

**PENGATURAN LEVEL KESULITAN DINAMIS MENGGUNAKAN METODE
SUPPORT VECTOR MACHINE (SVM) PADA GAME
EDUKASI BENCANA GUNUNG MELETUS**

SKRIPSI

**Oleh:
MUHAMMAD FAHRUL AKBAR
NIM. 16650071**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**PENGATURAN LEVEL KESULITAN DINAMIS MENGGUNAKAN
METODE *SUPPORT VECTOR MACHINE* (SVM) PADA *GAME*
EDUKASI BENCANA 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:
MUHAMMAD FAHRUL AKBAR
NIM. 16650071

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN

**PENGATURAN LEVEL KESULITAN DINAMIS MENGGUNAKAN
METODE SUPPORT VECTOR MACHINE (SVM) PADA GAME
EDUKASI BENCANA GUNUNG MELETUS**

SKRIPSI

**Oleh :
MUHAMMAD FAHRUL AKBAR
NIM. 16650071**

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 10 April 2023

Pembimbing I,



Dr. Fresy Nugroho, M.T
NIP. 19710722 201 1011 001

Pembimbing II,



Dr. Yunifa Miftachul Arif, M.T
NIP. 19830616 201 1011 004

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




Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

PENGATURAN LEVEL KESULITAN DINAMIS MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM) PADA GAME EDUKASI BENCANA GUNUNG MELETUS

SKRIPSI

Oleh :
MUHAMMAD FAHRUL AKBAR
NIM. 16650071

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 12 Juni 2023

Susunan Dewan Penguji

Ketua Penguji : Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

Anggota Penguji I : Juniardi Nur Fadilah, M.T
NIP. 19920605 201903 1 015


Anggota Penguji II : Dr. Fresy Nugroho, M.T
NIP. 19710722 201 1011 001

Anggota Penguji III : Dr. Yunifa Miftachul Arif, M.T
NIP. 19830616 201 1011 004

()
()
()
()

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




Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Fahrul Akbar

NIM : 16650071

Program Studi : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : Pengaturan Level Kesulitan Dinamis Menggunakan Metode
Support Vector Machine (SVM) Pada *Game* Bencana Gunung
Meletus

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-banar merupakan hasil karya saya 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,

Yang membuat pernyataan,



Muhammad Fahrul Akbar

NIM. 16650071

MOTTO

... Passion Before Fashion ...

HALAMAN PERSEMBAHAN

Skripsi ini saya persembahkan untuk perempuan hebat yang tiada lelah memberikan segala hal kepada saya. Semoga Skripsi ini dapat sedikit mengobati rasa kecewa yang kerap kali saya berikan.

Terima kasih, Bu

KATA PENGANTAR

Syukur alhamdulillah penulis ucapkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Skripsi ini dengan baik.

Selanjutnya penulis mengucapkan terima kasih kepada semua pihak yang telah membantu terselesaikannya Skripsi ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Prof. Dr. H. M. Zainuddin MA, selaku rektor UIN Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT, selaku Ketua Prodi Teknik Informatika yang senantiasa mendorong dan memberikan solusi di setiap permasalahan mahasiswanya.
4. Dr. Fresy Nugroho, M.T dan Dr, Yunifa Miftachul Arif, M.T selaku dosen pembimbing, yang selalu memberikan ilmu pengetahuan yang tidak ada habis-habisnya, pengarahan, dukungan dan motivasi dalam penyelesaian Skripsi.

5. Dr. Fachrul Kurniawan, M.MT dan Juniardi Nur Fadilah, M.T, selaku Dosen Penguji yang telah membimbing dan memberikan masukan kepada penulis sehingga tercapai hasil skripsi yang lebih baik
6. Seluruh dosen dan staf Teknik Informatika yang telah memberikan ilmu dan pengalaman yang berharga selama masa perkuliahan.
7. Ibu, ayah, dan kakak penulis yang senantiasa memberikan kasih sayang, doa, dan dukungan berupa moril maupun materiel kepada penulis.
8. Kerabat-kerabat penulis yang tidak hanya menanyakan kapan lulus, namun juga memberikan doa dan dukungan kepada penulis.
9. Teman-teman Teknik Informatika 2016 (Andromeda) yang senantiasa saling memberi semangat dan berjuang bersama.
10. Semua pihak yang ikut membantu dalam menyelesaikan Skripsi ini yang tidak dapat penulis sebutkan satu persatu.

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. Amin Ya Rabbal Alamin.

Malang, Mei 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
مستخلص البحث.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Pernyataan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Batasan Penelitian	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terkait	6
2.2 <i>Game</i>	7
2.3 <i>Non player character</i> (NPC).....	8
2.4 Asap Kabut.....	8
2.5 <i>Taksonomi Bloom</i>	9
2.6 <i>Dynamic Difficulty Adjustment</i>	14
2.7 <i>Support Vector Machine</i> (SVM)	17
2.8 <i>System Usability Scale</i> (SUS)	23
BAB III DESAIN DAN PERANCANGAN.....	25
3.1 Alur Penelitian	25
3.3 Perancangan Sistem <i>Game</i>	26
3.3.1 Desain Menu <i>Flow</i>	26
3.3.2 <i>Gameplay Flowchart</i>	27
3.4 <i>Storyboard</i>	28
3.5 Perancangan Implementasi <i>Support Vector Machine</i> (SVM).....	29
3.5.1 Desain Sistem <i>support vector machine</i> (SVM).....	29
3.5.2 Implementasi <i>Support Vector Machine</i> (SVM)	30
3.6 Skenario Pengujian <i>System Usability Scale</i> (SUS)	43
BAB IV HASIL DAN PEMBAHASAN	45
4.1 Implementasi <i>User Interface</i>	45
4.1.1 Tampilan Awal <i>Game</i>	45
4.1.2 Tampilan <i>Gameplay</i>	46

4.1.3 Tampilan Kuis.....	46
4.1.4 Tampilan Menang	47
4.1.5 Tampilan Kalah.....	47
4.2 Implementasi <i>Support Vector Machine</i>	48
4.3 Klasifikasi Level Menggunakan SVM.....	55
4.4 Pengujian <i>System Usability Scale</i> (SUS)	57
4.5 Integrasi Sains dan Islam	59
BAB V KESIMPULAN DAN SARAN	62
5.1 Kesimpulan	62
5.2 Saran.....	62
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Densitas Citra Uji	9
Gambar 2.2 Penerapan level kesulitan pada <i>game</i>	17
Gambar 2.3 block diagram SVM	18
Gambar 2.4 Contoh Pertanyaan SUS	23
Gambar 3.1 Alur Penelitian.....	25
Gambar 3.2 <i>Gameplay flowchart</i>	28
Gambar 3.3 Block diagram SVM	30
Gambar 4.1 Tampilan Awal <i>Game</i>	45
Gambar 4.2 Tampilan <i>Gameplay</i>	46
Gambar 4.3 Tampilan Kuis	46
Gambar 4.4 Tampilan Menang	47
Gambar 4.5 Tampilan Kalah	47
Gambar 4.6 Hasil perhitungan nilai E_i	52
Gambar 4.7 Nilai <i>alpha</i> ke-i level 1	52

DAFTAR TABEL

Tabel 2.1 Aspek Kognitif.....	11
Tabel 2.2 Aspek Afektif.....	12
Tabel 2.3 Aspek Psikomotorik.....	14
Tabel 3.1 Dataset <i>Support Vector Machine</i>	31
Tabel 3.2 hasil normalisasi data	31
Tabel 3.3 Hasil dari perhitungan kernel.....	33
Tabel 3.4 definisi nilai lambda, compexity, epsilon, dan <i>alpha</i>	33
Tabel 3.5 Hasil Perhitungan <i>matrix hessian</i>	35
Tabel 3.6 Hasil perhitungan <i>Alpha</i> terbaru	38
Tabel 3.7 Daftar kuesioner SUS.....	43
Tabel 4.1 <i>Dataset</i>	48
Tabel 4.2 Hasil Normalisasi	49
Tabel 4.3 Hasil Perhitungan Kernel.....	50
Tabel 4.4 <i>Matrix Hessian</i> level 1	51
Tabel 4.5 Nilai <i>alpha</i> kelas 1 sampai kelas 9.....	54
Tabel 4.6 Nilai bias kelas 1 sampai kelas 9	54
Tabel 4.7 Kuesioner SUS.....	57
Tabel 4.8 Hasil Perhitungan Skor SUS	58

ABSTRAK

Akbar, Muhammad Fahrul. 2023. **Pengaturan Level Kesulitan Dinamis Menggunakan Metode *Support Vector Machine* (SVM) Pada *Game* Bencana Gunung Meletus**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Fresy Nugroho, M.T (II) Yunifa Miftachul Arif, M.T

Kata Kunci: Gunung Meletus, Metode *Support Vector Machine*, *Game* Edukasi

Gunung meletus adalah fenomena alam yang terjadi akibat adanya endapan magma pada perut bumi dan disebarkan oleh gas yang memiliki kekuatan tinggi. Letusan gunung berapi merupakan salah satu bencana alam yang sangat dahsyat. Di Indonesia gunung meletus merupakan salah satu bencana yang banyak ditakuti masyarakat, oleh sebab itu untuk melatih penanggulangan bencana gunung meletus maka masyarakat diberikan edukasi berupa suatu *game* bertema *adventure* yang menggambarkan persiapan ketika bencana gunung meletus akan tiba. Kelebihan *game* sebagai sarana sosialisasi untuk masyarakat yang dapat memberikan pengalaman praktis sehingga peserta dapat memahami materi tersebut dengan gampang. *Game* ini menggunakan metode *Support Vector Machine* untuk mengatur tingkat level kesulitan secara otomatis yang disesuaikan dengan kemampuan pemainnya. Data masukan pada *game* ini terdapat 7 variabel diantaranya *score*, darah pemain, waktu, jumlah barang, tipe barang, jumlah musuh dan tipe musuh. Pengujian yang dilakukan menggunakan pengujian *System Usability Scale* (SUS) dengan skor SUS sebesar 62.5%.

ABSTRACT

Akbar, Muhammad Fahrul 2023. **The Configuration Of Dynamic Difficulty Level Utilizing Support Vector Machine (SVM) Method For Educational *Game* Of a Vulcano Eruption**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Fresy Nugroho, M.T (II) Dr. Yunifa Miftachul Arif. M.T.

Volcanic eruption is a natural phenomenon that occurs due to the deposition of magma in the bowels of the earth and is emitted by gases that have high strength. Volcanic eruptions are one of the most devastating natural disasters. In Indonesia, volcanic eruption is one of the most fearest disasters by many people, therefore to train disaster management of volcanic eruptions, the community is given education in the form of an adventure-themed *game* that describes the preparations when the volcanic eruption occurs. The advantages of *games* as a means of socialization for the community that can provide practical experience so that participants can understand the material easily. This *game* uses the Support Vector Machine method to automatically adjust the difficulty level according to the player's ability. There are 7 variables including score as input data in this *game*, consisting of player blood, time, number of items, type of item, number of enemies and type of enemy. Tests are conducted utilizing *System Usability Scale* (SUS) testing with a SUS score of 62.5%.

Keywords: Vulcano Eruption, Support Vector Machine Method, Educational *Game*

مستخلص البحث

الأكبر، محمد فخر. ٢٠٢٣. ضبط الصعوبة الديناميكي باستخدام الطريقة دعم شاحنات النقل (SVM) في لعبة التعليمية الكارثة انفجار الجبل. البحث العلمي، قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفات : (1) فريزي نوجروهو، المحستير (2) يونيفا مفتاح عارف، المحستير.

الكلمات المفتاحية: ثوران بركان، دعم شاحنات النقل، اللعبة التعليمية

ثوران البركان هو ظاهرة طبيعية تحدث بسبب رواسب الصهارة في أحشاء الأرض وتنبعث من الغازات ذات القوة العالية. ثوران البركان هو واحدة من أكثر الكوارث الطبيعية تدميراً. في إندونيسيا، تعتبر ثوران بركان إحدى الكوارث التي يخشى الكثير من الناس، لتدريب إدارة ثوران بركان البركانية ثم تثقيف الجمهور في شكل ألعاب مغامرات الذي يصور الاستعدادات لموعد حدوث ثوران بركاني. يمكن أن توفر مزايا الألعاب كوسيلة للتواصل مع المجتمع خبرة عملية حتى يتمكن المشاركون من فهم المواد بسهولة. هذه اللعبة تستخدم طريقة دعم شاحنات النقل لضبط مستوى الصعوبة الذي يتم تعديله تلقائياً حسب قدرة اللاعب. إدخال البيانات في هذه اللعبة هناك 7 متغيرات بما في ذلك النتيجة وصحة اللاعب والوقت وعدد العناصر ونوع العنصر وعدد الأعداء ونوع العدو. إجراء الاختبارات باستخدام اختبار مقياس قابلية استخدام مقياس قابلية استخدام النظام (SUS) مع درجة 62.5%.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Letusan gunung berapi, bencana alam berbahaya disebabkan oleh gas bertekanan tinggi yang mengeluarkan endapan magma dari perut bumi, melibatkan pelepasan cairan pijar yang dikenal sebagai magma dari lapisan bumi pada suhu yang sangat tinggi (didit damayanti, pria wahyu RG, 2017). Seperti pada firman Allah SWT dalam alquran surah Al-mursalat ayat 10 :

وَإِذَا الْجِبَالُ نُصِبَتْ

“Dan apabila gunung-gunung telah dihancurkan menjadi debu”.

Menurut tafsir Syaikh Abdurrahman bi As’adi menjelaskan apabila gunung-gunung telah diletuskan hingga menjadi debu yang beterbangan.

Kesadaran dan pemahaman masyarakat tentang letusan gunung berapi di Indonesia memainkan peran penting dalam mengurangi ketakutan dan dampak bencana ini di kalangan penduduk yang tinggal di daerah vulkanik. Oleh karena itu, sangat penting untuk mengkomunikasikan dan mendidik masyarakat secara efektif tentang berbagai aspek dan karakteristik fenomena vulkanik (Ayu Prisma Dewi, 2017).

Sosialisasi adalah transmisi atau penerusan adat atau nilai dan aturan dari satu generasi ke generasi lainnya dalam suatu kelompok atau masyarakat. Beberapa

sosiolog menyebut sosialisasi sebagai teori peran. Karena proses sosial telah mengajarkan peran-peran yang harus dipenuhi oleh individu. Saat ini ada dua cara bersosialisasi secara offline maupun online. Dalam bersosialisasi secara offline, terdapat beberapa kendala yang dapat menghambat transfer informasi, antara lain waktu dan tempat yang tidak dapat dijangkau oleh peserta. Begitu juga sosialisasi internet di Indonesia memiliki beberapa hambatan berupa kualitas sinyal dan fokus sosialisasi tersebut pada letusan gunung berapi adalah berapa banyak orang yang berada di gunung berapi dan kebanyakan dari mereka tidak terbiasa dengan media teknologi informasi. (Rahmawan Putra & Nugroho, 2016).

Badan Nasional Penanggulangan Bencana memiliki banyak cara untuk melakukan sosialisasi dengan cara media *game*. Pada zaman modern ini *game* sangat digemari oleh halayak umum baik tua maupun muda *game* sangat banyak dimainkan. *Game* mempunyai banyak fungsi diantaranya sebagai sarana edukasi. *Game* edukasi adalah suatu model permainan yang digunakan untuk bersosialisasi agar lebih mudah difahami oleh masarakat tentang mitigasi bencana gunung meletus. Adapun kelebihan *Game* dapat memberikan pelajaran dan simulasi apabila terjadinya gunung meletus (Haryanto & Lakoro, 2019).

Selain kelebihan, *game* edukasi juga memiliki kelemahan dan kekurangan tingkat kesulitan pada *game* sangat monoton sehingga pemain menyebabkan permainan yang kurang atraktif dan monoton. Jika dilihat dari mekanisme tingkat kesulitan pada *game* saat ini cenderung menggunakan ketetapan pada nilai yang sudah di tentukan. Tentunya tingkat kesulitan tidak akan mengikuti kemampuan dari pemain itu sendiri. Sebagian besar *game* hanya memiliki tiga tingkat kesulitan

yaitu mudah, sedang dan sulit. Maka dari itu peneliti membuat *Game* yang mempunyai level permainan dan dapat mengikuti dari kemampuan pemain (Mustafa, 2016).

Pada penelitian sebelumnya tentang pengaturan level pada *game* edukasi, tingkat kesulitan diatur sesuai dengan perhitungan manual yang telah ditetapkan. Beberapa penelitian telah dilakukan untuk mengetahui tingkat kesulitan *game* edukasi. Diantaranya, dengan menggunakan metode. Kemudian bobot tersebut akan digunakan sebagai acuan dari tingkat kesulitan pada *game*. (Murdha Pamungkas, 2022)

Pada penelitian ini peneliti menggunakan metode *Support Vector Machine* dimana metode tersebut bagian dari *machine learning* yang dapat mengklasifikasikan data yang membagi sepuluh kelas dan Sembilan *hyperplane* dimana data yang paling dekat dari garis *hyperplan* pada setiap kelas akan dijadikan acuan dari tingkat kesulitan kabut pada permainan di dalam *game* dibandingkan dengan metode *term – inverse document frequency* metode *support vector machine* dapat memproses tujuh input berupa skor, darah, waktu, jumlah musuh, tipe musuh, nilai barang dan jumlah barang dengan keluaran level kabut dan pertanyaan dengan mudah karena *support vector machine* dapat membagi *datatest* menjadi sepuluh bagian sehingga proses komputasi menjadi cepat (Monika Parapat & Tanzil Furqon, 2018).

Support Vector Machines (SVM) adalah adalah suatu metode yang handal dalam menyelesaikan masalah klasifikasi data. Penggunaan model SVM mengolah data menjadi data latih digunakan dalam membentuk model SVM penggunaan

model SVM mengolah data menjadi data latih dan data uji. Data latih digunakan dalam membentuk SVM, sementara parameter bebasnya dipilih dari awal. Selanjutnya model SVM yang di hasilkan digunakan untuk mengklasifikasi data uji (Kasim & Sudarsono, 2019).

Berdasarkan latar belakang penjelasan di atas, penulis mengambil judul **“Pengaturan Level Kesulitan Dinamis Menggunakan Metode *Support Vector Machine* (SVM) Pada *Game* Edukasi Bencana Gunung Meletus”**.

1.2 Pernyataan Masalah

Berdasarkan latar belakang penelitian yang telah dijelaskan di atas, adapun rumusan masalah yang dapat diambil yaitu, seberapa tinggi skor *usability scale* dari *game* edukasi bencana gunung meletus.

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplementasikan *Support Vector Machine* dalam mengatur kesulitan level pada *game* edukasi gunung meletus.

1.4 Manfaat Penelitian

Manfaat penelitian ini untuk membuat level sehingga meningkatkan daya Tarik pada *game*. Akibatnya setiap pemain akan semakin terasah dan tertantang karena semakin tinggi level akan semakin susah.

1.5 Batasan Penelitian

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

1. *Game* ini mempunyai 7 masukan berupa score, waktu, darah, tipe musuh, jumlah musuh, tipe barang dan nilai barang
2. *Game* ini mempunyai 4 keluaran berupa tingkat kesulitan kabut.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Support Vector Machines (SVM) adalah adalah suatu metode yang handal dalam menyelesaikan masalah klasifikasi data. Penggunaan model SVM mengolah data menjadi data latih digunakan dalam membentuk model SVM penggunaan model SVM mengolah data menjadi data latih dan data uji. Data latih digunakan dalam membentuk SVM, sementara parameter bebasnya dipilih dari awal. Selanjutnya model SVM yang di hasilkan digunakan untuk mengklasifikasi data uji (Kasim & Sudarsono, 2019).

Jajang dkk menjelaskan dalam penelitiannya metode *Support Vector Machine* digunakan untuk pengelompokan data siswa yang melakukan her atau peningkatan dan juga menerapkan kernel agar bisa di rubah menjadi data visual berupa grafik, bar, dll (Purnama et al., 2020).

Helena dkk juga menggunakan *Support Vector Machine* sebagai permodelan dalam pengelompokan jenis pantun. Dalam penelitiannya Helena dkk membagi jenis pantun dalam 3 kategori yaitu pantun untuk anak-anak, pantun muda, dan pantun orang tua. Setelah data terkumpul, dilakukan tahap pra proses. Tahap pra proses ini bertujuan untuk membuat data sehingga siap untuk diolah di tahapan ekstraksi fitur. Tahap pra proses terdiri atas *text segmentation*, *text folding*, *tokenization*, *stopword removal*, dan *stemming*. Tahapan ekstraksi fitur mempunyai

tujuan untuk menggali informasi potensial serta mempresentasikan kata-kata sebagai *vector* fitur. Tahapan selanjutnya yaitu memisahkan data latih dan data uji. Kemudian pada proses klasifikasi dilakukan dengan metode *multiclass* SVM untuk mendapatkan hasil akhir dari klasifikasi pembuatan sistem (Irmada & Ria Astriratma, 2020).

2.2 Game

Game merupakan sebuah kegiatan di mana terdapat keberhasilan dalam menyelesaikan tantangan dan juga kekalahan ketidak tidak bisa menyelesaikan sebuah tantangan dan yang pasti terdapat beberapa aturan dalam menyelesaikan tantangan tersebut. Pengertian didukung oleh pendapat dari Putri dan Crissano, di dalam *game* mereka terdapat sebuah sistem yang membuat aturan permainan yang mengatur jalannya permainan dan dapat mengatur tingkah laku pemain sehingga pemain dapat berhubungan dengan sistem yang telah dibuat (Putri & Crissano, 2017).

Game memiliki kelebihan dan kekurangan. Salah satu kelebihan *game* selain sebagai hiburan adalah dapat meningkatkan cara berpikir para pemain dalam menyelesaikan masalah yang ada di dalam kehidupan nyata maupun di dalam *game*. Hal ini dijelaskan Novalendry dalam penelitiannya, dia menyebutkan bahwa dalam permainan terdapat sebuah sistem yang menuntut para pemainnya agar memanfaatkan fitur-fitur yang ada di dalam *game* untuk menyelesaikan masalah dan meningkatkan strategi untuk menyelesaikan tantangan *game* (Dony Novalendry, 2013). Namun, terdapat kekurangan yang membuat *game*, yaitu dapat

membuat para pemain ketergantungan terhadap dunia *game*, sehingga para pemain seringkali melupakan kegiatan di dunia nyata sehari-hari.

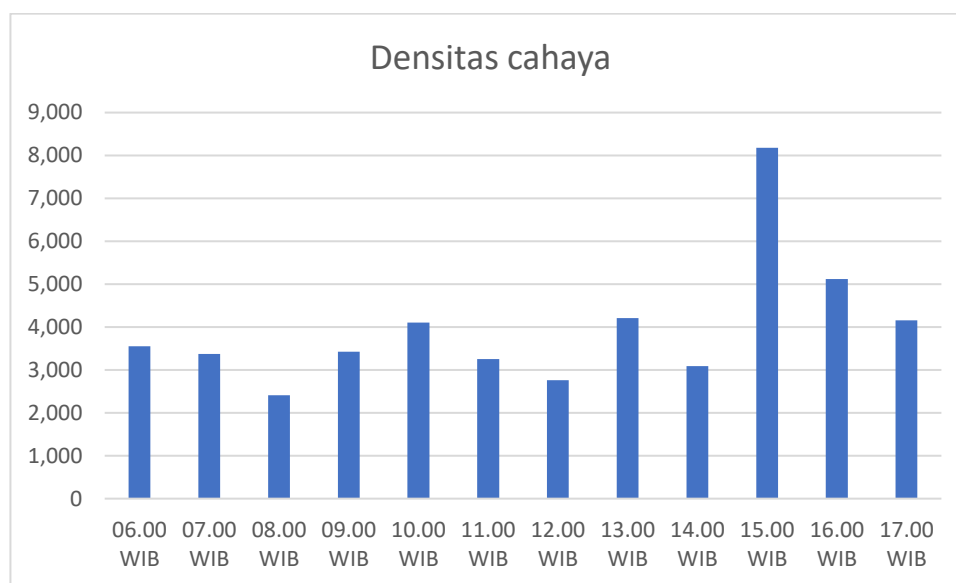
2.3 *Non player character* (NPC)

Di dalam *game* terdapat karakter yang hanya bisa di kendalikan oleh komputer, karakter ini disebut dengan *Non Player-Character* (NPC). NPC ini digunakan untuk jalannya cerita di sebuah *game*. NPC terkadang berperan sebagai musuh dari pemain dan terkadang berperan sebagai karakter yang membantu pemain dalam menyelesaikan misi dari pemain. Di dalam penerapannya, terkadang NPC dibuat dengan kecerdasan buatan agar dapat terlihat seperti pemain. Lalu, NPC ini akan mengisi peran yang kurang dalam sebuah cerita di dalam *game* (Warpefelt, 2016).

2.4 Asap Kabut

Suryani menjelaskan dalam penelitiannya bahwa suatu kondisi yang menutupi suatu wilayah dalam jangka waktu yang lama dan disebabkan oleh banyak faktor disebut dengan Asap kabut. Terdapat banyak kandungan kimia di dalam Asap kabut misalnya, misalnya nitrogen dioksida, ozon troposfer, senyawa organik yang mudah menguap (VOC). VOC dibuat oleh proses penguapan bahan kimia, sedangkan banyak nitrogen oksida dibuat dengan membakar bahan bakar fosil seperti mobil dan motor listrik. Selain nitrogen dioksida, ozon troposferik *Volatile Organic Compound* (VOC). VOC berasal dari proses penguapan bahan kimia, Sementara oksida nitrogen banyak dihasilkan dari proses pembakaran dari bahan bakar fosil seperti mesin mobil, mesin listrik (Suryani, 2012).

Studi sebelumnya menggunakan metode kebisingan Perlin, yang dilakukan setelah perhitungan densitas (kedalaman) untuk memberikan simulasi yang lebih realistis. Studi ini dihitung secara matematis di Matlab untuk menghasilkan nebula sintetis heterogen dalam 2D, yang dilanjutkan dengan Unity 3D untuk menghasilkan nebula 3D sintetis heterogen.(Giantara et al., 2020).



Gambar 2.1 Densitas Citra Uji

Dalam studi ini, estimasi transmisi Transmission map digunakan ketika estimasi transmisi gambar kabut diperlukan untuk membuat pemandangan kabut.

2.5 Taksonomi Bloom

Taksonomi Bloom terdiri dari pengetahuan, pemahaman, aplikasi, analisis, sintesis dan evaluasi. Selanjutnya, *Taksonomi Bloom* direvisi dan menambahkan beberapa komponen seperti perubahan istilah pada tiap tingkat, pengaturan tingkat yang lebih detail dan penambahan jumlah dimensi. Perubahan ini telah diterapkan pada desain *e-learning* yang mengandung instruksi secara *online*. *E-learning* ini

berbasis objek dan keluaran pembelajaran. Terkait desain yang diajukan terdapat 3 aspek pembelajaran terpadu yaitu aspek kognitif, aspek psikomotorik dan aspek afektif (Nugroho et al., 2019). Pada dunia pendidikan *Taksonomi Bloom* bermaksud mempermudah guru membuat klasifikasi apa saja yang harus dipelajari anak didiknya dalam waktu tertentu. Untuk merespon tuntutan pendidikan *Taksonomi Bloom* memperhatikan perkembangan kontemporer dalam bidang terkait pendidikan (Ruwaida, 2019). Berikut merupakan penjelasan dari aspek kognitif, aspek psikomotorik dan aspek afektif:

1. Aspek Kognitif

Aspek kognitif merupakan ranah yang umumnya dikaitkan dengan proses belajar. Kognisi adalah kemampuan psikis atau mental untuk mengamati, melihat, menyangka, memperhatikan, menduga dan menilai (Abdurakhman & Rusli, 2015). Dalam aspek kognitif, sejauh mana peserta didik dan pada level yang lebih atas seorang peserta didik mampu menguraikan kembali kemudian memadukannya dengan pemahaman yang sudah ia peroleh untuk diberi penilaian atau pertimbangan. Berikut ini merupakan tabel keluaran pertanyaan dari aspek kognitif berdasarkan hasil wawancara dari Lutfi Arisatun Niswah sebagai guru SDN 4 Kepanjen yang menyatakan bahwa tingkatan pertanyaan kognitif dibagi menjadi 3 tingkatan yaitu *low*, *medium*, *high* yang mana pada pada tingkatan *low* terdapat 4 tingkatan, *medium* 3 tingkatan dan *high* 3 tingkatan

Tabel 2.1 Aspek Kognitif

Level	Output Pertanyaan	Kognitif
1	Apa yang kamu ketahui tentang gunung meletus?	K1-1
2	Berapakah jumlah gunung meletus di Indonesia?	K1-2
3	Sebutkan nama gunung meletus di daerah anda?	K1-3
4	Sebutkan tipe gunung meletus di indonesia?	K1-4
5	Bagaimana ciri ciri gunung meletus ketika erupsi?	K2-1
6	Apa nama lembaga yang memantau gunung berapi di Indonesia?	K2-2
7	Apa nama gunung berapi terbesar di Indonesia?	K2-3
8	Sebutkan nama gunung berapi di pulau Sumatera?	K3-1
9	Sebutkan nama gunung berapi di pulau Jawa?	K3-2
10	Sebutkan gunung berapi di pulau sulawesi?	K3-3

2. Aspek Afektif

Aspek Afektif merupakan ranah yang berkaitan dengan perasaan seorang individu. Seorang siswa tidak mungkin menunjukkan sikap dan minat yang positif terhadap suatu mata pelajaran tertentu akan sulit untuk mencapai prestasi yang optimum pada mata pelajaran tersebut (Nurhidayati & Sunarsih, 2013). Dalam ranah ini peserta didik dinilai sejauh mana ia mampu menginternalisasikan nilai – nilai pembelajaran ke dalam dirinya. Ranah ini erat kaitannya dengan tata nilai dan konsep diri. Berikut ini merupakan tabel keluaran pertanyaan dari aspek afektif berdasarkan hasil wawancara dari Lutfi Arisatun

Niswah sebagai guru SDN 4 Kepanjen yang menyatakan bahwa tingkatan pertanyaan afektif dibagi menjadi 3 tingkatan yaitu *low*, *medium*, *high* yang mana pada pada tingkatan low terdapat 4 tingkatan, medium 3 tingkatan dan high 3 tingkatan :

Tabel 2.2 Aspek Afektif

Level	Output Pertanyaan	Afektif
1	Bagaimana sikapmu jika keluargamu terdampak erupsi gunung meletus?	A1-1
2	Bagaimana sikapmu jika temanmu terdampak erupsi gunung meletus?	A1-2
3	Bagaimana sikapmu jika saudaramu terdampak erupsi gunung meletus?	A1-3
4	Bagaimana sikapmu jika melihat orang yang mengabaikan peringatan gunung meletus?	A1-4
5	Bagaimana sikapmu jika melihat orang yang tidak peduli akan terjadinya gunung meletus?	A2-1
6	Bagaimana sikapmu jika melihat warga desa sedang bergotong-royong?	A2-2
7	Apa yang kamu rasakan jika ada warga yang enggan bergotong-royong membantu korban bencana erupsi gunung meletus?	A2-3
8	Apa yang kamu rasakan jika kamu kehilangan keluargamu karena bencana erupsi gunung meletus ?	A3-1

Level	Output Pertanyaan	Afektif
9	Apa yang kamu rasakan jika desamu dikabarkan akan terjadi erupsi gunung meletus?	A3-2
10	Apa yang kamu rasakan jika saudaramu terkena erupsi gunung berapi?	A3-3

3. Aspek Psikomotorik

Aspek psikomotorik merupakan ranah yang berkaitan dengan aspek – aspek keterampilan yang melibatkan fungsi sistem saraf dan otot dan berfungsi psikis. Ranah ini terdiri dari kesiapan, peniruan, membiasakan, menyesuaikan dan menciptakan (Nurhidayati & Sunarsih, 2013). Ketika peserta didik telah memahami dan menginternalisasikan nilai – nilai mata pelajaran dalam dirinya, maka tahap selanjutnya adalah bagaimana peserta didik mampu mengaplikasikan pemahamannya dalam kehidupan sehari – hari melalui perbuatan atau tindakan. Berikut ini merupakan tabel keluaran pertanyaan dari aspek psikomotorik berdasarkan hasil wawancara dari Lutfi Arisatun Niswah sebagai guru SDN 4 Kepanjen yang menyatakan bahwa tingkatan pertanyaan psikomotorik dibagi menjadi 3 tingkatan yaitu *low*, *medium*, *high* yang mana pada pada tingkatan low terdapat 4 tingkatan, medium 3 tingkatan dan high 3 tingkatan :

Tabel 2.3 Aspek Psikomotorik

Level	Output Pertanyaan	Psikomotorik
1	Apa yang kamu lakukan jika mendengar kabar akan terjadi bencana erupsi gunung meletus?	P1-1
2	Apa yang kamu lakukan saat terjadi bencana gunung berapi?	P1-2
3	Apa yang kamu lakukan jika mendengar akan terjadi bencana gunung berapi di rumah saudaramu?	P1-3
4	Apa yang kamu lakukan setelah terjadi erupsi gunung meletus?	P1-4
5	Apa usaha yang dapat kamu lakukan untuk membantu tim sar dalam menangani bencana gunung meletus?	P2-1
6	Apa yang kamu lakukan ketika temanmu mengajakmu membantu korban bencana erupsi gunung meletus?	P2-2
7	Apa yang kamu lakukan ketika melihat kotak sumbangan bencana erupsi gunung meletus?	P2-3
8	Apa yang kamu lakukan ketika melihat korban bencana erupsi gunung meletus yang butuh selimut?	P3-1
9	Apa yang kamu lakukan jika melihat korban bencana erupsi gunung meletus yang kelaparan?	P3-2
10	Apa yang kamu rasakan ketika keluargamu terdampak bencana erupsi gunung meletus?	P3-3

2.6 Dynamic Difficulty Adjustment

Dynamic Difficulty Adjustment (DDA) dikenal sebagai *Auto Dynamic Difficulty* dan *Dynamic Game Balancing*. Metode ini adalah teknik dimana penyesuaian kemampuan level, skenario dan perilaku di dalam *game* dapat berubah secara *real-time* atau dinamis berdasarkan kemampuan dan keterampilan *player*. Penerapan metode *Dynamic Difficulty Adjustment* tidak sering digunakan di dalam *game*, padahal ketika diimplementasikan dengan benar metode ini dapat bermanfaat bagi keseluruhan pengalaman bermain bagi *player*, dimana

menawarkan pengalaman baru setiap kali dimainkan dan menghindari *player* merasa bosan jika *game* terlalu mudah atau frustrasi jika *game* terlalu sulit (Colwell & Glavin, 2018). Tujuan utama dari metode *Dynamic Difficulty Adjustment (DDA)* adalah untuk membuat *player* tertarik dari awal *game* hingga akhir *game* dan memberikan sebuah tingkat kesulitan yang bagus di seluruh level.

Pada metode *Dynamic Difficulty Adjustment (DDA)* untuk menghitung *skill* (efF_i) pada *player* dan musuh berdasarkan nilai atribut saat ini dalam *game* ($Fval_i$) serta nilai maksimum (F_{max}) dan minimum (F_{min}) dari setiap atribut digunakan rumus (Shena, 2019) sebagai berikut:

$$efF_i = \begin{cases} \frac{Fval_i - Fmin_i}{Fmax_i - Fmin_i} \\ \frac{Fmax_i - Fval_i}{Fmax_i - Fmin_i} \end{cases} \quad (2.1)$$

Keterangan:

efF_i = *skill* pada *player*

$Fval_i$ = nilai atribut saat ini

$Fmax_i$ = nilai maksimum

$Fmin_i$ = nilai minimum

Untuk menentukan nilai total *skill* (ef) pemain atau musuh digunakan rumus sebagai berikut:

$$ef = \frac{\sum_{i=1}^n (efF_i * weight_i)}{\sum_{i=1}^n weight_i} \quad (2.2)$$

Rumus (2) menghitung semua atribut nilai keterampilan dengan bobot masing-masing atribut. Bobot ini merepresentasikan pentingnya atribut dalam mendefinisikan keterampilan pemain. Bobot yang digunakan sebagai basis atau

proses dinamis adalah atribut skor atau kekuatan memiliki bobot 0.817 dan atribut kecepatan memiliki bobot 0.8124

Nilai ini berdasarkan penelitian yang dilakukan oleh (Lee 2019.) Hasil penelitian menunjukkan bahwa bobot dari atribut skor adalah 0.8176, sedangkan bobot dari atribut musuh adalah 1 dikurangi 0.8176 yaitu 0.8124.

Dalam membandingkan skill musuh dan pemain dapat menggunakan rumus 2.3 dimana perbandingan tersebut digunakan untuk penyesuaian level. ef_p adalah nilai skill pemain dan ef_0 adalah nilai skill musuh yang dihitung menggunakan rumus 2.2.

$$diffef = |ef_p - ef_0| \quad (2.3)$$

Untuk membandingkan penyesuaian kesulitan dapat menggunakan rumus 2.4 dimana $diffef$ adalah perbedaan nilai antara player dan musuh. Jika nilainya lebih tinggi dari presentase *skill* musuh (P_{lim}) maka pemain Akan naik level.

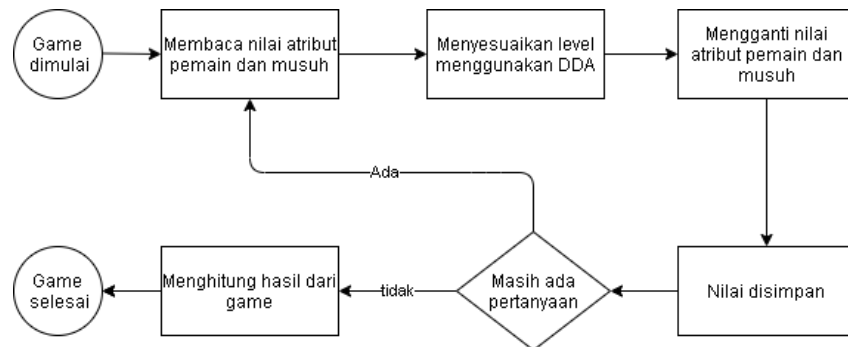
$$diffef > P_{lim} * ef_p \quad (2.4)$$

Sedangkan untuk menghitung penyesuaian atribut pada *game* dapat dihitung dengan rumus 2.5 dan rumus 2.6 dimana BVPB merupakan atribut yang merepresentasikan ketika nilai pemain lebih besar. Sedangkan SVPB merupakan atribut yang merepresentasikan ketika nilai pemain lebih kecil.

$$adjF_i = \begin{cases} diffef * (F_{maxi} - F_{mini}) & \text{for BVPB} \\ -diffef * (F_{maxi} - F_{mini}) & \text{for SVPB} \end{cases} \quad (2.5)$$

$$Fval_{oi} = \begin{cases} Fvalold_{oi} + adjF_i & \text{for } ef_p > ef_0 \\ Fvalold_{oi} - adjF_i & \text{for } ef_p < ef_0 \end{cases} \quad (2.6)$$

Berikut merupakan gambaran dari metode *dynamic difficulty adjustment* yang akan di jelaskan melalui gambar 2.1.



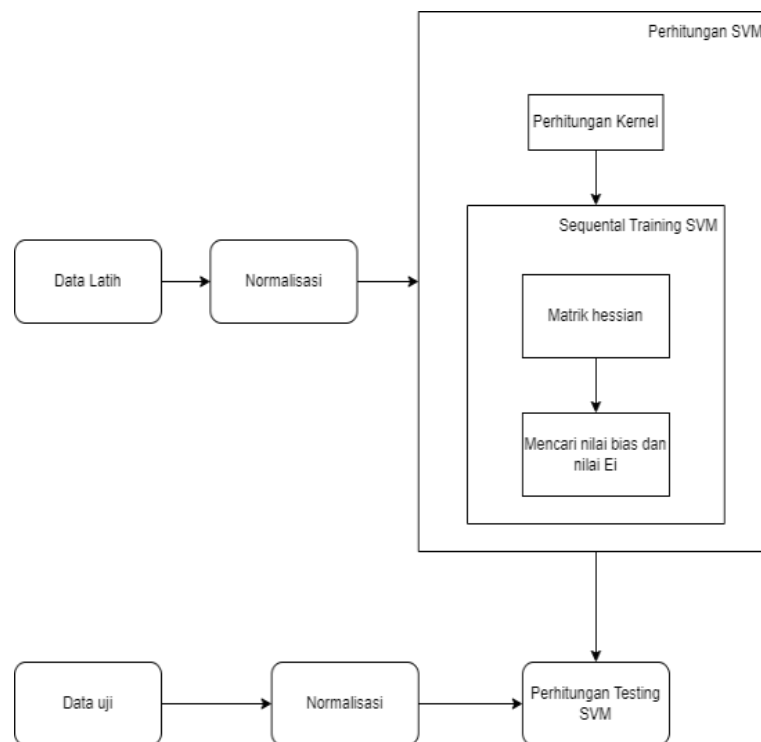
Gambar 2. 2 Penerapan level kesulitan pada *game*

Langkah awal pada gambar 2.1 adalah membaca nilai atribut player dan enemy kemudian penyesuaian level menggunakan pengaturan level kesulitan. Selanjutnya nilai atribut player dan enemy disesuaikan dengan nilai baru berdasarkan level. Proses tersebut akan diulangi jika ada pertanyaan dan nilai akan dihitung jika tidak ada pertanyaan.

2.7 Support Vector Machine (SVM)

Support Vector Machine bermula dari teori pembelajaran statistika, yang mana perhitungan akhirnya memiliki hasil yang lebih baik daripada metode statistika yang lain. Saat pembangunan model klasifikasi dengan SVM, terdapat Satu set data pelatihan yang dipilih dalam proses pelatihan yang sedang dipelajari. Kemudian ada juga sebagian kecil dari data pelatihan yang disimpan untuk pelatihan. Digunakan dalam proses prediksi. Jadi dalam proses iterasi pelatihan SVM Tidak semua data pelatihan disertakan. Inilah yang membuat SVM menonjol, yang disebut algoritme ini *support vector machine*, di mana vektor atau data pendukungnya adalah Ini berkontribusi pada proses pelatihan dan prediksi (Prasetyo, 2014).

Gambar 2.2 merupakan alur dari algoritma *support vector machine*. *Input* dari algoritma ini merupakan *dataset*. Selanjutnya Akan di bagi menjadi 2 yaitu data latih dan data uji. Pada data latih Akan di lakukan perhitungan data training. Selanjutnya pada data training di lakukan klasifikasi level yang Akan menghasilkan output berupa level.



Gambar 2.3 block diagram SVM

Rumus yang di gunakan untuk perhitungan metode *support vector machine* adalah sebagai berikut:

1. Normalisasi

Normalisasi digunakan supaya nilai antar fitur berada pada interval yang Sama. Min-Max Normalization adalah metode normalisasi yang

sering di gunakan untuk mengatasi permasalahan nilai antar fitur yang memiliki jarak yang terlalu jauh (Larose, 2005).

$$X^* = \frac{X - \min(x)}{\max(x) - \min(x)} \quad (2.7)$$

Keterangan:

X^* = hasil normalisasi data.

X = data yang belum di normalisasi.

$\text{Min}(x)$ = nilai minimum dari semua data.

$\text{Max}(x)$ = nilai maksimum dari semua data.

2. Kernel

Fungsi kernel berguna untuk pemetaan fitur yang lama pada fitur yang baru. Penggunaan fungsi kernel pada SVM dapat juga disebut dengan *kernel trick*. Di bawah ini merupakan beberapa pilihan dalam menentukan fungsi kernel yang Akan di gunakan pada *support vector machine*:

a. Kernel linier

Kernel linier adalah fungsi kernel yang paling sederhana. Kernel linier digunakan ketika data yang dianalisis sudah terpisah secara linier.

$$K(x, y) = x \cdot y \quad (2.8)$$

Keterangan:

$K(x, y)$ = hasil dari perhitungan kernel

X = data ke-x

Y = data ke-y

b. Kernel Polynomial

Kernel polynomial digunakan ketika data yang di gunakan tidak terpisah secara linier.

$$K(x, y) = (x \cdot y + C)^d \quad (2.9)$$

Keterangan:

$K(x, y)$ = hasil perhitungan kernel.

X = data ke-x

Y = data ke-y

C = Complexity

3. *Sequential Training SVM*

Salah satu proses training SVM adalah *sequential training*. Dengan proses tersebut dibutuhkan waktu yang lebih cepat dalam permasalahan karena memiliki algoritma yang sederhana. Berikut urutan proses *sequential training*.

a. Melakukan inisialisasi a_i dan parameter lain yaitu, $\lambda, \gamma, C, \varepsilon$, keterangan:

a_i = alfa, berguna mencari *support vector*.

λ = Lambda

γ = gamma, digunakan untuk mencari kecepatan learning

C = *complexity C*, nilai kompleksitas, digunakan untuk membatasi nilai alfa pada saat proses training

ε = epsilon, digunakan dalam mengukur error pada klasifikasi.

b. Melakukan perhitungan *matrix hessian*

Matrix hessian adalah perkalian antara kernel dengan nilai Y. nilai

Y yaitu nilai berupa vector yang berisi nilai 1 dan -1.

$$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2) \quad (2.10)$$

Keterangan:

y_i = kelas data ke-i

y_j = kelas data ke-j

λ = lamda yang telah di inisialisasi

$K(x_i x_j)$ = fungsi kernel / hasil dari perhitungan kernel

- c. Mencari E_i , δa , dan a_i pada data yang dipakai mulai dari data ke-i sampai ke-j dengan rumus di bawah ini:

$$E_i = \sum_{j=1}^n a_i D_{ij} \quad (2.11)$$

Keterangan:

E_i = eror rate

D_{ij} = matriks hessian

a_i = alfa ke-i

$$\delta a_i = \min \{ \max \{ \gamma(1 - E_i), a_i \}, C - a_i \} \quad (2.12)$$

Keterangan:

a_i = alfa ke-i

γ = gamma

E_i = eror rate

C = nilai kompleksitas

$$a_i = a_i + \delta a_i \quad (2.13)$$

Keterangan:

a_i = alfa ke-i

δa_i = delta alfa ke-i

4. Nilai Bias

Rumus untuk mencari nilai bias sebagai berikut:

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-) \quad (2.14)$$

Keterangan:

$w \cdot x^+$ = nilai bobot untuk kelas positif

$w \cdot x^-$ = nilai bobot untuk kelas negative

Adapun rumus yang digunakan untuk mencari bobot vektor adalah sebagai berikut:

$$w = \sum_{i=1}^n a_i y_i x_i \quad (2.15)$$

Keterangan:

a_i = nilai bobot data ke-i

y_i = kelas data ke-i

x_i = data ke-i

5. Testing SVM

Rumus yang digunakan untuk menghitung testing SVM adalah sebagai berikut:

$$f(x) = \sum_{i=1}^m a_i y_i k(x_i, x) + b \quad (2.16)$$

Keterangan:

a_i = alfa ke-i

y_i = kelas data latih ke-i

$k(x_i, x)$ = fungsi kernel yang dipakai

b = nilai bias

2.8 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode yang telah di kembangkan oleh John Brooke pada tahun 1986. Pada saat itu Joh Brooke menggunakan SUS sebagai metode yang digunakan untuk mengukur kegunaan system aplikasi dari segi yang subjektif (H.N et al., 2015).

Kode	Item Pernyataan
R1	Saya akan sering menggunakan/mengunjungi situs ini
R2	Saya menilai situs ini terlalu kompleks (memuat banyak hal yang tidak perlu)
R3	Saya menilai situs ini mudah dijelajahi
R4	Saya membutuhkan bantuan teknis untuk menggunakan/menjelajahi situs ini
R5	Saya menilai fungsi/fitur yang disediakan pada situs ini dirancang dan disiapkan dengan baik
R6	Saya menilai terlalu banyak inkonsistensi pada situs ini
R7	Saya merasa kebanyakan orang akan mudah menggunakan/menjelajahi situs ini dengan cepat
R8	Saya menilai situs ini sangat rumit untuk dijelajahi
R9	Saya merasa sangat percaya diri menjelajahi situs ini
R10	Saya perlu belajar banyak hal sebelum saya dapat menjelajahi situs ini dengan baik

Gambar 2.4 Contoh Pertanyaan SUS
(Sumber: (H.N et al., 2015))

SUS terdiri dari beberapa pertanyaan, lalu pertanyaan itu memiliki 5 skala Likert. Responden diminta untuk memberi penilaian antara 1 sampai 5. Setiap item pernyataan memiliki skor kontribusi. Setiap skor kontribusi item akan berkisar antara 0 hingga 4. Untuk item 1,3,5,7, dan 9 skor kontribusinya adalah posisi skala dikurangi 1. Untuk item 2,4,6,8, dan 10, skor kontribusinya adalah 5 dikurangi posisi skala. Kalikan jumlah skor kontribusi dengan 2.5 untuk mendapatkan nilai

keseluruhan system usability. Skor SUS berkisar dari 0 hingga 100 (Brooke, 2020).

Berikut rumus perhitungan skor SUS:

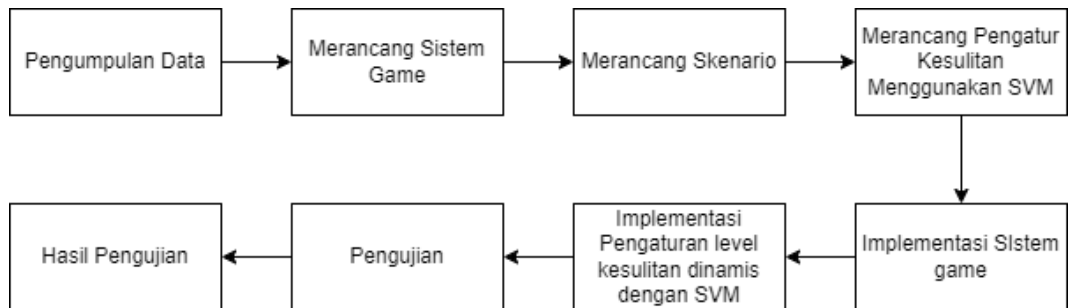
$$\begin{aligned} \text{skor SUS} = & ((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + \\ & (5 - R6) + (R7 - 1) + (5 - R8) + (R9 - 1) + (5 - R20)) \end{aligned} \quad (2.17)$$

Lalu skor SUS keseluruhan diperoleh dari rata-rata skor individual.

BAB III

DESAIN DAN PERANCANGAN

3.1 Alur Penelitian



Gambar 3.1 Alur Penelitian

Penelitian ini memiliki alur sebagai berikut:

1. Penelitian ini dimulai dengan pengumpulan data sebagai dasar dari data latih dan data uji.
2. Setelah data terkumpul, penelitian ini berlanjut dengan merancang sistem *game*. Sistem *game* ini yang dirancang disini berupa *menu flow* dan *flowchart game*.
3. Setelah alur *game* tergambar, tahap selanjutnya adalah merancang skenario *game*. Rancangan skenario *game* ini berupa *storyboard game*.
4. Selanjutnya membuat rancangan pengaturan level kesulitan dinamis menggunakan *Support Vector Machine*.
5. Setelah rancangan dibuat maka pada tahap ini menerapkan sistem *game* yang telah di buat pada poin ke 2 dan 3.

6. Selanjutnya menerapkan pengatur level kesulitan dinamis yang telah dibuat pada poin ke 4.
7. Setelah menerapkan poin 6, maka langkah terakhir dilakukan pengujian *System Usability Scale* (SUS) terhadap *game* yang telah buat.
8. Lalu hasil pengujian merupakan Skor SUS dari pengujian SUS pada poin 7

3.3 Perancangan Sistem *Game*

Pada penelitian ini terdapat beberapa perancangan dalam pembuatan *game*. Rancangan pertama yaitu *Desain Menu flow* yaitu desain yang mengatur halaman apa saja yang akan tersedia pada *game*. Selanjutnya *Flowchart game*,

3.3.1 Desain Menu *Flow*

Desain menu flow dimanfaatkan untuk mempermudah pembuatan main menu yang akan digunakan di dalam *game*. Rancangan menu flow yang diajukan pada penelitian ini adalah sebagai berikut. Selanjutnya *Flowchart game* yaitu alur dari jalannya *game* yang telah di buat.

1. *Title Screen*
 - a. *Play with SVM*, untuk menghubungkan *game* dengan pengatur level kesulitan menggunakan SVM.
 - b. *Play without SVM*, untuk menghubungkan *game* tanpa menggunakan SVM.
 - c. *Setting*, Menghubungkan ke tampilan setting.
 - d. *Quit*, untuk keluar dari *game*.

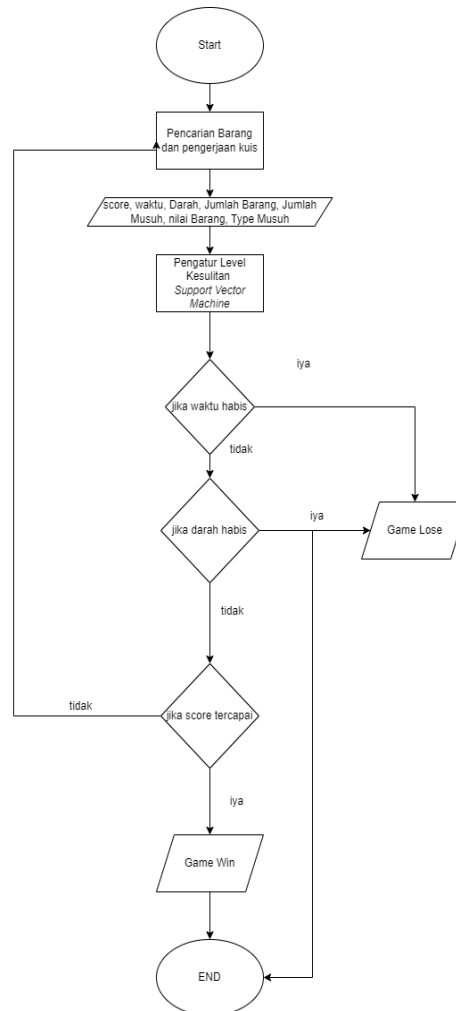
2. *Game Playing*

3. *Game Pause*

3.3.2 *Gameplay Flowchart*

1. *Game* dimulai
2. Pemain wajib mengumpulkan barang dan mengerjakan kuis untuk mendapatkan poin
3. Seiring dengan berjalannya permainan, akan di ambil 7 nilai yang akan di jadikan parameter untuk pengatur level otomatis, yaitu score, waktu, darah, jumlah barang, nilai barang, jumlah musuh, dan total musuh. Lalu nilai tersebut akan di kirimkan untuk mengatur level.
4. Ketika nilai tersebut telah di kirim, nilai input akan di kirim ke api yang berisi perhitungan metode SVM untuk di hitung menggunakan *support vector machine one against one* untuk mengatur level kesulitan.
5. Lalu seiring berjalannya permainan, *game* akan dinyatakan kalah ketika waktu permainan sudah habis, namun ketika waktu masih ada maka *game* masih bisa di mainkan.
6. Selain waktu permainan, Ketika darah pemain sudah habis maka *game* dinyatakan kalah, sebaliknya ketika darah masih tersedia maka *game* masih bisa berlanjut.
7. Lalu *game* akan dinyatakan menang, Ketika pemain telah mendapatkan skor yang telah di tetapkan oleh sistem.

Berdasarkan alur pada poin 1 sampai 7 maka dapat di gambarkan sebuah flowchart seperti pada Gambar 3.4.



Gambar 3.2 *Gameplay flowchart*

3.4 Storyboard

Game ini menceritakan pendaki yang akan bersiap-siap untuk mendaki gunung kelud, pendaki tersebut perlu mengumpulkan barang-barang yang akan digunakan untuk mendaki di hutan. Di dalam hutan pendaki akan berhadapan dengan hewan-hewan berbahaya. Apabila menyentuh hewan tersebut, pendaki akan

terluka. Dalam mencari barang, pendaki harus cepat karena waktu pencarian barang terbatas oleh waktu. Pada *game* ini, pemain akan berperan sebagai pendaki yang mencari barang-barang tersebut.

3.5 Perancangan Implementasi *Support Vector Machine* (SVM)

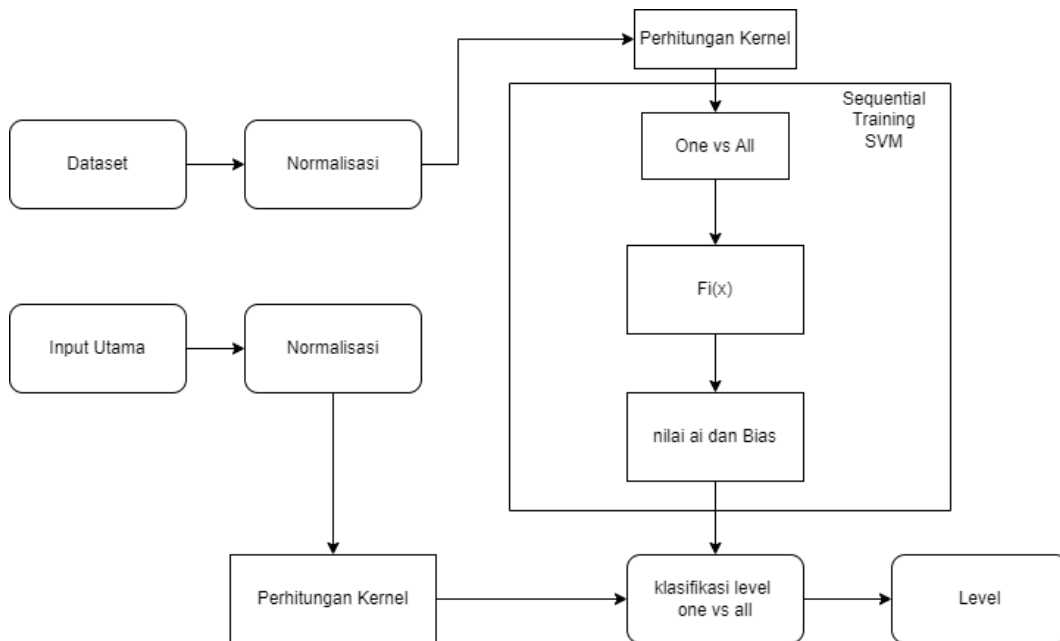
Support vector machine dirancang sebagai algoritma yang akan diimplementasikan pada pengatur kesulitan, supaya pengatur kesulitan lebih dinamis di banding tanpa *support vector machine*.

3.5.1 Desain Sistem *support vector machine* (SVM)

Penelitian ini menggunakan *support vector machine* sebagai algoritma penentu utama nilai penyeimbang kesulitan untuk *game* balancing. Proses kerja *support vector machine* dalam penelitian ini adalah sebagai berikut:

1. Input utama dalam algoritma ini adalah score, waktu, darah, jumlah barang, jumlah musuh, nilai barang, tipe musuh.
2. Selanjutnya setiap input utama akan di normalisasi agar setiap nilai memiliki interval yang sama.
3. Hasil dari normalisasi akan di gunakan untuk proses klasifikasi menggunakan SVM.

Gambar di bawah ini merupakan block diagram dari algoritma *Support Vector machine*.



Gambar 3.3 Block diagram SVM

3.5.2 Implementasi *Support Vector Machine* (SVM)

Pada implementasi algoritma *support vector machine* dibutuhkan pelatihan data latih terlebih dahulu data latih yang akan di gunakan pada implementasi ini adalah hasil dari perhitungan tanpa SVM yang terdapat pada tabel 3.3. Adapun proses yang dilakukan seperti pada gambar 3.5.

1. Dataset

Pada tahap pertama dalam algoritma SVM adalah menyiapkan dataset yang akan digunakan sebagai *training* dari metode SVM. Pada penelitian ini dataset berasal dari pengatur level kesulitan tanpa menggunakan SVM.

Tabel 3.1 Dataset *Support Vector Machine*

No	Score	Waktu	Darah	jumlah barang	jumlah musuh	nilai barang	type musuh	Level
1	65	260	68	5	1	6	2	3
2	10	280	100	5	2	10	1	1
3	40	270	50	5	3	8	2	2
4	50	181	43	15	4	6	3	5
5	60	235	85	10	7	6	2	3
6	120	164	21	30	1	10	4	6
7	80	209	50	10	6	8	3	4
8	60	214	27	2	2	6	2	3
9	75	190	74	20	1	6	3	4
10	90	179	91	20	9	6	3	5

2. Normalisasi data

Pada selanjutnya perhitungan data latih adalah normalisasi data tahap ini menyederhanakan nilai agar jarak antar nilai sama antara 0 sampai 1. Data yang di gunakan ialah data pada tabel 3.3 dan rumus yang digunakan terdapat pada rumus 3.2.

$$X^* = \frac{X - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

Selanjutnya dilakukan perhitungan normalisasi data berdasarkan rumus 3.2 yang akan menghasilkan nilai pada tabel 3.4.

Tabel 3.2 hasil normalisasi data

Score	Waktu	Darah	jumlah barang	jumlah musuh	nilai barang	type musuh
0	1	1	0.1	0.1	1	0
0.2	0.9	0.4	0.1	0.3	0.5	0.3
0.3	0.8	0.6	0.1	0	0	0.3
0.5	0.6	0.8	0.3	0.8	0	0.3
0.8	0.4	0.1	0	0.1	0	0.3
0.8	0.4	0.4	0.3	0.6	0.5	0.7
0.9	0.2	0.7	0.6	0	0	0.7

Score	Waktu	Darah	jumlah barang	jumlah musuh	nilai barang	type musuh
0.9	0.1	0.3	0.5	0.4	0	0.7
0.9	0.1	0.9	0.6	1	0	0.7
1	0	0	1	0	1	1

3. Perhitungan Kernel

Setelah normalisasi data di dapatkan langkah selanjutnya ialah menghitung nilai kernel. Fungsi kernel berguna untuk pemetaan fitur yang lama pada fitur yang baru. Dan rumus yang di gunakan untuk menghitung nilai kernel menggunakan rumus 3.3.

$$K(x, y) = (x \cdot y + C)^d \quad (3.2)$$

Dari rumus 3.3 maka di lakukan perhitungan kernel, di bawah ini merupakan proses perhitungan kernel menggunakan rumus 3.3.

$$\begin{aligned} K(1,1) &= (0 \times 0) + (1 \times 1) + (1 \times 1) + (0.1 \times 0.1) + (0.1 \times 0.1) + (1 \times 1) + (0 \times 0) \\ &= 54.71 \end{aligned}$$

$$\begin{aligned} K(1,2) &= (0 \times 0.2) + (1 \times 0.9) + (1 \times 0.4) + (0.1 \times 0.1) + (0.1 \times 0.3) + (1 \times 0.5) \\ &\quad + (0 \times 0.3) = 44.89 \end{aligned}$$

Pada proses perhitungan diatas merupakan proses perhitungan $k(1,1)$ dan $k(1,2)$ artinya hasil dari perhitungan antara kolom 1 dengan baris 1 dan hasil perhitungan antara kolom 1 dan baris 2. Perhitungan tersebut dilanjutkan sampai perhitungan antara maximal kolom yaitu $k(10,10)$ sehingga terbentuk sebuah matrik seperti pada tabel 3.5.

Tabel 3.3 Hasil dari perhitungan kernel

	1	2	3	4	5	6	7	8	9	10
1	54.76	44.89	36	49	26.01	50.41	40.96	38.44	56.25	51.84
2	44.89	36.482	28.73	40.96	20.794	43.034	34.574	32.262	48.72	44.89
3	36	28.73	21.996	33.063	15.524	35.284	27.773	25.705	40.577	37.21
4	49	40.96	33.063	46.923	26.01	50.41	41.603	39.063	57.003	53.29
5	26.01	20.794	15.524	26.01	11.834	29.594	23.232	21.344	35.046	32.49
6	50.41	43.034	35.284	50.41	29.594	55.354	46.512	43.824	62.726	59.29
7	40.96	34.574	27.773	41.603	23.232	46.512	38.564	36.12	53.436	50.41
8	38.44	32.262	25.705	39.063	21.344	43.824	36.12	33.756	50.552	47.61
9	56.25	48.72	40.577	57.003	35.046	62.726	53.436	50.552	70.728	67.24
10	51.84	44.89	37.21	53.29	32.49	59.29	50.41	47.61	67.24	64

4. *Sequential Training*

Pada tahap selanjutnya yaitu proses perhitungan *sequential training* yaitu proses training dari data latih yang ada. Tahap awal dari proses ini adalah mendefinisikan lambda (λ), nilai *complexity* (C), gamma (γ), *alpha* (α). Nilai tersebut di pilih secara random. Berikut definisi nilai lambda (λ), nilai *complexity* (C), gamma (γ), *alpha* (α).

Tabel 3.4 definisi nilai *lambda*, *complexity*, *epsilon*, dan *alpha*

<i>Lamda</i>	<i>Gamma</i>	<i>Complexity</i>	<i>Epsilon</i>	<i>Alpha</i>
0.5	0.5	1	0.001	0.001

Pada penelitian ini, *game* edukasi menggunakan klasifikasi level antara level 1 sampai level 10 sehingga perlu menerapkan klasifikasi SVM *one vs all*. One vs all merupakan metode klasifikasi SVM yang digunakan untuk klasifikasi data lebih dari satu kelas. Pada tahap pertama yang digunakan adalah inisialisi kelas dari setiap data. Karena *game* edukasi ini mengklasifikasi data antara 1 sampai 10, maka di perlukan perhitungan SVM kelas 1 ($f^1(x)$), perhitungan SVM kelas 2 ($f^2(x)$) sampai dengan

perhitungan SVM kelas 9 ($f^9(x)$). Pada kelas data (y) pada $f^1(x)$, labale level 1 yang ada pada dataset akan di kategori kan kelas positif(1) dan label level selain 1, kelas datanya akan di kategorikan menjadi kelas negatif(-1).

Setelah menentukan kelas data di setiap dataset, Langkah selanjutnya adalah menghitung *matrix hessian*. Rumus yang di pakai untuk menghitung adalah rumus 3.4.

$$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2) \quad (3.3)$$

Keterangan:

y_i = kelas data ke-i

y_j = kelas data ke-j

λ = lamda yang telah di inisialisasi

$K(x_i x_j)$ = fungsi kernel / hasil dari perhitungan kernel

Selanjutnya dari rumus 3.4 dilakukan proses perhitungan *matrix hessian*. Proses perhitungan matrix hessian antara lain:

$$D_{11} = -1 \times -1 \times (54.71 + 0.5^2) = 55.01$$

$$D_{12} = -1 \times 1 \times (44.89 + 0.5^2) = -45.14$$

Pada proses di atas dijelaskan hasil dari perhitungan matrix hessian pada baris ke-1 dan kolom ke-2 adalah 55.01. lalu hasil perhitungan matrix hessian pada baris ke-1 dan kolom ke-2 adalah -45.14. pada fungsi kernel di ambil dari hasil perhitungan kernel pada tabel 3.5. Selanjutnya, perhitungan matrix di lakukan secara terus menerus sampai pada baris ke-10 dan juga kolom ke-10 (D_{1010}). Sehingga didapatkan hasil perhitungan matrix hessian secara keseluruhan pada tabel 3.7.

Tabel 3.5 Hasil Perhitungan *matrix hessian*

	1	2	3	4	5	6	7	8	9	10
1	55.01	-45.14	36.25	49.25	26.26	50.66	41.21	38.69	56.5	52.09
2	-45.14	36.732	-28.98	-41.21	-21.044	-43.284	-34.824	-32.512	-48.97	-45.14
3	36.25	-28.98	22.246	33.313	15.774	35.534	28.023	25.955	40.827	37.46
4	49.25	-41.21	33.313	47.173	26.26	50.66	41.853	39.313	57.253	53.54
5	26.26	-21.044	15.774	26.26	12.084	29.844	23.482	21.594	35.296	32.74
6	50.66	-43.284	35.534	50.66	29.844	55.604	46.762	44.074	62.976	59.54
7	41.21	-34.824	28.023	41.853	23.482	46.762	38.814	36.37	53.686	50.66
8	38.69	-32.512	25.955	39.313	21.594	44.074	36.37	34.006	50.802	47.86
9	56.5	-48.97	40.827	57.253	35.296	62.976	53.686	50.802	70.978	67.49
10	52.09	-45.14	37.46	53.54	32.74	59.54	50.66	47.86	67.49	64.25

Setelah perhitungan *matrix hessian*, langkah selanjutnya adalah mencari nilai error ke-i (E_i). Langkah ini mencari nilai (E_1) sampai (E_{10}). Maksud dari E_1 ialah nilai error pada iterasi pertama atau baris pertama. Rumus yang di gunakan adalah rumus 3.5.

$$E_i = \sum_{j=1}^n a_i D_{ij} \quad (3.4)$$

Keterangan:

E_i = eror rate

D_{ij} = *matrix hessian* baris ke-i dan kolom ke-i

a_i = *alpha* ke-i

Dari rumus 3.5 dilakukan perhitungan untuk mencari nilai E_i . Untuk nilai a_i di ambil dari nilai *alpha* yang telah di definisikan pada tabel 3.6. Proses perhitungan mencari nilai error dilakukan seperti pada dibawah ini:

$$\begin{aligned} E_i &= (55.01 \times 0.001) + (-45.14 \times 0.001) + (36.25 \times 0.001) + (49.25 \times 0.001) \\ &\quad + (26.26 \times 0.001) + (50.66 \times 0.001) + (41.21 \times 0.001) \\ &\quad + (38.69 \times 0.001) + (56.5 \times 0.001) + (52.09 \times 0.001) = \mathbf{0.3608} \end{aligned}$$

$$\begin{aligned}
E_2 &= (-45.14 \times 0.001) + (36.732 \times 0.001) + (-28.98 \times 0.001) + (-41.21 \times 0.001) \\
&\quad + (-21.0436 \times 0.001) + (-43.284 \times 0.001) + (-34.824 \times 0.001) \\
&\quad + (-32.512 \times 0.001) + (-38.97 \times 0.001) + (-45.14 \times 0.001) \\
&= \mathbf{-0.3044}
\end{aligned}$$

$$\begin{aligned}
E_3 &= (36.25 \times 0.001) + (-28.29 \times 0.001) + (22.246 \times 0.001) + (33.313 \times 0.001) \\
&\quad + (15.774 \times 0.001) + (35.534 \times 0.001) + (28.023 \times 0.001) \\
&\quad + (25.955 \times 0.001) + (40.827 \times 0.001) + (37.46 \times 0.001) = \mathbf{0.2464}
\end{aligned}$$

$$\begin{aligned}
E_4 &= (49.25 \times 0.001) + (-41.21 \times 0.001) + (33.313 \times 0.001) + (47.173 \times 0.001) \\
&\quad + (26.26 \times 0.001) + (50.66 \times 0.001) + (41.853 \times 0.001) \\
&\quad + (39.313 \times 0.001) + (57.253 \times 0.001) + (53.54 \times 0.001) = \mathbf{0.3574}
\end{aligned}$$

$$\begin{aligned}
E_5 &= (26.26 \times 0.001) + (-21.044 \times 0.001) + (15.774 \times 0.001) + (26.26 \times 0.001) \\
&\quad + (12.084 \times 0.001) + (29.844 \times 0.001) + (23.482 \times 0.001) \\
&\quad + (21.594 \times 0.001) + (35.296 \times 0.001) + (32.74 \times 0.001) = \mathbf{0.2023}
\end{aligned}$$

$$\begin{aligned}
E_6 &= (50.66 \times 0.001) + (-43.284 \times 0.001) + (35.534 \times 0.001) + (50.66 \times 0.001) \\
&\quad + (29.844 \times 0.001) + (55.604 \times 0.001) + (46.762 \times 0.001) \\
&\quad + (44.074 \times 0.001) + (62.976 \times 0.001) + (59.54 \times 0.001) = \mathbf{0.3924}
\end{aligned}$$

$$\begin{aligned}
E_7 &= (41.21 \times 0.001) + (-34.824 \times 0.001) + (28.023 \times 0.001) + (41.843 \times 0.001) \\
&\quad + (23.482 \times 0.001) + (46.762 \times 0.001) + (38.814 \times 0.001) \\
&\quad + (36.37 \times 0.001) + (53.686 \times 0.001) + (50.66 \times 0.001) = \mathbf{0.326}
\end{aligned}$$

$$\begin{aligned}
E_8 &= (38.69 \times 0.001) + (-32.512 \times 0.001) + (25.955 \times 0.001) + (39.313 \times 0.001) \\
&\quad + (21.594 \times 0.001) + (44.074 \times 0.001) + (36.37 \times 0.001) \\
&\quad + (34.006 \times 0.001) + (50.802 \times 0.001) + (47.86 \times 0.001) = \mathbf{0.3062}
\end{aligned}$$

$$\begin{aligned}
E_9 &= (59.5 \times 0.001) + (-48.97 \times 0.001) + (40.827 \times 0.001) + (57.253 \times 0.001) \\
&\quad + (35.296 \times 0.001) + (62.976 \times 0.001) + (53.686 \times 0.001) \\
&\quad + (50.802 \times 0.001) + (70.978 \times 0.001) + (67.49 \times 0.001) = \mathbf{0.4468}
\end{aligned}$$

$$\begin{aligned}
E_{10} &= (52.09 \times 0.001) + (-45.14 \times 0.001) + (37.46 \times 0.001) + (53.54 \times 0.001) \\
&\quad + (32.74 \times 0.001) + (59.54 \times 0.001) + (50.66 \times 0.001) \\
&\quad + (47.86 \times 0.001) + (67.49 \times 0.001) + (64.25 \times 0.001) = \mathbf{0.420}
\end{aligned}$$

Setelah nilai E_i di dapatkan tahap selanjutnya mencari *delta alpha* (δa) dilakukan perulangan di setiap baris pada nilai E_i . Proses perhitungan δa dilakukan menggunakan rumus 3.6.

$$\delta a_i = \min \{ \max | \gamma(1 - E_i), a_i |, C - a_i \} \quad (3.5)$$

Keterangan:

a_i = alfa ke-i

γ = gamma

E_i = eror rate

C = nilai kompleksitas

Perhitungan δa :

$$\delta a_1 = \min \{ \max | 0.5(1 - 0.3608), 0.001 |, 0.001 - 0.001 \} = 0.3196$$

$$\delta a_2 = \min \{ \max | 0.5(1 - (-0.3044)), 0.001 |, 0.001 - 0.001 \} = 0.6522$$

$$\delta a_3 = \min \{ \max | 0.5(1 - 0.2464), 0.001 |, 0.001 - 0.001 \} = 0.3768$$

$$\delta a_4 = \min \{ \max | 0.5(1 - 0.3574), 0.001 |, 0.001 - 0.001 \} = 0.3213$$

$$\delta a_5 = \min \{ \max | 0.5(1 - 0.2023), 0.001 |, 0.001 - 0.001 \} = 0.3989$$

$$\delta a_6 = \min \{ \max | 0.5(1 - 0.3924), 0.001 |, 0.001 - 0.001 \} = 0.3038$$

$$\delta a_7 = \min \{ \max | 0.5(1 - 0.326), 0.001 |, 0.001 - 0.001 \} = 0.337$$

$$\delta a_8 = \min \{ \max | 0.5(1 - 0.3062), 0.001 |, 0.001 - 0.001 \} = 0.3469$$

$$\delta a_9 = \min \{ \max | 0.5(1 - 0.4468), 0.001 |, 0.001 - 0.001 \} = 0.2766$$

$$\delta a_{10} = \min \{ \max | 0.5(1 - 0.4205), 0.001 |, 0.001 - 0.001 \} = 0.2898$$

Setelah nilai *delta alpha* didapatkan maka langkah selanjutnya adalah menghitung nilai *alpha* terbaru. Adapun rumus yang di pakai adalah rumus 3.7.

$$a_i = a_i + \delta a_i \quad (3.6)$$

Pada rumus 3.7, nilai a_i di dapatkan dari nilai $alpha$ yang telah di definisikan pada tabel 3.6. Selanjutnya nilai δa_i di dapatkan dari nilai $delta$ $alpha$ yang baru di dapatkan pada tabel 3.9. selanjutnya untuk proses perhitungan $alpha$ baru ialah:

$$a_1 = 0.001 + 0.3196 = 0.3206$$

$$a_2 = 0.001 + 0.6522 = 0.6532$$

$$a_3 = 0.001 + 0.3768 = 0.3778$$

$$a_4 = 0.001 + 0.3213 = 0.3223$$

$$a_5 = 0.001 + 0.3989 = 0.3999$$

$$a_6 = 0.001 + 0.3038 = 0.3048$$

$$a_7 = 0.001 + 0.337 = 0.338$$

$$a_8 = 0.001 + 0.3469 = 0.3479$$

$$a_9 = 0.001 + 0.2766 = 0.2776$$

$$a_{10} = 0.001 + 0.2898 = 0.2908$$

Pada perhitungan di atas merupakan proses perhitungan mencari nilai $Alpha$ terbaru, Langkah selanjutnya adalah mencari nilai $alpha$ maksimum. Pada tahap ini di lakukan perhitungan ulang pada perhitungan E_i , δa , dan a_i . Perhitungan tersebut menggunakan nilai yang telah didapatkan pada perhitungan di atas. Sehingga hasil perhitungan nilai $alpha$ terbaru seperti pada tabel 3.9.

Tabel 3.6 Hasil perhitungan $Alpha$ terbaru

NO	$deltaAlpa$	Alpa	kelas(yi)	kategori kelas
1	0.32061	0.6412	3	Negatif
2	0.67939	1.3326	1	Positif
3	0.32061	0.6984	2	Negatif
4	0.32061	0.6429	5	Negatif
5	0.32061	0.7205	3	Negatif
6	0.32061	0.6254	6	Negatif

NO	<i>deltaAlpa</i>	Alpa	kelas(y_i)	kategori kelas
7	0.32061	0.6586	4	Negatif
8	0.32061	0.6685	3	Negatif
9	0.32061	0.5982	4	Negatif
10	0.32061	0.6114	5	Negatif

Setelah nilai *Alpha* terbaru telah di dapatkan langkan selanjutnya adalah mencari nilai bias. Untuk mencari nilai bias menggunakan rumus di bawah ini

$$b = -\frac{1}{2} (w \cdot x^+ + w \cdot x^-) \quad (3.7)$$

Keterangan:

$w \cdot x^+$ = nilai bobot untuk kelas positif

$w \cdot x^-$ = nilai bobot untuk kelas negative

Seperti keterangan pada rumus di atas, maka untuk mencari nilai bias membutuhkan nilai bobot. Oleh karena itu, di lakukan perhitungan untuk mencari nilai bobot terlebih dahulu. Setelah nilai bobot negatif dan positif diketahui maka di lakukan perhitungan untuk mencari bilai bias. Adapun rumus yang di gunakan untuk mencari nilai bobot negatif (w^-) dan positif (w^+) adalah

$$w^+ = \sum_{i=1}^n a_i y_i K(x^i x^+) \quad (3.8)$$

Keterangan:

a_i = nilai bobot data ke-i

y_i = kelas data ke-i

$K(x^i x^+)$ = perhitungan kernel dari nilai *alpha* tertinggi dari kelas positif ke-i

$$w^- = \sum_{i=1}^n a_i y_i K(x^i x^-) \quad (3.9)$$

Keterangan:

a_i = nilai bobot data ke-i

y_i = kelas data ke-i

$K(x^i x^-)$ = perhitungan kernel dari nilai *alpha* tertinggi dari kelas negatif ke-i

Pada rumus 3.9 dan 3.10 merupakan rumus untuk mencari bobot dari kelas positif dan negatif. Selanjutnya di perlukan perhitungan kernel dari nilai maksimal kelas positif dan negatif. Seperti pada tabel 3.9 nilai maksimal dari kelas positif adalah data indeks ke-2 sedangkan kelas negatif adalah data indeks ke-5. Untuk nilai *indeks* yang akan di pakai pada perhitungan kernel adalah nilai yang ada pada tabel normalisasi pada tabel 3.5.

Perhitungan kernel pada indeks ke-2 :

$$\begin{aligned} K(2,1) &= (0.2 \times 0) + (0.9 \times 1) + (0.4 \times 1) + (0.1 \times 0.1) + (0.3 \times 0.1) + (0.5 \times 1) + (0.3 \times 0) \\ &= 44.89 \end{aligned}$$

$$\begin{aligned} K(2,2) &= (0.2 \times 0.2) + (0.9 \times 0.9) + (0.4 \times 0.4) + (0.1 \times 0.1) + (0.3 \times 0.3) + (0.5 \times 0.5) \\ &\quad + (0.3 \times 0.3) = 36.482 \end{aligned}$$

$$\begin{aligned} K(2,3) &= (0.2 \times 0.3) + (0.9 \times 0.8) + (0.4 \times 0.6) + (0.1 \times 0.1) + (0.3 \times 0) + (0.5 \times 0) \\ &\quad + (0.3 \times 0.3) = 29.73 \end{aligned}$$

$$\begin{aligned} K(2,4) &= (0.2 \times 0.5) + (0.9 \times 0.6) + (0.4 \times 0.8) + (0.1 \times 0.3) + (0.3 \times 0.8) + (0.5 \times 0) \\ &\quad + (0.3 \times 0.3) = 40.96 \end{aligned}$$

$$K(2,5) = (0.2 \times 0.8) + (0.9 \times 0.4) + (0.4 \times 0.1) + (0.1 \times 0) + (0.3 \times 0.1) + (0.5 \times 0) \\ + (0.3 \times 0.3) = 20.794$$

$$K(2,6) = (0.2 \times 0.8) + (0.9 \times 0.4) + (0.4 \times 0.4) + (0.1 \times 0.3) + (0.3 \times 0.6) + (0.5 \times 0.5) \\ + (0.3 \times 0.7) = 43.034$$

$$K(2,7) = (0.2 \times 0.9) + (0.9 \times 0.2) + (0.4 \times 0.7) + (0.1 \times 0.6) + (0.3 \times 0) + (0.5 \times 0) \\ + (0.3 \times 0.7) = 34.574$$

$$K(2,8) = (0.2 \times 0.9) + (0.9 \times 0.1) + (0.4 \times 0.3) + (0.1 \times 0.5) + (0.3 \times 0.4) + (0.5 \times 0) \\ + (0.3 \times 0.7) = 32.262$$

$$K(2,9) = (0.2 \times 0.9) + (0.9 \times 0.1) + (0.4 \times 0.9) + (0.1 \times 0.6) + (0.3 \times 1) + (0.5 \times 0) \\ + (0.3 \times 0.7) = 48.72$$

$$K(2,10) = (0.2 \times 1) + (0.9 \times 0) + (0.4 \times 0) + (0.1 \times 1) + (0.3 \times 0) + (0.5 \times 1) + (0.3 \times 1) \\ = 44.89$$

Perhitungan kernel pada indeks ke-3:

$$K(3,1) = (0.3 \times 0) + (0.8 \times 1) + (0.6 \times 1) + (0.1 \times 0.1) + (0 \times 0.1) + (0 \times 1) + (0.3 \times 0) \\ = 26.01$$

$$K(3,2) = (0.3 \times 0.2) + (0.8 \times 0.9) + (0.6 \times 0.4) + (0.1 \times 0.1) + (0 \times 0.3) + (0 \times 0.5) \\ + (0.3 \times 0.3) = 20.794$$

$$K(3,3) = (0.3 \times 0.3) + (0.8 \times 0.8) + (0.6 \times 0.6) + (0.1 \times 0.1) + (0 \times 0) + (0 \times 0) \\ + (0.3 \times 0.3) = 15.524$$

$$K(3,4) = (0.3 \times 0.5) + (0.8 \times 0.6) + (0.6 \times 0.8) + (0.1 \times 0.3) + (0 \times 0.8) + (0 \times 0) \\ + (0.3 \times 0.3) = 26.01$$

$$K(3,5) = (0.3 \times 0.8) + (0.8 \times 0.4) + (0.6 \times 0.1) + (0.1 \times 0) + (0 \times 0.1) + (0 \times 0) \\ + (0.3 \times 0.3) = 11.834$$

$$K(3,6) = (0.3 \times 0.8) + (0.8 \times 0.4) + (0.6 \times 0.4) + (0.1 \times 0.3) + (0 \times 0.6) + (0 \times 0.5) \\ + (0.3 \times 0.7) = 29.594$$

$$K(3,7) = (0.3 \times 0.9) + (0.8 \times 0.2) + (0.6 \times 0.7) + (0.1 \times 0.6) + (0 \times 0) + (0 \times 0) \\ + (0.3 \times 0.7) = 23.232$$

$$K(3,8) = (0.3 \times 0.9) + (0.8 \times 0.1) + (0.6 \times 0.3) + (0.1 \times 0.5) + (0 \times 0.4) + (0 \times 0) \\ + (0.3 \times 0.7) = 21.344$$

$$K(3,9) = (0.3 \times 0.9) + (0.8 \times 0.1) + (0.6 \times 0.9) + (0.1 \times 0.6) + (0 \times 1) + (0 \times 0) \\ + (0.3 \times 0.7) = 35.046$$

$$K(3,10) = (0.3 \times 1) + (0.8 \times 0) + (0.6 \times 0) + (0.1 \times 1) + (0 \times 0) + (0 \times 1) + (0.3 \times 1) \\ = 32.49$$

Setelah perhitungan kernel di temukan maka Langkah selanjutnya adalah menghitung bobot kelas positif dan negatif. Untuk menghitung bobot kelas positif menggunakan rumus 3.9 dan menghitung kelas negatif menggunakan rumus 3.10.

Perhitungan bobot kelas positif:

$$w^+ = (0.6412 * -1 * 44.89) + (1.3326 * 1 * 36.482) + (0.6484 * -1 * 28.73) \\ + (0.6429 * -1 * 40.96) + (0.7205 * -1 * 20.794) \\ + (0.6254 * -1 * 43.034) + (0.6586 * -1 * 34.574) \\ + (0.6685 * -1 * 32.262) + (0.5982 * -1 * 48.72) \\ + (0.6114 * -1 * 44.89) = -169.39$$

Perhitungan bobot kelas negatif:

$$w^- = (0.6412 * -1 * 26.01) + (1.3326 * 1 * 20.794) + (0.6484 * -1 * 15.524) \\ + (0.6429 * -1 * 26.01) + (0.7205 * -1 * 11.834) \\ + (0.6254 * -1 * 29.594) + (0.6586 * -1 * 23.232) \\ + (0.6685 * -1 * 21.344) + (0.5982 * -1 * 35.046) \\ + (0.6114 * -1 * 32.49) = -113.97$$

Setelah nilai bobot ditemukan maka dapat dilanjutkan dengan menghitung nilai bias. Perhitungan nilai bias menggunakan rumus 3.8.

$$b = -\frac{1}{2} (-169.39 + (-113.97)) = 141.68$$

3.6 Skenario Pengujian *System Usability Scale* (SUS)

Pengujian dilakukan dengan melibatkan sekelompok peserta yang diminta untuk bermain *game* edukasi tersebut dan memberikan penilaian terhadap usabilitynya menggunakan skala SUS. Peserta diberikan instruksi untuk memainkan *game* dan menyelesaikan serangkaian tugas atau tantangan yang disediakan. Selama proses pengujian, peserta diamati dalam interaksi mereka dengan *game*, dan perhatian khusus diberikan pada kesulitan yang mereka alami dan umpan balik verbal atau reaksi non-verbal yang ditunjukkan. Setelah selesai memainkan *game*, peserta diminta untuk mengisi skala SUS yang terdiri dari sepuluh pernyataan tentang usability *game* edukasi tersebut. Selain itu, mereka juga diberikan kesempatan untuk memberikan tanggapan kualitatif mengenai pengalaman mereka dalam bermain *game*. Adapun kuesioner SUS yang digunakan adalah sebagai berikut.

Tabel 3.7 Daftar kuesioner SUS

Kode	Pertanyaan
R1	Saya merasa mudah untuk belajar dengan bermain <i>game</i> ini.
R2	Saya merasa sistem kontrol <i>game</i> ini mudah dipahami dan digunakan.
R3	Saya merasa pengaturan level kesulitan memberikan tantangan yang sesuai
R4	Saya merasa <i>game</i> ini memberikan umpan balik yang memadai tentang tindakan atau keputusan saya.
R5	Saya merasa antarmuka pengguna <i>game</i> ini mudah dinavigasi dan intuitif.
R6	Saya merasa <i>game</i> ini menawarkan variasi dan tantangan yang cukup.
R7	Saya merasa <i>game</i> ini memotivasi saya untuk terus bermain dan belajar lebih lanjut.
R8	Saya merasa fitur dan fungsionalitas <i>game</i> ini sesuai dengan tujuan edukasinya.
R9	Saya merasa pengaturan level kesulitan <i>game</i> memberikan tantangan yang menarik untuk dikuasai
R10	Saya merasa pengaturan level kesulitan <i>game</i> memungkinkan adanya peningkatan keterampilan seiring berjalannya waktu

Pernyataan-pernyataan di atas digunakan dalam mengukur usability pengaturan level kesulitan dinamis pada *game* bencana gunung meletus. Responden akan diminta untuk menilai sejauh mana mereka setuju atau tidak setuju dengan masing-masing pernyataan menggunakan skala Likert. Skor dari kuesioner akan digunakan untuk menghitung skor SUS secara keseluruhan, yang akan memberikan gambaran tentang tingkat usability pengaturan level kesulitan dinamis pada *game* bencana gunung meletus.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab hasil dan pembahasan, peneliti menjelaskan hasil dari pengaturan level kesulitan dinamis menggunakan metode SVM. Selanjutnya peneliti menggunakan *System Usability Scale* (SUS). Sebagai metode yang digunakan untuk pengujian dari *game* yang telah dibuat.

4.1 Implementasi *User Interface*

4.1.1 Tampilan Awal *Game*



Gambar 4.1 Tampilan Awal *Game*

Pada gambar 4.1 merupakan tampilan ketika *game* di mulai, pada awal *game* terdapat pilihan bermain menggunakan metode SVM dan bermain tanpa menggunakan metode SVM.

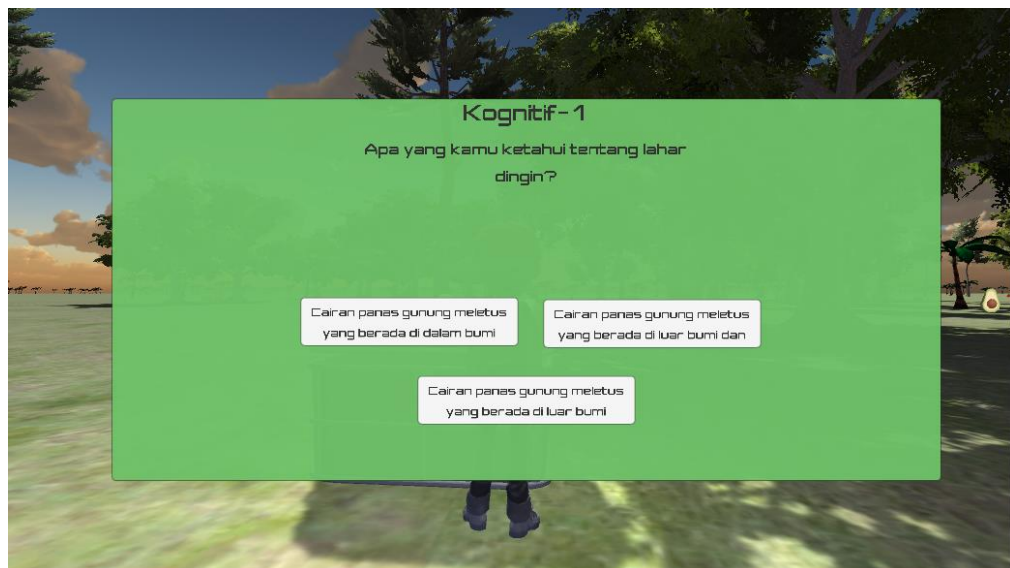
4.1.2 Tampilan *Gameplay*



Gambar 4.2 Tampilan *Gameplay*

Pada gambar 4.2 merupakan tampilan Ketika *game* berjalan.

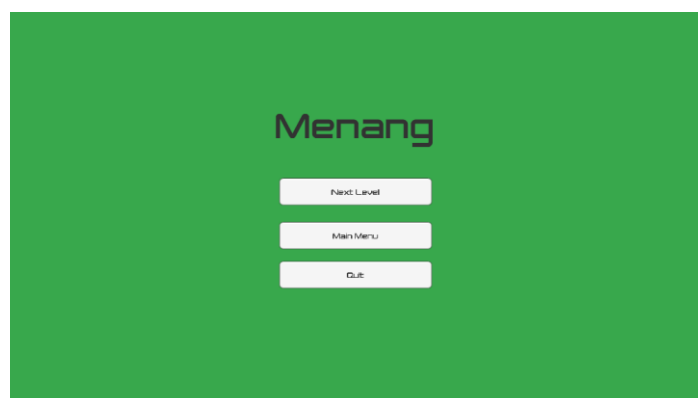
4.1.3 Tampilan Kuis



Gambar 4.3 Tampilan Kuis

Ketika pemain menemukan peti, peti tersebut akan menampilkan pertanyaan tentang bencana alam, Ketika pemain berhasil menjawab akan mendapatkan poin tambahan. Pada gambar 4.3 merupakan tampilan Ketika menampilkan kuis.

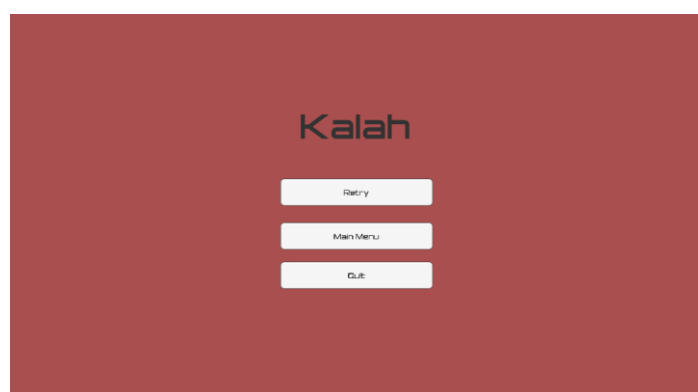
4.1.4 Tampilan Menang



Gambar 4.4 Tampilan Menang

Ketika player telah mencapai skor yang telah di tentukan akan menampilkan panel menang seperti pada gambar 4.4.

4.1.5 Tampilan Kalah



Gambar 4.5 Tampilan Kalah

Ketika Player tidak mencapai skor yang di tentukan dalam waktu yang di tentukan atau Ketika darah player habis, maka akan menampilkan panel kalah seperti pada gambar 4.5.

4.2 Implementasi *Support Vector Machine*

Seluruh hasil pengujian tanpa menggunakan *Support Vector Machine* yang berjumlah 100 data dijadikan *dataset* yang akan digunakan untuk mengolah arsitektur *Support Vector Machine*.

Tabel 4.1 *Dataset*

skor	waktu	Darah Player	Nilai barang	Jumlah Barang	Tipe Musuh	Jumlah Musuh	Level
3	200	100	9	5	4	8	1
15	192	96	7	16	1	1	1
78	150	75	7	6	2	1	3
150	102	51	5	11	3	1	5
201	68	34	9	11	2	1	7
210	62	31	7	15	6	1	7
222	54	27	7	15	5	1	8
228	50	25	9	11	1	1	8
231	48	24	5	5	5	1	8
243	40	20	9	8	2	1	9
255	32	16	10	5	5	1	9
6	198	99	9	6	3	2	1
66	158	79	9	16	1	2	3
153	100	50	7	16	4	2	6
270	22	11	10	9	2	2	9
288	10	5	7	12	3	2	10
300	2	1	5	17	2	2	10
30	182	91	7	19	1	3	1
36	178	89	9	17	4	3	2
39	176	88	9	20	6	3	2
51	168	84	5	8	1	3	2
63	160	80	9	6	4	3	3
87	144	72	5	16	3	3	3

skor	waktu	Darah <i>Player</i>	Nilai barang	Jumlah Barang	Tipe Musuh	Jumlah Musuh	Level
123	120	60	9	6	6	3	5
129	116	58	9	20	6	3	5
141	108	54	9	14	6	3	5
147	104	52	9	13	4	3	5
174	86	43	5	10	2	3	6
183	80	40	7	9	5	3	7
246	38	19	7	11	6	3	9

Selanjutnya dilakukan normalisasi data pada dataset yang telah ada.

Tabel 4.2 Hasil Normalisasi

Skor	Waktu	Darah <i>Player</i>	Nilai barang	Jumlah Barang	Tipe Musuh	Jumlah Musuh
0	1	1	0.83	0	0.6	0.37
0.04	0.96	0.96	0.5	0.73	0	0
0.25	0.75	0.75	0.5	0.07	0.2	0
0.49	0.51	0.51	0.17	0.4	0.4	0
0.67	0.33	0.33	0.83	0.4	0.2	0
0.7	0.3	0.3	0.5	0.67	1	0
0.74	0.26	0.26	0.5	0.67	0.8	0
0.76	0.24	0.24	0.83	0.4	0	0
0.77	0.23	0.23	0.17	0	0.8	0
0.81	0.19	0.19	0.83	0.2	0.2	0
0.85	0.15	0.15	1	0	0.8	0
0.01	0.99	0.99	0.83	0.07	0.4	0.05
0.21	0.79	0.79	0.83	0.73	0	0.05
0.51	0.49	0.49	0.5	0.73	0.6	0.05
0.9	0.1	0.1	1	0.27	0.2	0.05
0.96	0.04	0.04	0.5	0.47	0.4	0.05
1	0	0	0.17	0.8	0.2	0.05
0.09	0.91	0.91	0.5	0.93	0	0.11
0.11	0.89	0.89	0.83	0.8	0.6	0.11
0.12	0.88	0.88	0.83	1	1	0.11
0.16	0.84	0.84	0.17	0.2	0	0.11
0.2	0.8	0.8	0.83	0.07	0.6	0.11
0.28	0.72	0.72	0.17	0.73	0.4	0.11
0.4	0.6	0.6	0.83	0.07	1	0.11

Skor	Waktu	Darah <i>Player</i>	Nilai barang	Jumlah Barang	Tipe Musuh	Jumlah Musuh
0.42	0.58	0.58	0.83	1	1	0.11
0.46	0.54	0.54	0.83	0.6	1	0.11
0.48	0.52	0.52	0.83	0.53	0.6	0.11
0.58	0.42	0.42	0.17	0.33	0.2	0.11
0.61	0.39	0.39	0.5	0.27	0.8	0.11
0.82	0.18	0.18	0.5	0.4	1	0.11

Langkah selanjutnya yaitu perhitungan kernel, pada perhitungan kernel dilakukan secara berurutan dimulai dari baris 1 akan di kalikan oleh baris 1 sampai dengan baris akhir yaitu 30. Pada perkalian baris 1 dengan baris satu akan di masukkan dalam sebuah matrik dengan kolom 1 dan baris 1 k(1,1). Lalu di lanjut sampai dengan perkalian baris 30 sampai dengan baris 30 akan di masukkan dalam sebuah matrik dengan baris dan kolom 30 k(30,30). sehingga dari nilai normalisasi pada tabel 4.2 di dapatkan nilai kernel seperti pada tabel 4.3.

$$K(1,1) = (0 \times 0) + (1 \times 1) + (1 \times 1) + (0.83 \times 0.83) + (0 \times 0) + (0.6 \times 0.6) \\ + (0.37 \times 0.37) = 96.04$$

$$K(1,2) = (0 \times 0.4) + (1 \times 0.96) + (1 \times 0.96) + (0.83 \times 0.5) + (0 \times 0.73) \\ + (0.6 \times 0) + (0.37 \times 0) = 71.4$$

Tabel 4.3 Hasil Perhitungan Kernel

x,y	1	2	3	4	5	6	7	30
1	96.04	71.4	57.3	52.13	55.65	65.12	60.68	51.41
2	71.4	50.43	38.81	34.69	37.66	45.54	41.86	42.94
3	57.3	38.81	29.19	26.14	29.13	36.18	33.01	34.16
4	52.13	34.69	26.14	23.82	27.13	34.02	31.05	33.2
5	55.65	37.66	29.13	27.13	31.01	38.43	35.35	36.11
6	65.12	45.54	36.18	34.02	38.43	46.65	43.27	45.08
7	60.68	41.86	33.01	31.05	35.35	43.27	40.04	41.82
..
..
30	51.41	42.94	34.16	33.2	36.11	45.08	41.82	41.12

Pada tahap ini merupakan proses inialisasi kelas, pada tahap ini merupakan proses inialisasi kelas. Dimulai dari pencarian kelas 1 maka pada data yang berada di dataset yang level 1 akan di inialisasi menjadi kelas 1 sedangkan selain level 1 akan di inialisasi menjadi kelas -1. Proses pencarian kelas ini akan di lakukan hingga 9 kali.

Selanjutnya menghitung matrix hessian level 1, dari perhitungan kernel pada tabel 4.3 maka di dapatkan matrix hessian sebagai berikut.

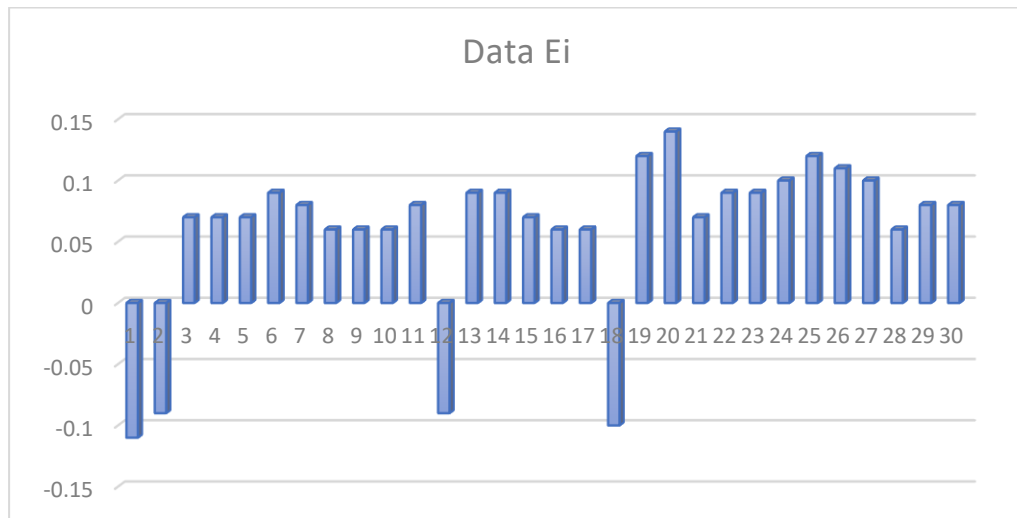
$$D_{11} = 1 \times 1 \times (54.71 + 0.5^2) = 18.49$$

$$D_{12} = 1 \times 1 \times (44.89 + 0.5^2) = 15.08$$

Tabel 4.4 *Matrix Hessian* level 1

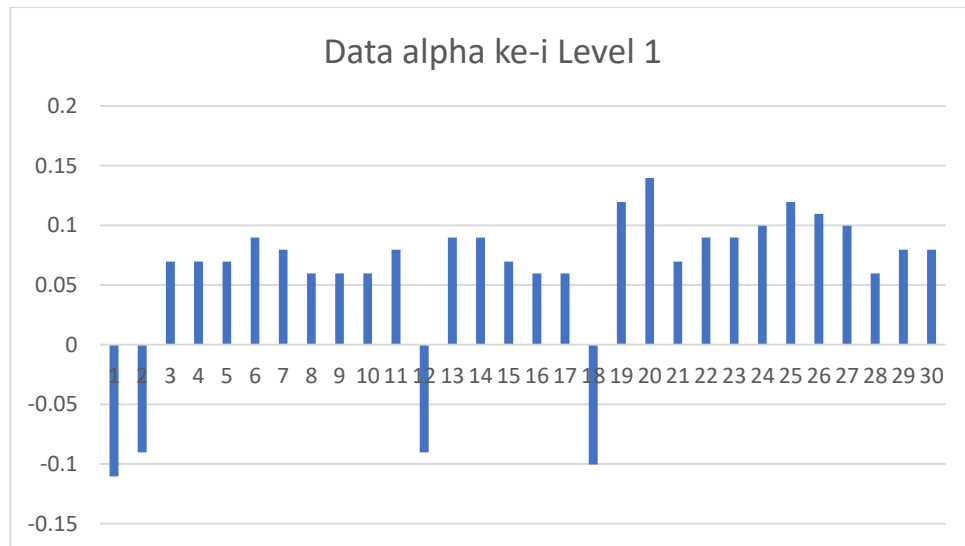
x,y	1	2	3	4	5	6	7	30
1	18.49	15.8	12.5	11.53	11.87	14.33	13.29	12.85
2	15.8	13.33	10.34	9.49	9.82	12.07	11.12	10.73
3	12.5	10.34	7.85	7.24	7.64	9.66	8.84	8.54
4	11.53	9.49	7.24	6.81	7.31	9.31	8.53	8.3
5	11.87	9.82	7.64	7.31	7.92	10.01	9.23	9.03
6	14.33	12.07	9.66	9.31	10.01	12.36	11.49	11.27
7	13.29	11.12	8.84	8.53	9.23	11.49	10.65	10.45
...
...
30	12.85	10.73	8.54	8.3	9.03	11.27	10.45	10.2

Setelah melakukan perhitungan matrix hessian Langkah selanjutnya adalah melakukan perhitungan nilai E^i dari setiap indeks.



Gambar 4.6 Hasil perhitungan nilai E_i

Untuk mencari nilai α terbaru di butuhkan learning rate, learning rate di dapatkan dari $0.01 / \max$ (matrix hessian) nilai tersebut adalah nilai dari $\Delta\alpha$. Selanjutnya masing-masing index nilai $\Delta\alpha$ di tambahkan nilai α yang di inisialisasi di awal akan menghasilkan nilai α terbaru.



Gambar 4.7 Nilai α ke-i level 1

Selanjutnya dilakukan menghitung nilai bias, dalam mencari nilai bias di butuhkan bobot dari kelas negatif (-1) dan positif (1). Dalam mencari nilai bobot di perlukan nilai kernel dari nilai maksimum dari *alpha* terbaru pada gambar 4.7 kelas negatif dan positif. Untuk kelas data positif nilai maksimum terdapat pada indeks dari dataset ke 2 dan kelas data negatif terdapat pada indeks dataset ke 20. Lalu langkah selanjutnya adalah pencarian bobot (w). bobot ini dibagi menjadi 2 yaitu bobot kelas data positif (w^+) dan juga bobot kelas data negatif (w^-). Dalam pencarian nilai bobot ini baik dari kelas positif atau kelas negatif, perlu dilakukan pencarian nilai bobot di setiap indeks data (w^i). Adapun contoh perhitungan pencarian nilai bobot indeks data positif (w^{i+}) dan juga bobot indeks data kelas negatif (w^{i-}) adalah sebagai berikut.

$$w^{i+} = \max(\text{alpha kelas positif}) \times \text{kelas data} \times \text{kernel data ke } i$$

$$w^{i-} = \max(\text{alpha kelas negatif}) \times \text{kelas data} \times \text{kernel data ke } i$$

$$w^{1+} = -0.09 \times 1 \times 63.20 = -5.87$$

$$w^{1-} = -0.14 \times 1 \times 90.25 = -14.08$$

$$w^{2+} = -0.09 \times 1 \times 53.32 = 5.87$$

$$w^{2-} = -0.09 \times 1 \times 78.40 = 10.63$$

Dari perhitungan di atas akan di lanjutkan hingga sampai ke indeks ke-30 (w^{30}). Selanjutnya setiap bobot dari indeks 1 sampai terakhir akan di jumlah kan sehingga akan menghasilkan bobot total. Bobot total juga di bagi menjadi 2 yaitu total bobot kelas positif (w^+) dan juga total bobot kelas negatif (w^-).

$$w^+ = w^1 + w^2 + \dots + \dots + w^{28} + w^{30}$$

$$w^- = w^1 + w^2 + \dots + \dots + w^{28} + w^{30}$$

$$w^+ = -5.87 + -4.15 + \dots + \dots + 3.60 + 3.51 = 77.428$$

$$w^- = 14.08 + 10.63 + \dots + \dots + -9.60 \pm 9.45 = -206.36$$

Setelah itu di cari nilai bias dari kelas 1 antara lain

$$b = -\frac{1}{2} (\text{total bobot positif} + \text{total bobot negatif})$$

$$b = -\frac{1}{2} (77.428 + (-206.36)) = 64.754$$

Selanjutnya Dilakukan perhitungan untuk mencari nilai *alpha* dan nilai bias pada kelas 2 hingga kelas 9. Pada tabel merupakan hasil dari nilai *alpha* dari kelas 1 sampai 9. Sedangkan pada tabel merupakan hasil dari nilai bias pada kelas 1 sampai 9.

Tabel 4.5 Nilai *alpha* kelas 1 sampai kelas 9

No	Alpha								
	level 1	level 2	level 3	level 4	level 5	level 6	level 7	level 8	level 9
1	-0.119	0.145	0.108	0.048	0.043	0.051	0.047	0.039	0.034
2	-0.082	0.097	0.08	0.034	0.03	0.036	0.034	0.027	0.023
3	0.064	0.065	-0.066	0.032	0.027	0.034	0.031	0.024	0.02
4	0.06	0.061	0.061	0.027	0.022	0.028	0.025	0.019	0.163
5	0.067	0.068	0.067	0.031	0.025	0.031	0.029	0.022	0.19
6	0.082	0.082	0.08	0.037	0.031	0.037	0.035	0.028	0.024
7	0.076	0.076	0.075	0.034	0.028	0.035	0.032	0.026	0.022
8	0.06	0.061	0.061	0.028	0.023	0.028	0.026	0.02	0.017
9	0.052	0.053	0.053	0.024	0.019	0.025	0.022	0.017	0.014
10	0.059	0.06	0.059	0.027	0.022	0.028	0.025	0.019	0.016

Tabel 4.6 Nilai bias kelas 1 sampai kelas 9

nilai bias								
level 1	level 2	level 3	level 4	level 5	level 6	level 7	level 8	level 9
65.132	196.350	161.482	71.574	105.062	76.174	93.508	71.402	83.27

4.3 Klasifikasi Level Menggunakan SVM

Sebagai demonstrasi, ada beberapa contoh pemain dengan kondisi sebagai berikut:

$Score = 180$, Waktu = 70, Darah Pemain = 100, Nilai Barang = 4, Jumlah Barang = 20, Tipe Musuh = 5, dan Jumlah Musuh = 15.

Selanjutnya dilakukan normalisasi pada nilai input diatas.

$$Score = \frac{180-3}{300-3} = 0.6$$

$$Waktu = \frac{70-0}{300-3} = 0.34$$

$$Darah Pemain = \frac{100-0}{100-0} = 1$$

$$Nilai Barang = \frac{4-4}{10-4} = 0$$

$$Jumlah Barang = \frac{20-5}{20-5} = 1$$

$$Tipe Musuh = \frac{5-1}{6-1} = 0.8$$

$$Jumlah Musuh = \frac{15-1}{20-1} = 0.74$$

Selanjutnya setelah data input tadi di normalisasi. Perlu melakukan nilai kernel dari input utama. Untuk melakukan pencarian kernel perlu melakukan perkalian antara hasil normalisasi input utama dan hasil normalisasi pada setiap indeks (i) dari dataset. Sehingga hasil perhitungan kernel dilambangkan $k(1, i)$.

$$\begin{aligned} K(1,1) &= (0.6 \times 0) + (0.34 \times 1) + (1 \times 1) + (0 \times 0.83) + (1 \times 0) + (0.8 \times 0.6) + (0.74 \times 0.37) \\ &= 33.52 \end{aligned}$$

$$\begin{aligned} K(1,2) &= (0.6 \times 0.04) + (0.34 \times 0.96) + (1 \times 0.96) + (0 \times 0.5) + (1 \times 0.73) + (0.8 \times 0) + (0.74 \times 0.0) \\ &= 26.49 \end{aligned}$$

$$K(1,3) = (0.6 \times 0.25) + (0.34 \times 0.75) + (1 \times 0.75) + (0 \times 0.5) + (1 \times 0.07) + (0.8 \times 0.2) + (0.74 \times 0) \\ = 18.49$$

$$K(1,4) = (0.6 \times 0.49) + (0.34 \times 0.51) + (1 \times 0.51) + (0 \times 0.17) + (1 \times 0.4) + (0.8 \times 0.4) + (0.74 \times 0) \\ = 16.47$$

$$K(1,5) = (0.6 \times 0.67) + (0.34 \times 0.33) + (1 \times 0.33) + (0 \times 0.83) + (1 \times 0.4) + (0.8 \times 0.2) + (0.74 \times 0) \\ = 17.53$$

Selanjutnya perhitungan di atas akan dilanjutkan perhitungannya hingga data indeks ke 30 pada dataset ($k(1,30)$) untuk mendapatkan nilai kernel dari $k(1,1)$ hingga nilai kernel $k(1,30)$. Selanjutnya perlu dilakukan pencarian dari nilai bobot (w) dari setiap indeks (i) dari nilai kernel. Untuk pencarian nilai bobot diperlukan nilai *alpha* kelas 1 beserta kelas data dari kelas 1. Nilai *alpha* sendiri di dapat dari hasil pencarian *alpha* seperti pada tabel 4.8.

$$w^i = \text{nilai } \alpha \text{ kelas 1 indeks ke } i * \text{kelas data} * \text{nilai kernel}$$

$$w^1 = -0.119 * 1 * 33.52 = -3.59$$

$$w^2 = -0.082 * 1 * 26.49 = -2.38$$

$$w^3 = 0.064 * -1 * 18.49 = -1.29$$

$$w^4 = 0.06 * -1 * 16.47 = -1.1$$

Selanjutnya perhitungan bobot akan di lanjut hingga mencapai bobot dari indeks terakhir yaitu w^{30} . Lalu setiap bobot dari bobot indeks pertama (w^1) sampai indeks bobot terakhir (w^{30}) akan dijumlahkan. Hasil penjumlahan tersebut merupakan total bobot (w). Berdasarkan perhitungan diatas hasil dari total bobot (w) adalah -67.97. Total bobot ini akan di tambahkan oleh nilai bias lalu hasil dari penjumlahan akan di *Sign* (). Jika hasil penjumlahan menghasilkan 1 maka hasil klasifikasi akan menghasilkan level 1, Ketika menghasilkan nilai -1 maka di

pelukan perhitungan nilai *alpha* dan bias pada kelas 2 dan seterusnya hingga menghasilkan 1 pada saat di sign.

$$F(x) = \text{sign}(\text{bobot} + \text{nilai bias kelas 1})$$

$$F(x) = \text{sign}(-63.97 + 65.132)$$

$$F(x) = 1$$

Karena hasil penjumlahan antara bobot dengan nilai bias kelas satu menghasilkan nilai positif (1), maka klasifikasi level pada input adalah kelas 1 atau level 1.

4.4 Pengujian *System Usability Scale* (SUS)

Tanggapan di peroleh dari 15 responden. Berdasarkan pekerjaannya, terdiri dari 10 responden yang bekerja di bidang teknologi pendidikan dan 5 yang masih menjadi siswa SMK. Adapun kuesioner SUS yang di bagikan ke responden adalah sebagai berikut.

Tabel 4.7 Kuesioner SUS

kode	Pertanyaan
R1	Saya merasa mudah untuk belajar dengan bermain <i>game</i> ini.
R2	Saya merasa sistem kontrol <i>game</i> ini mudah dipahami dan digunakan.
R3	Saya merasa pengaturan level kesulitan memberikan tantangan yang sesuai
R4	Saya merasa <i>game</i> ini memberikan umpan balik yang memadai tentang tindakan atau keputusan saya.
R5	Saya merasa antarmuka pengguna <i>game</i> ini mudah dinavigasi dan intuitif.
R6	Saya merasa <i>game</i> ini menawarkan variasi dan tantangan yang cukup.
R7	Saya merasa <i>game</i> ini memotivasi saya untuk terus bermain dan belajar lebih lanjut.
R8	Saya merasa fitur dan fungsionalitas <i>game</i> ini sesuai dengan tujuan edukasinya.
R9	Saya merasa pengaturan level kesulitan <i>game</i> memberikan tantangan yang menarik untuk dikuasai
R10	Saya merasa pengaturan level kesulitan <i>game</i> memungkinkan adanya peningkatan keterampilan seiring berjalannya waktu

Hasil kuesioner kemudian dihitung dengan rumus yang telah di tentukan untuk mendapatkan skor SUS.

Tabel 4.8 Hasil Perhitungan Skor SUS

Responden	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Skor SUS
1	3	4	3	3	5	4	3	4	5	5	72.5
2	4	3	5	4	4	4	3	5	5	4	77.5
3	4	3	4	3	5	3	4	2	4	3	62.5
4	3	4	4	3	3	2	3	3	3	3	52.5
5	3	3	3	4	3	5	3	3	3	2	55
6	3	3	4	5	2	3	3	4	3	3	57.5
7	4	4	5	3	3	3	2	3	4	3	60
8	5	4	4	4	1	1	2	4	2	3	50
9	4	2	2	3	3	4	3	3	3	4	52.5
10	2	4	2	2	3	2	3	4	2	5	47.5
11	5	3	3	5	2	2	4	3	2	4	57.5
12	5	4	2	3	3	3	4	3	4	5	65
13	4	3	3	4	3	3	5	4	3	5	67.5
14	3	4	3	2	2	5	4	5	3	4	62.5
15	2	3	3	4	5	5	4	4	3	4	67.5

Keterangan nilai:

1 = Sangat Tidak setuju

2 = Tidak Setuju

3 = Kurang Setuju

4 = Setuju

5 = Sangat Setuju

Berdasarkan hasil perhitungan skor SUS pada tabel 4.8. maka di dapatkan rata-rata skor SUS sebesar 62.5 dari 100 poin.

4.5 Integrasi Sains dan Islam

1. *Mu'amalah Ma'a Allah*

Game yang telah dirancang memiliki tema yang berkaitan dengan bencana alam. Bencana alam merupakan sebuah peringatan yang diberikan kepada manusia agar mereka tetap mengingat dan mengakui kebesaran Allah berdasarkan seperti yang di firmankan Allah pada surah *Al-Qari'ah* ayat 5.

تَكُونُ الْجِبَالُ كَالْعِهْنِ الْمَنْفُوشِ ۝

“*dan gunung-gunung seperti bulu yang dihambur-hamburkan.*”

Berdasarkan tafsir Ibnu Katsir, dijelaskan bahwa Gunung-gunung di hari itu seakan-akan seperti bulu domba yang diawut-awut hingga menjadi beterbangan (Katsir, n.d.). Dari ayat di atas merupakan bukti dari kebesaran Allah swt dan Allah juga bisa menciptakan bencana apapun kepada umatnya yang tidak beriman. Sehingga bencana alam dijadikan tema dasar dengan harapan agar pemain dapat senantiasa mengingat kebesaran Allah.

2. *Mu'amalah Ma'a An-nas*

Dalam Alqur'an., Allah memberi perintah kepada setiap manusia agar saling menghargai dan saling memahami. Allah berfirman pada surah *Al-Ankabut* ayat 43.

وَتِلْكَ الْأَمْثَالُ نَضْرِبُهَا لِلنَّاسِ وَمَا يَعْقِلُهَا إِلَّا الْعَالِمُونَ

“*Dan perumpamaan-perumpamaan ini kami buat untuk manusia, dan tiada yang memahaminya kecuali orang yang berilmu.*”

Menurut Syaikh Abdurahman bin Nashir As'adi dalam tafsirnya ayat di atas adalah pujian untuk perumpamaan-perumpamaan yang dilontarkan, dan suatu anjuran untuk dihayati dan dipahami, dan pujian pula bagi orang yang memahaminya, dan bahwa itu merupakan tanda bahwa dia termasuk ahli ilmu (As-Sa'di, n.d.). Dengan demikian, dapat diketahui bahwa orang yang tidak memahaminya, (padahal perumpamaan itu sangat penting), adalah orang yang bukan ahli ilmu.

Dari ayat dan tafsir di atas dijelaskan bahwa pentingnya manusia agar memiliki ilmu. Dan *game* ini dirancang sebagai media pembelajaran antar manusia. Seperti pada guru mengajar muridnya. Di harapkan dengan adanya *game* ini proses ajar dan mengajar dapat berjalan semakin lancar.

3. *Mu'amalah Ma'a Alam*

Game ini memiliki tema bencana alam. Ia memberikan pengalaman kepada pemain tentang bahaya yang disebabkan oleh bencana alam, yang diilustrasikan melalui letusan gunung berapi. Selain itu, Allah juga menciptakan alam agar kita sebagai manusia bisa tinggal di bumi dengan aman dan juga agar kita bisa memanfaatkan sebaik mungkin. Seperti pada firman Allah surah an-nahl ayat 15:

وَأَلْقَى فِي الْأَرْضِ رَوَاسِيَ أَنْ تَمِيدَ بِكُمْ وَأَهْرَاءَ وَسُبُلًا لَّعَلَّكُمْ تَهْتَدُونَ ۝

“Dan Dia menancapkan gunung-gunung di bumi supaya bumi itu tidak goncang bersama kamu, (dan Dia menciptakan) sungai-sungai dan jalan-jalan agar kamu mendapat petunjuk “.

Menurut tafsir jalalain Allah menciptakan gunung-gunung tegak kokoh bersama manusia Allah juga menciptakan sungai dan jalan-jalan agar manusia bisa mendapat petunjuk(Jalalain, n.d.). Pada *game* ini juga terdapat berbagai macam buah buahan untuk menambah darah sedangkan terdapat hewan yang ketika karakter utama menabrak hewan tersebut akan mengurangi darah karakter. Ini di tunjukkan agar pemain bisa memanfaatkan buah-buahan sebagai makanan dan tidak mengganggu kehidupan hewan karena pada dasarnya semua makhluk Allah itu hidup berdampingan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari implementasi *support vector machine* sebagai pengatur level kesulitan dinamis pada *game* edukasi bencana gunung Meletus, maka dapat disimpulkan sebagai berikut:

Pada pembuatan sistem pengaturan level kesulitan pada *game* edukasi bencana gunung meletus menggunakan metode *Support Vector Machine* yang telah berhasil diterapkan, pada penerapan *game* ini membutuhkan variabel-yang digunakan untuk proses klasifikasi level. Variabel tersebut didapatkan ketika pemain bermain *game*. Dari hasil pengujian *System Usability Scale* (SUS), yang dimana dari 10 kuesioner dengan penilaian skala Likert terhadap 15 responden menghasilkan penilaian skor SUS sebesar 62.5%

5.2 Saran

Berdasarkan penelitian yang dilakukan, ditemukan beberapa kekurangan dalam proses penelitian. Oleh karena itu, penulis memiliki beberapa saran yang dapat dilakukan dalam penelitian selanjutnya.

1. Penambahan jumlah data yang digunakan, karena hal tersebut dapat mempengaruhi hasil model klasifikasi.
2. Pada penelitian ini *game* masih bersifat singleplayer. Hal ini dapat dikembangkan oleh peneliti selanjutnya ke dalam *game* multiplayer.

DAFTAR PUSTAKA

- As-Sa'di, S. A. bin N. (n.d.). *Surat Al-'Ankabut Ayat 43*.
<https://Tafsirweb.Com/7269-Surat-Al-Ankabut-Ayat-43.Html>.
- Ayu Prisma Dewi, D. (2017). Pengembangan *Game* Edukasi “Krishna Adventure” Dengan Metode Pembelajaran Menyenangkan (Joyful Learning). *It-Edu*, 2(01), 155–161.
- Brooke, J. (2020). SUS: A “Quick and Dirty” Usability Scale. *Usability Evaluation In Industry, November 1995*, 207–212.
<https://doi.org/10.1201/9781498710411-35>
- didit damayanti, pria wahyu RG, M. (2017). Hubungan Pengetahuan Tentang Manajemen Bencana Dengan Prevention Masyarakat Dalam Menghadapi Bencana Gunung Meletus Pada Kepala Keluarga Di Rt 06/Rw 01 Dusun Puncu Desa Puncu Kecamatan Puncu-Kediri Didit. *Hubungan Pengetahuan Tentang Manajemen Bencana Dengan Prevention Masyarakat Dalam Menghadapi Bencana Gunung Meletus Pada Kepala Keluarga Di Rt 06/Rw 01 Dusun Puncu Desa Puncu Kecamatan Puncu-Kediri*, 5(2), 1–8.
- Dony Novaliendry. (2013). Aplikasi *Game* Geografi Berbasis Multimedia Interaktif (Studi Kasus Siswa Kelas IX SMPN 1 RAO). *Jurnal Teknologi Dan Pendidikan*, 6(2), 106–118.
- Giantara, F., Yanti, N., Handayani, S., & Anis, Y. (2020). Pola Pendidikan keluarga Saat Bencana Kabut Asap di Kota Pekanbaru. *Jurnal Obsesi : Jurnal Pendidikan Anak Usia Dini*, 4(2), 778.
<https://doi.org/10.31004/obsesi.v4i2.446>
- H.N, I. A., Nugroho, P. I., & Ferdiana, R. (2015). Pengujian Usability Website Menggunakan *System Usability Scale*. *JURNAL IPTEKKOM : Jurnal Ilmu Pengetahuan & Teknologi Informasi*, 17(1), 31.
<https://doi.org/10.33164/iptekkom.17.1.2015.31-38>
- Haryanto, H., & Lakoro, R. (2019). Desain *Game* Imersif untuk Edukasi Mitigasi Bencana menggunakan Konsep Appreciative Learning. *Prosiding SENIATI*, 5(2), 43–48. <https://ejournal.itn.ac.id/index.php/seniati/article/view/2284>
- Jalalain. (n.d.). *An-Nahl: 15*. <https://Tafsir.Learn-Quran.Co/Id/Surat-16-an-Nahl/Ayat-15#>.
- Kasim, A. A., & Sudarsono, M. (2019). Algoritma Support Vector Machine (SVM) untuk Klasifikasi Ekonomi Penduduk Penerima Bantuan Pemerintah di

- Kecamatan Simpang Raya Sulawesi Tengah. *Seminar Nasional APTIKOM*, 568–573.
- Katsir, I. (n.d.). *Al-Qari'ah: 5*. <https://Tafsir.Learn-Quran.Co/Id/Surat-101-Al-Qariah/Ayat-5>.
- Monika Parapat, I., & Tanzil Furqon, M. (2018). Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(10), 3163–3169.
- Murdha Pamungkas, seta. (2022). *Personalisasi penyeimbang kesulitan dan skenario berbasis profil pemain pada game edukasi bencana gunung meletus menggunakan metode feedforward neural network*.
- Mustafa, J. I. (2016). *Pengembangan Game Edukatif Berbasis Android Sebagai Media Pembelajaran Akuntansi Di Kelas XI IPS SMA Negeri 1 Imogiri*.
- Putri, A. N., & Crissano, A. V. (2017). Fuzzy Controller Application For Player Movement On 3D Engine Games Based Smart Agent. *Transformatika*, 15(1), 53–59.
- Rahmawan Putra, D., & Nugroho, A. (2016). Pengembangan Game Edukatif Berbasis Android Sebagai Media Pembelajaran Akuntansi. *Jurnal Pendidikan Akuntansi Indonesia*, XIV(1), 25–34.
- Suryani, A. S. (2012). Penanganan Asap Kabut Akibat Kebakaran Hutan di Wilayah Perbatasan Indonesia. *Aspirasi: Jurnal Masalah-Masalah Sosial*, 59–76.
- Warpefelt, H. (2016). *The Non-Player Character: Exploring the believability of NPC presentation and behavior* (Issue 16).

Lampiran 1

Perhitungan kernel level 1

x,y	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	73.96	63.2	49.98	46.1	47.47	57.3	53.14	42.38	38.81	41.09	47.61	66.1	63.84	58.68	42.51	39.69	36.24	66.59	79.57	90.25	48.44	64.16	58.52	64.16	79.21	70.9	62.57	41.6	52.71	51.41
2	63.2	53.31	41.34	37.94	39.27	48.27	44.48	34.7	31.48	33.55	39.49	55.96	54	49.43	34.88	32.36	29.27	56.45	68.47	78.41	39.9	54.29	49.16	54.41	68.34	60.66	52.99	33.91	44.02	42.94
3	49.98	41.34	31.39	28.97	30.55	38.63	35.34	26.73	23.94	25.83	31.16	43.59	42.41	39.16	27.2	25.1	22.47	44.26	55.02	64	29.92	42.64	38.32	43.3	55.88	49.07	42.25	25.65	34.6	34.16
4	46.1	37.94	28.97	27.25	29.25	37.25	34.14	25.73	23.01	24.97	30.32	40	39.47	37.21	26.53	24.61	22.09	40.88	51.32	60.04	27.34	39.66	35.73	40.91	53.23	46.72	40.13	24.25	33.05	33.2
5	47.47	39.27	30.55	29.25	31.68	40.06	36.92	28.19	25.36	27.49	33.17	41.3	41.23	39.59	29.3	27.39	24.8	42.38	53.05	61.94	28.7	41.4	37.55	43.14	55.82	49.26	42.53	26.3	35.51	36.11
6	57.3	48.27	38.63	37.25	40.06	49.42	45.94	36.14	32.94	35.37	41.8	50.51	50.51	48.82	37.45	35.31	32.38	51.74	63.47	73.17	36.51	50.69	46.46	52.71	66.65	59.47	52.07	33.94	44.32	45.08
7	53.14	44.48	35.34	34.14	36.92	45.94	42.61	33.21	30.14	32.48	38.68	46.61	46.72	45.26	34.53	32.49	29.7	47.84	59.16	68.54	33.27	46.9	42.87	48.94	62.42	55.51	48.37	31.02	40.98	41.82
8	42.38	34.7	26.73	25.73	28.19	36.14	33.21	24.98	22.33	24.36	29.77	36.57	36.72	35.49	26.15	24.4	22	37.68	47.8	56.27	24.92	36.87	33.33	38.74	50.83	44.62	38.25	23.05	31.74	32.53
9	38.81	31.48	23.94	23.01	25.36	32.94	30.14	22.33	19.83	21.75	26.88	33.27	33.43	32.29	23.45	21.8	19.54	34.33	44.02	52.16	22.21	33.57	30.2	35.38	46.97	41.01	34.92	20.49	28.73	29.5
10	41.09	33.55	25.83	24.97	27.49	35.37	32.48	24.36	21.75	23.78	29.14	35.38	35.64	34.61	25.59	23.89	21.53	36.52	46.5	54.87	24.01	35.78	32.34	37.75	49.71	43.59	37.32	22.37	30.96	31.86
11	47.61	39.49	31.16	30.32	33.17	41.8	38.68	29.77	26.88	29.14	35.08	41.45	41.84	40.88	31.19	29.33	26.73	42.73	53.49	62.44	29.12	41.99	38.29	44.22	57.11	50.57	43.8	27.49	36.95	38.03
12	66.1	55.96	43.59	40	41.3	50.51	46.61	36.57	33.27	35.38	41.45	58.68	56.58	51.77	36.71	34.1	30.91	59.15	71.42	81.56	42.14	56.88	51.59	56.91	71.14	63.28	55.43	35.83	46.19	45
13	63.84	54	42.41	39.47	41.23	50.51	46.72	36.72	33.43	35.64	41.84	56.58	55.12	51.22	37.2	34.71	31.58	57.29	69.44	79.48	40.75	55.38	50.39	56.01	70.19	62.51	54.77	35.54	45.94	45.32
14	58.68	49.43	39.16	37.21	39.59	48.82	45.26	35.49	32.29	34.61	40.88	51.77	51.22	48.72	36.47	34.22	31.25	52.79	64.58	74.33	37.23	51.44	46.96	52.91	66.82	59.52	52.05	33.71	43.97	44.2
15	42.51	34.88	27.2	26.53	29.3	37.45	34.53	26.15	23.45	25.59	31.19	36.71	37.2	36.47	27.56	25.85	23.43	37.96	48.15	56.67	25.24	37.33	33.9	39.56	51.81	45.62	39.21	23.93	32.82	33.97
16	39.69	32.36	25.1	24.61	27.39	35.31	32.49	24.4	21.8	23.89	29.33	34.1	34.71	34.22	25.85	24.22	21.9	35.36	45.23	53.51	23.17	34.83	31.57	37.14	49.04	43.05	36.85	22.15	30.75	32
17	36.24	29.27	22.47	22.09	24.8	32.38	29.7	22	19.54	21.53	26.73	30.91	31.58	31.25	23.43	21.9	19.71	32.15	41.6	49.56	20.61	31.7	28.62	33.99	45.43	39.69	33.76	19.8	27.98	29.27
18	66.59	56.45	44.26	40.88	42.38	51.74	47.84	37.68	34.33	36.52	42.73	59.15	57.29	52.79	37.96	35.36	32.15	59.72	72.08	82.28	42.7	57.58	52.35	57.85	72.21	64.34	56.45	36.75	47.26	46.29
19	79.57	68.47	55.02	51.32	53.05	63.47	59.16	47.8	44.02	46.5	53.49	71.42	69.44	64.58	48.15	45.23	41.6	72.08	85.6	96.69	53.25	69.76	64.01	70.12	85.86	77.27	68.6	46.7	58.48	57.46
20	90.25	78.41	64	60.04	61.94	73.17	68.54	56.27	52.16	54.87	62.44	81.56	79.48	74.33	56.67	53.51	49.56	82.28	96.69	108.5	62.08	79.82	73.68	80.25	97.03	87.89	78.63	55.05	67.79	66.72
21	48.44	39.9	29.92	27.34	28.7	36.51	33.27	24.92	22.21	24.01	29.12	42.14	40.75	37.23	25.24	23.17	20.61	42.7	53.25	62.08	28.58	40.99	36.66	41.4	53.69	46.97	40.28	24.04	32.69	32.05
22	64.16	54.29	42.64	39.66	41.4	50.69	46.9	36.87	33.57	35.78	41.99	56.88	55.38	51.44	37.33	34.83	31.7	57.58	69.76	79.82	40.99	55.65	50.64	56.25	70.46	62.76	55	35.71	46.13	45.48
23	58.52	49.16	38.32	35.73	37.55	46.46	42.87	33.33	30.2	32.34	38.29	51.59	50.39	46.96	33.9	31.57	28.62	52.35	64.01	73.68	36.66	50.64	45.95	51.44	65.09	57.74	50.33	32.06	42	41.6
24	64.16	54.41	43.3	40.91	43.14	52.71	48.94	38.74	35.38	37.75	44.22	56.91	56.01	52.91	39.56	37.14	33.99	57.85	70.12	80.25	41.4	56.25	51.44	57.46	71.88	64.22	56.43	37.11	47.8	47.72
25	79.21	68.34	55.88	53.23	55.82	66.65	62.42	50.83	46.97	49.71	57.11	71.14	70.19	66.82	51.81	49.04	45.43	72.21	85.86	97.03	53.69	70.46	65.09	71.88	87.92	79.45	70.76	48.91	61.09	61.06
26	70.9	60.66	49.07	46.72	49.26	59.47	55.51	44.62	41.01	43.59	50.57	63.28	62.51	59.52	45.62	43.05	39.69	64.34	77.27	87.89	46.97	62.76	57.74	64.22	79.45	71.43	63.22	42.73	54.18	54.28
27	62.57	52.99	42.25	40.13	42.53	52.07	48.37	38.25	34.92	37.32	43.8	55.43	54.77	52.05	39.21	36.85	33.76	56.45	68.6	78.63	40.28	55	50.33	56.43	70.76	63.22	55.51	36.46	47.1	47.25
28	41.6	33.91	25.65	24.25	26.3	33.94	31.02	23.05	20.49	22.37	27.49	35.83	35.54	33.71	23.93	22.15	19.8	36.75	46.7	55.05	24.04	35.71	32.06	37.11	48.91	42.73	36.46	21.5	29.85	30.2
29	52.71	44.02	34.6	33.05	35.51	44.32	40.98	31.74	28.73	30.96	36.95	46.19	45.94	43.97	32.82	30.75	27.98	47.26	58.48	67.79	32.69	46.13	42	47.8	61.09	54.18	47.1	29.85	39.59	40.07
30	51.41	42.94	34.16	33.2	36.11	45.08	41.82	32.53	29.5	31.86	38.03	45	45.32	44.2	33.97	32	29.27	46.29	57.46	66.72	32.05	45.48	41.6	47.72	61.06	54.28	47.25	30.2	40.07	41.12

Lampiran 2

Matrik hessian level 1

x,y	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	18.49	15.8	-12.5	-11.53	-11.87	-14.33	-13.29	-10.6	-9.7	-10.27	-11.9	16.52	-15.96	-14.67	-10.63	-9.92	-9.06	16.65	-19.89	-22.56	-12.11	-16.04	-14.63	-16.04	-19.8	-17.72	-15.64	-10.4	-13.18	-12.85
2	15.8	13.33	-10.34	-9.49	-9.82	-12.07	-11.12	-8.67	-7.87	-8.39	-9.87	13.99	-13.5	-12.36	-8.72	-8.09	-7.32	14.11	-17.12	-19.6	-9.97	-13.57	-12.29	-13.6	-17.08	-15.16	-13.25	-8.48	-11	-10.73
3	-12.5	-10.34	7.85	7.24	7.64	9.66	8.84	6.68	5.98	6.46	7.79	-10.9	10.6	9.79	6.8	6.28	5.62	-11.06	13.75	16	7.48	10.66	9.58	10.82	13.97	12.27	10.56	6.41	8.65	8.54
4	-11.53	-9.49	7.24	6.81	7.31	9.31	8.53	6.43	5.75	6.24	7.58	-10	9.87	9.3	6.63	6.15	5.52	-10.22	12.83	15.01	6.83	9.92	8.93	10.23	13.31	11.68	10.03	6.06	8.26	8.3
5	-11.87	-9.82	7.64	7.31	7.92	10.01	9.23	7.05	6.34	6.87	8.29	-10.33	10.31	9.9	7.33	6.85	6.2	-10.6	13.26	15.49	7.17	10.35	9.39	10.78	13.96	12.31	10.63	6.58	8.88	9.03
6	-14.33	-12.07	9.66	9.31	10.01	12.36	11.49	9.04	8.23	8.84	10.45	-12.63	12.63	12.2	9.36	8.83	8.09	-12.93	15.87	18.29	9.13	12.67	11.61	13.18	16.66	14.87	13.02	8.49	11.08	11.27
7	-13.29	-11.12	8.84	8.53	9.23	11.49	10.65	8.3	7.53	8.12	9.67	-11.65	11.68	11.31	8.63	8.12	7.43	-11.96	14.79	17.13	8.32	11.72	10.72	12.24	15.61	13.88	12.09	7.75	10.24	10.45
8	-10.6	-8.67	6.68	6.43	7.05	9.04	8.3	6.24	5.58	6.09	7.44	-9.14	9.18	8.87	6.54	6.1	5.5	-9.42	11.95	14.07	6.23	9.22	8.33	9.68	12.71	11.15	9.56	5.76	7.93	8.13
9	-9.7	-7.87	5.98	5.75	6.34	8.23	7.53	5.58	4.96	5.44	6.72	-8.32	8.36	8.07	5.86	5.45	4.88	-8.58	11	13.04	5.55	8.39	7.55	8.84	11.74	10.25	8.73	5.12	7.18	7.38
10	-10.27	-8.39	6.46	6.24	6.87	8.84	8.12	6.09	5.44	5.94	7.29	-8.84	8.91	8.65	6.4	5.97	5.38	-9.13	11.63	13.72	6	8.95	8.08	9.44	12.43	10.9	9.33	5.59	7.74	7.96
11	-11.9	-9.87	7.79	7.58	8.29	10.45	9.67	7.44	6.72	7.29	8.77	-10.36	10.46	10.22	7.8	7.33	6.68	-10.68	13.37	15.61	7.28	10.5	9.57	11.06	14.28	12.64	10.95	6.87	9.24	9.51
12	16.52	13.99	-10.9	-10	-10.33	-12.63	-11.65	-9.14	-8.32	-8.84	-10.36	14.67	-14.15	-12.94	-9.18	-8.53	-7.73	14.79	-17.86	-20.39	-10.54	-14.22	-12.9	-14.23	-17.78	-15.82	-13.86	-8.96	-11.55	-11.25
13	-15.96	-13.5	10.6	9.87	10.31	12.63	11.68	9.18	8.36	8.91	10.46	-14.15	13.78	12.81	9.3	8.68	7.9	-14.32	17.36	19.87	10.19	13.85	12.6	14	17.55	15.63	13.69	8.89	11.49	11.33
14	-14.67	-12.36	9.79	9.3	9.9	12.2	11.31	8.87	8.07	8.65	10.22	-12.94	12.81	12.18	9.12	8.55	7.81	-13.2	16.14	18.58	9.31	12.86	11.74	13.23	16.7	14.88	13.01	8.43	10.99	11.05
15	-10.63	-8.72	6.8	6.63	7.33	9.36	8.63	6.54	5.86	6.4	7.8	-9.18	9.3	9.12	6.89	6.46	5.86	-9.49	12.04	14.17	6.31	9.33	8.47	9.89	12.95	11.4	9.8	5.98	8.21	8.49
16	-9.92	-8.09	6.28	6.15	6.85	8.83	8.12	6.1	5.45	5.97	7.33	-8.53	8.68	8.55	6.46	6.06	5.48	-8.84	11.31	13.38	5.79	8.71	7.89	9.28	12.26	10.76	9.21	5.54	7.69	8
17	-9.06	-7.32	5.62	5.52	6.2	8.09	7.43	5.5	4.88	5.38	6.68	-7.73	7.9	7.81	5.86	5.48	4.93	-8.04	10.4	12.39	5.15	7.92	7.16	8.5	11.36	9.92	8.44	4.95	7	7.32
18	16.65	14.11	-11.06	-10.22	-10.6	-12.93	-11.96	-9.42	-8.58	-9.13	-10.68	14.79	-14.32	-13.2	-9.49	-8.84	-8.04	14.93	-18.02	-20.57	-10.67	-14.39	-13.09	-14.46	-18.05	-16.09	-14.11	-9.19	-11.82	-11.57
19	-19.89	-17.12	13.75	12.83	13.26	15.87	14.79	11.95	11	11.63	13.37	-17.86	17.36	16.14	12.04	11.31	10.4	-18.02	21.4	24.17	13.31	17.44	16	17.53	21.47	19.32	17.15	11.68	14.62	14.36
20	-22.56	-19.6	16	15.01	15.49	18.29	17.13	14.07	13.04	13.72	15.61	-20.39	19.87	18.58	14.17	13.38	12.39	-20.57	24.17	27.11	15.52	19.95	18.42	20.06	24.26	21.97	19.66	13.76	16.95	16.68
21	-12.11	-9.97	7.48	6.83	7.17	9.13	8.32	6.23	5.55	6	7.28	-10.54	10.19	9.31	6.31	5.79	5.15	-10.67	13.31	15.52	7.14	10.25	9.17	10.35	13.42	11.74	10.07	6.01	8.17	8.01
22	-16.04	-13.57	10.66	9.92	10.35	12.67	11.72	9.22	8.39	8.95	10.5	-14.22	13.85	12.86	9.33	8.71	7.92	-14.39	17.44	19.95	10.25	13.91	12.66	14.06	17.61	15.69	13.75	8.93	11.53	11.37
23	-14.63	-12.29	9.58	8.93	9.39	11.61	10.72	8.33	7.55	8.08	9.57	-12.9	12.6	11.74	8.47	7.89	7.16	-13.09	16	18.42	9.17	12.66	11.49	12.86	16.27	14.44	12.58	8.02	10.5	10.4
24	-16.04	-13.6	10.82	10.23	10.78	13.18	12.24	9.68	8.84	9.44	11.06	-14.23	14	13.23	9.89	9.28	8.5	-14.46	17.53	20.06	10.35	14.06	12.86	14.36	17.97	16.06	14.11	9.28	11.95	11.93
25	-19.8	-17.08	13.97	13.31	13.96	16.66	15.61	12.71	11.74	12.43	14.28	-17.78	17.55	16.7	12.95	12.26	11.36	-18.05	21.47	24.26	13.42	17.61	16.27	17.97	21.98	19.86	17.69	12.23	15.27	15.27
26	-17.72	-15.16	12.27	11.68	12.31	14.87	13.88	11.15	10.25	10.9	12.64	-15.82	15.63	14.88	11.4	10.76	9.92	-16.09	19.32	21.97	11.74	15.69	14.44	16.06	19.86	17.86	15.8	10.68	13.54	13.57
27	-15.64	-13.25	10.56	10.03	10.63	13.02	12.09	9.56	8.73	9.33	10.95	-13.86	13.69	13.01	9.8	9.21	8.44	-14.11	17.15	19.66	10.07	13.75	12.58	14.11	17.69	15.8	13.88	9.12	11.77	11.81
28	-10.4	-8.48	6.41	6.06	6.58	8.49	7.75	5.76	5.12	5.59	6.87	-8.96	8.89	8.43	5.98	5.54	4.95	-9.19	11.68	13.76	6.01	8.93	8.02	9.28	12.23	10.68	9.12	5.37	7.46	7.55
29	-13.18	-11	8.65	8.26	8.88	11.08	10.24	7.93	7.18	7.74	9.24	-11.55	11.49	10.99	8.21	7.69	7	-11.82	14.62	16.95	8.17	11.53	10.5	11.95	15.27	13.54	11.77	7.46	9.9	10.02
30	-12.85	-10.73	8.54	8.3	9.03	11.27	10.45	8.13	7.38	7.96	9.51	-11.25	11.33	11.05	8.49	8	7.32	-11.57	14.36	16.68	8.01	11.37	10.4	11.93	15.27	13.57	11.81	7.55	10.02	10.28