunakan dan juga hasil dari perancangan desain *game*, apakah perancangan telah sesuai dengan hasil yang diharapkan. Tujuan dari penelitian ini adalah untuk mengetahui bagaimana cara implementasi dari algoritma *floyd-warshall* terhadap suatu *NPC*.

Sebelum menjelaskan hasil implementasi, terlebih dahulu dipaparkan tentang spesifikasi sistem perangkat keras (*hardware*) dan juga perangkat lunak (*software*) yang digunakan dalam pembuatan dan pengujian pada penelitian ini.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan pada pembuatan dan pengujian *game* adalah sebagai berikut.

Tabel 4. 1 Tabel Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Intel(R) Core(TM) i5-6200 2.30 GHz
2	RAM	12 GB
3	VGA	NVIDIA GEFORCE 930 MW
4	HDD	1 TB
5	Monitor	14'

No	Perangkat Keras	Spesifikasi
6	<i>Speaker</i>	<i>On</i>
7	<i>Mouse & Keyboard</i>	<i>On</i>

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan pada pembuatan dan pengujian *game* adalah sebagai berikut.

Tabel 4. 2 Tabel Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	<i>Windows 10 64 Bit</i>
2	<i>Game Engine</i>	<i>Unity 3D</i>
3	Desain 2D	<i>Adobe Photoshop CC</i>
4	<i>Script Writer</i>	<i>Monodevelop</i>

4.2 Implementasi *Game*

Implementasi *Game* merupakan pengaplikasian tampilan dari *game* yang telah dirancang dan dibuat. Dalam implementasinya, *game* ini terdapat beberapa tampilan fungsi yang berkaitan antara satu dengan yang lainnya. Tampilan fungsi dari *game* yang dirancang antara lain adalah:

4.2.1 Antarmuka *Main Menu*

Merupakan Menu utama yang muncul setelah permainan dijalankan. Akan terdapat 4 tombol utama pada *scene* ini, yaitu tombol “*PLAY*”, “*HOW TO PLAY*”, “*OPTION*”, dan “*EWIT*”. Pada tombol “*PLAY*” akan langsung menjalankan *Gameplay* nya.



Gambar 4. 1 MainMenu

4.2.2 Antarmuka *How to play*

Merupakan tampilan yang akan muncul apabila tombol “*How to play*” ditekan. Pada tampilan ini berisi tentang tata cara atau *hint* untuk menyelesaikan game ini. Hint dijelaskan secara singkat pada panel ini.



Gambar 4. 2 Panel *How To Play*

4.2.3 Antarmuka *Option*

Merupakan tampilan yang berisi *slider* pengaturan *background*. Seret ke kiri untuk *volume down*, dan seret ke kanan untuk *volume up*. Sedangkan tombol *back* untuk kembali ke *MainMenu*.



Gambar 4. 3 Panel *Option*

4.2.4 Konten *Game*



Gambar 4. 4 Tampilan *gameplay*

Bagian pertama adalah misi – misi yang mudah. Misi seputar pengenalan fungsi *keyboard* yang dapat digunakan dalam *game* ini. Lalu bagian kedua adalah misi – misi yang lumayan sulit dikarenakan *player* akan diminta melakukan serangkaian misi seputar mencari barang atau kabur ke luar desa dengan *countdown* di dalam *game* nya. Bagian ini ditandai dengan panel “*Introduction*” dan “Pra Bencana” pada bagian tengah atas.

4.2.5 Misi Game



Gambar 4. 5 Tampilan Misi

Misi pada game ini dibuat berurutan sehingga *player* tidak akan bisa mengambil misi jika tidak sesuai dengan urutan misi. Misi pada game ini dapat dilihat pada tampilan *how to play* atau mengikuti petunjuk pada *gameplay* nya. Serangkaian misi ini akan diberi waktu yang mana jika waktu habis akan *game over*.

4.2.6 Mission Complete

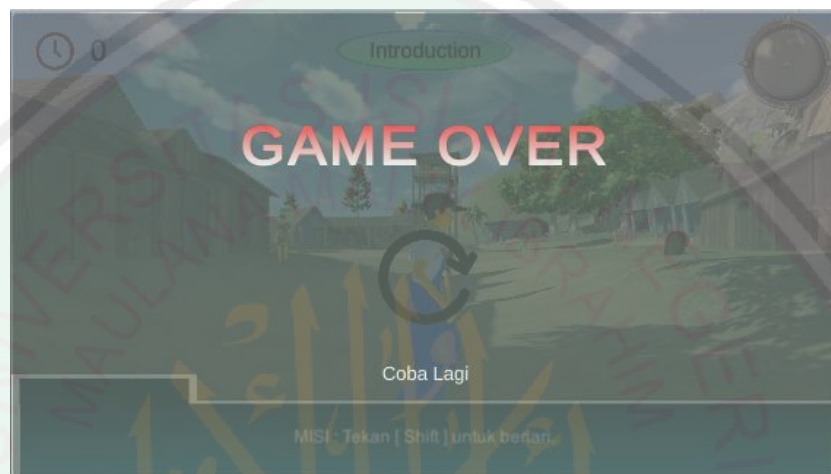
Panel *mission complete* merupakan panel yang dibuat agar saat semua misi *player* telah tercapai maka panel akan dimunculkan pada permainan. Pada panel ini terdapat *button* untuk Kembali ke *Menu* utama.



Gambar 4. 6 Mission Complete

4.2.7 Game Over

Jika pada saat menyelesaikan misi, *player* tidak mampu menyelesaikannya dalam waktu yang telah ditentukan atau waktu telah habis, maka otomatis muncul panel *GameOver* sehingga *player* wajib memulai lagi permainan tersebut dari pertama (awal). Tombol ini langsung loadscene ke *gameplay*, bukan ke *mainmenu*.



Gambar 4. 7 Game Over

4.2.8 Karakter dan Objek yang Digunakan

Tabel 4. 3 Karakter dan Objek

No	Karakter/Objek	Gambar	Keterangan
1	<i>Player</i>		Karakter <i>third person controller</i> yang dikontrol oleh pemain. Dibuat untuk menjalankan serangkaian misi yang telah disediakan.

No	Karakter/Objek	Gambar	Keterangan
2	NPC Ayah		<p>NPC yang bertugas untuk memberikan misi kepada <i>player</i> sehingga <i>player</i> dapat menyelesaikan misi tersebut. NPC ini juga menyampaikan hal penting terkait mitigasi bencana gunung meletus.</p>
3	NPC Warga		<p>NPC yang digunakan sebagai pelengkap permainan dengan fungsi patrol atau <i>wander</i>, dan juga penempatan algoritma <i>floyd-warshall</i>.</p>
4	Item Quest		<p>Terdapat berbagai macam <i>item quest</i> penting yang harus dikumpulkan sesuai dengan saat akan terjadinya bencana</p>

No	Karakter/Objek	Gambar	Keterangan
			gunung meletus. Diantaranya senter, handphone, makanan, minuman, dokumen penting, masker, dan obat-obatan.
5	Objek Panggil Warga		Objek yang digunakan untuk mentrigger algoritma <i>floyd-warshall</i> sehingga NPC warga dapat berkumpul pada objek yang telah ditentukan sebagai titik target

4.3 Implementasi Algoritma *Floyd-Warshall*



Gambar 4. 8 Pengujian Pertama

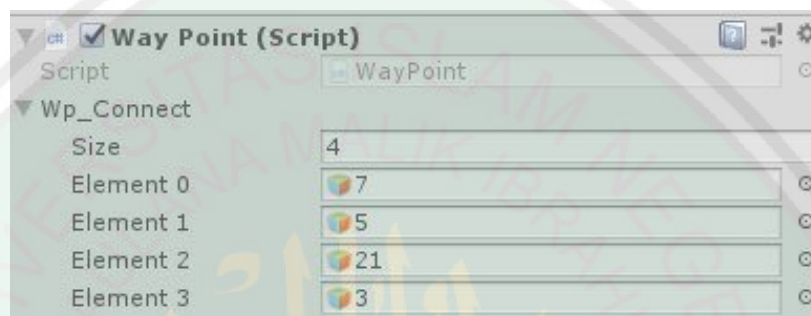
Pengujian pertama dimulai dengan cara menggunakan algoritma *floyd-warshall* pada lintasan (titik awal-titik akhir) yang pendek sehingga dapat dengan cepat diketahui apakah algoritma dapat berjalan atau tidak. Selain itu untuk melihat dan mengetahui proses dari pencarian rute terpendek maka dilakukan pengujian pada objek dari posisi awal menuju posisi atau titik target yang telah ditentukan. Setiap *node* memiliki jarak yang berbeda. *NPC* diberikan sebuah algoritma *floyd-warshall* untuk mencari jarak terpendek dari setiap *node*. Dengan dibuatnya algoritma *floyd-warshall* diharapkan *NPC* mencari jarak terpendek menuju ke titik yang telah ditentukan. Titik-titik yang telah dibuat diletakkan seperti pada gambar 4.8 dibawah ini.



Gambar 4. 9 Pemilihan Jalur

Uji coba dilakukan untuk mengetahui cara kerja dan proses *NPC* menuju ke titik yang telah ditentukan. Pengujian pertama dilakukan dengan menempatkan *NPC* pada titik nomor 6 dan titik target pada nomor 16 seperti pada gambar 4.9. Pada proses pertama ini yang menjelaskan bahwa algoritma *floyd-warshall* lebih optimal. Dikarenakan dalam pemilihan tiga jalur, algoritma ini menghitung jarak dari tiap titik dengan titik lainnya, dan bukan langsung memilih jarak terpendek

pada penentuan jalurnya. Sehingga pada proses pertama *NPC* memilih titik nomor 7 meskipun jarak ke titik 5 lebih pendek. Jarak dari titik 6 ke titik 7 adalah (37.98) dan jarak dari titik 6 ke titik 5 adalah (18.86), berbeda (19.12) pada kedua hasil jarak tersebut, namun algoritma ini memilih jarak terpanjang pada proses pertamanya. Percabangan jalur yang dapat dilewati dari titik nomor 6 dapat dilihat pada gambar di bawah ini.

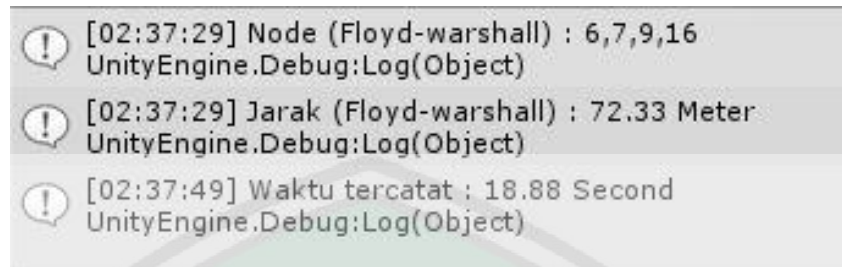


Gambar 4. 10 Hubungan tiap point

Penentuan jalur digunakan agar *NPC* tidak tertabrak dengan objek lain seperti rumah atau pohon, sehingga dapat terus berjalan pada jalur yang telah ditentukan. Pada proses selanjutnya pada titik nomor 7 hanya bisa memilih dua jalur, yaitu ke titik nomor 9 dan ke titik nomor 6. Pada proses kedua titik nomor 7 memilih titik nomor 9 dibanding kembali ke nomor 6 karena merupakan hasil dari perhitungan jarak terdekat menuju ke nomor 16. Begitu juga pada proses ketiga titik nomor 9 yang memilih ke titik nomor 16 dibanding kembali ke titik nomor 7.

Hasil dari pencarian jalur terpendek pada *NPC* ditampilkan dalam bentuk *debug.log* pada *tab console*. Terdapat 3 hasil yang ditampilkan yaitu, hasil perhitungan *floyd-warshall* dalam bentuk urutan *node* atau *waypoint* yang dilalui, lalu hasil perhitungan total jarak menggunakan fungsi *distance* dari titik awal ke titik akhir dalam satuan meter, dan yang terakhir adalah hasil perhitungan total waktu menggunakan fungsi *stopwatch* yang ditempuh dari titik awal ke titik akhir

dalam satuan *second*. Gambar 4.11 merupakan tampilan *tab console* hasil dari pengujian pertama.



```
[02:37:29] Node (Floyd-warshall) : 6,7,9,16
UnityEngine.Debug:Log(Object)
[02:37:29] Jarak (Floyd-warshall) : 72.33 Meter
UnityEngine.Debug:Log(Object)
[02:37:49] Waktu tercatat : 18.88 Second
UnityEngine.Debug:Log(Object)
```

Gambar 4. 11 Hasil Pengujian pertama

Sehingga pada pengujian pertama pencarian jarak terpendek diperoleh dari titik nomor 6 menuju ke titik nomor 16 adalah melalui *node* 6, 7, 9, 16. Lalu total jarak yang dari titik awal ke titik akhir adalah 72.33 meter. Dan total waktu perjalanan *NPC* dari titik awal ke titik akhir adalah 18.88 *second*.

Pengujian berikutnya adalah melanjutkan pengujian sebelumnya sampai dengan 30 kali pengujian. Pengujian ini dilakukan dengan cara menentukan titik awal dan titik akhir pada *inspector NPC*. Pada game ini dibuat *waypoint* atau *node* sebanyak 38 dalam bentuk *Game Object*. Untuk setiap *node* nya diberi hubungan sesuai dengan *node* yang ada di sebelahnya. Pada 30 pengujian ini *NPC* akan menemui percabangan dan akan menentukan jalur mana yang akan dilalui untuk menuju ke titik akhir atau titik yang telah ditentukan. Gambar 4.12 adalah gambaran denah jalur yang telah diberikan *waypoint*.



Gambar 4. 12 Waypoint pada jalur

Langkah berikutnya adalah mencatat *debug.log* pada tiap pengujian untuk mengetahui tingkat keberhasilan algoritma *floyd-warshall* pada *game* ini. Pada tabel 4.4 merupakan hasil dari perhitungan algoritma *floyd-warshall* beserta perhitungan total jarak dan total waktu dari titik awal ke titik akhir, selain itu dibuat juga kolom *error* untuk mengetahui apakah terjadi *error* saat *NPC* melakukan pencarian jalur.

Terdapat 30 pengujian data dari jarak tiap *node*. Prosesnya adalah dengan cara memasukan nilai titik awal dan nilai titik akhir. Setelah algoritma berjalan maka posisi *NPC* akan otomatis mengikuti posisi titik awal yang telah ditentukan pada *inspector*. Pengujian ini di lakukan guna mengecek jika algoritma *floyd-warshall* dapat berfungsi dengan baik dan benar. Hasil dari pengujian dicatat dalam bentuk tabel. Berikut adalah hasil dari pengujian algoritma *floyd-warshall*.

Tabel 4. 4 Pengujian dari titik awal ke titik akhir

No	Titik Awal	Titik Akhir	Waktu	Jarak	Jalur	Error
1	2	16	38.47 s	152.13 m	2>3>6>7>9>16	Tidak
2	36	16	51.4 s	198.16 m	36>37>38>22> 21>6>7>9>16	Tidak
3	33	16	50.67 s	201.39 m	33>31>32>1>8 >17>16	Tidak
4	29	17	41.50 s	166.38 m	29>28>27>26> 18>17	Tidak
5	28	17	24.5 s	97.82 m	28>27>26>18> 17	Tidak
6	38	17	39.57 s	156.81 m	38>22>21>6>7 >9>16>17	Tidak
7	34	18	40.69 s	161.02 m	34>32>1>8>17 >18	Tidak
8	31	18	39.18 s	152.72 m	31>32>1>8>17 >18	Tidak
9	24	18	26.89 s	105.22 m	24>23>1>8>17 >18	Tidak
10	25	19	20.89 s	72.69 m	25>26>18>19	Tidak
11	14	19	14.71 s	56.18 m	14>10>15>20> 19	Tidak
12	4	19	23.25 s	90.24 m	4>1>8>17>18> 19	Tidak
13	37	20	54.16 s	212.47 m	37>38>22>21> 6>7>9>16>13> 12>20	Tidak
14	23	20	24.61 s	97.13 m	23>1>8>17>18 >19>20	Tidak
15	30	20	41.58 s	165.24 m	30>24>23>1>8 >17>18>19>20	Tidak
16	38	12	43.07 s	170.31 m	38>22>21>6>7 >9>16>13>12	Tidak
17	35	12	54.18 s	214.61 m	35>34>32>1>8 >17>16>13>12	Tidak
18	31	12	44.07 s	175.44 m	31>32>1>8>17 >16>13>12	Tidak
19	33	13	53.45 s	211.69 m	33>31>32>1>8 >17>16>13	Tidak
20	30	13	39.44 s	156.42 m	30>24>23>1>8 >17>16>13	Tidak

No	Titik Awal	Titik Akhir	Waktu	Jarak	Jalur	Error
21	29	13	46.80 s	187.09 m	29>28>27>26>18>17>16>13	Tidak
22	28	11	26.00 s	104.25 m	28>27>26>18>19>11	Tidak
23	32	11	29.87 s	116.44 m	32>1>8>17>18>19>11	Tidak
24	6	11	29.10 s	112.33 m	6>7>9>16>17>18>19>11	Tidak
25	24	15	33.62 s	133.51 m	24>23>1>8>17>18>19>20>15	Tidak
26	27	15	23.70 s	92.55 m	27>26>18>19>20>15	Tidak
27	37	15	55.36 s	216.18 m	37>38>22>21>6>7>9>16>13>12>15	Tidak
28	4	10	33.41 s	129.89 m	4>1>8>17>18>19>20>15>10	Tidak
29	2	10	54.93 s	212.26 m	2>3>6>7>9>16>13>12>15>10	Tidak
30	23	10	33.32 s	127.41 m	23>1>8>17>18>19>20>15>10	Tidak

Pada Tabel 4.4 dapat di simpulkan jika tidak terjadi *error* dalam 30 kali pengujian dikarenakan karakter *NPC* dapat sampai ke titik tujuan. Dengan demikian implementasi algoritma *floyd-warshall* pada game mitigasi gunung meletus memiliki tingkat kesuksesan atau keberhasilan sebesar 100% dalam menentukan jalur atau rute terpendek.

4.4 Integrasi dengan Islam

Gunung merupakan ciptaan Allah yang mana telah menjadi sebuah bagian dari isi di muka bumi ini. Bagi orang beriman, beberapa fenomena gunung bukanlah hal baru. Allah telah banyak menytir dalam Al-Quran bahwa gunung tidaklah diam seperti yang manusia lihat, akan tetapi gunung-gunung bergerak sesuai dengan kehendak Allah.

وَتَرَى الْجِبَالَ تَحْسَبُهَا جَامِدَةً وَهِيَ تَمُرُّ مَرَّ السَّحَابِ صُنْعَ اللَّهِ الَّذِي أَتَقَنَ كُلَّ شَيْءٍ إِنَّهُ خَبِيرٌ بِمَا تَفْعَلُونَ

Artinya:

“Dan engkau akan melihat gunung-gunung, yang engkau kira tetap di tempatnya, padahal ia berjalan (seperti) awan berjalan. (Itulah) ciptaan Allah yang mencipta dengan sempurna segala sesuatu. Sungguh, Dia Mahateliti apa yang kamu kerjakan.” (Q.S An-Naml:88)

Secara teori, gerakan gunung-gunung disebabkan oleh gerakan kerak bumi yang mana kerak bumi ini seperti mengapung di atas lapisan magma yang lebih rapat. Sehingga fenomena gunung seperti bencana gunung meletus dapat terjadi karena pergerakan ataupun pergeseran lempeng bumi. Oleh karena itu, tugas manusia terhadap manusia yang lainnya tentu adalah saling tolong menolong. Terdapat banyak jenis dari perbuatan tolong menolong, salah satunya adalah memperingati bahaya, terutama bahaya seputar bencana alam.

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

Artinya:

“Dan tolong-menolonglah kamu dalam mengerjakan kebajikan dan takwa, dan jangan tolong-menolong dalam perbuatan dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksaan-Nya.” (Q.S Al-Maidah: 2).

Kebaikan (kebajikan) yang tertera di ayat di atas mencakup seluruh unsur agama Islam; prinsip-prinsip keimanan, penegakan syariat seperti mendirikan shalat, membayar zakat dan infak kepada orang yang membutuhkan dan amalan hati seperti bersabar dan menepati janji.

وَمَنْ أَحْيَاهَا فَكَأَنَّمَا أَحْيَا النَّاسَ جَمِيعًا ۗ وَلَقَدْ جَاءَتْهُمْ رُسُلُنَا بِالْبَيِّنَاتِ ثُمَّ إِنَّ كَثِيرًا مِّنْهُمْ بَعْدَ ذَلِكَ فِي الْأَرْضِ لَمُسْرِفُونَ

Artinya:

“Dan barangsiapa yang memelihara kehidupan seorang manusia, maka seolah-olah dia telah memelihara kehidupan manusia semuanya. Dan sesungguhnya telah datang kepada mereka rasul-rasul Kami dengan (membawa) keterangan-keterangan yang jelas, kemudian banyak diantara mereka sesudah itu sungguh-sungguh melampaui batas dalam berbuat kerusakan dimuka bumi.” (Q.S Al-Maidah: 32).

Tafsir Al-Mukhtashar / Markaz Tafsir Riyadh di bawah pengawasan Syaikh Dr. Shalih bin Abdullah bin Humaid (Imam Masjidil Haram) menjelaskan, bahwa kata وَمَنْ أَحْيَاهَا (Dan barangsiapa yang memelihara kehidupan seorang manusia) Yakni yang memaafkan orang yang berhak dibunuh. Mujahid berkata: memelihara kehidupannya adalah dengan menyelamatkan seseorang dari tenggelam, kebakaran, bangunan runtuh, atau mara bahaya.



BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan dari hasil implementasi serta pengujian yang telah dilakukan oleh peneliti bahwa dapat ditarik kesimpulan sebagai berikut.

Dalam penelitian yang telah dilakukan pengujian dari algoritma *floyd-warshall* yang diterapkan pada NPC sebanyak 30 kali pengujian dengan cara menentukan titik awal dan titik akhir pada *inspector* NPC. Titik atau node ini dibuat dalam bentuk *gameobject*. Dengan penentuan letak *waypoint* atau node ini dapat mencegah objek atau NPC bertabrakan dengan objek yang lain seperti bangunan rumah, pohon, maupun karakter NPC yang lain. Dari 30 kali pengujian didapatkan 30 data yang menunjukkan 100% sukses dari segi ketepatan dan kecepatan dalam menentukan jalur atau rute terpendek.

5.2 Saran

Penulis sadar, dalam pembuatan game ini masih banyak kekurangan yang nantinya perlu untuk dilakukan pengembangan, diantaranya :

1. Perbaiki asset yang lebih bagus baik dari asset 2D maupun 3D.
2. Penambahan jalur sehingga denah peta dapat lebih luas dikarenakan masih terdapat 38 waypoint dalam game ini.
3. Perbaiki seputar interface atau antarmuka sangat diperlukan agar tampilan dapat menjadi lebih menarik.
4. Game ini hanya dapat dioperasikan pada Desktop, alangkah baiknya jika nantinya dapat dikembangkan ke berbagai platform.

DAFTAR PUSTAKA

- Maarif, S., A. Kinseng, R., & Pramono, R. (2016). Dimensi Sosial Dalam Penanganan Bencana (Studi Kasus penanganan GempaBumi di Kecamatan Pangalengan, Kabupaten Bandung. *Jurnal Riset Kebencanaan Indonesia*, 2, 95–105.
- Marlina, L., & Suyitno, A. (2017). *Penerapan Algoritma Dijkstra Dan Floyd-Warshall Untuk Menentukan Rute Terpendek Tempat Wisata Di Batang*. 12.
- Muzdalifah, L., Oktafianto, K., & Mustika, E. D. (2018). *Model Jaringan Distribusi Beras Optimal Menggunakan Algoritma Floyd Warshall*. 11.
- Prasetya, E. (2017). Dari 17.504 Pulau di Indonesia, 16.056 telah diverifikasi PBB. Retrieved August 3, 2019, from Merdeka.com website: <https://www.merdeka.com/peristiwa/dari-17504-pulau-di-indonesia-16056-telah-diverifikasi-pbb.html>
- Pratama, A., Laila N, A., & Putra W, A. (2014). *Pemodelan Kawasan Rawan Bencana Erupsi Gunung Api Berbasis Data Penginderaan Jauh (Studi Kasus Di Gunung Api Merapi)*. 3, 7.
- Ramadiani, Bukhori, D., Azainil, & Dengen, N. (2018). Floyd-warshall algorithm to determine the shortest path based on android. *IOP Conference Series: Earth and Environmental Science*, 144, 012019. <https://doi.org/10.1088/1755-1315/144/1/012019>

- Sani, A. F., Tastrawati, N. K. T., & Dwipayana, I. M. E. (2013). Algoritma Floyd Warshall Untuk Menentukan Jalur Terpendek Evakuasi Tsunami Di Kelurahan Sanur. *E-Jurnal Matematika*, 2(1), 1. <https://doi.org/10.24843/MTK.2013.v02.i01.p020>
- Suyitno, A. (2016). *Simulasi Jaringan Jalan Di Kota Semarang Berbasis Algoritma Floyd-Warshall Untuk Menangani Masalah Lintasan Terpendek*. 8.
- Vebri. (2018). Waspada 68 Gunung Api Aktif di Indonesia, Pantau Status Gunung Berapi Melalui Aplikasi Magma. Retrieved August 3, 2019, from MSN website: <https://www.msn.com/id-id/berita/nasional/waspada-68-gunung-api-aktif-di-indonesia-pantau-status-gunung-berapi-melalui-aplikasi-magma/ar-BBRwrVc>
- Zakky. (2018). Letak Geografis Indonesia Beserta Dampak dan Pengaruhnya. Retrieved August 3, 2019, from ZonaReferensi.com website: <https://www.zonareferensi.com/letak-geografis-indonesia/>