

**IMPLEMENTASI *SUPPORT VECTOR MACHINE* MENGGUNAKAN  
*SEQUENTIAL MINIMUM OPTIMIZATION* SEBAGAI PENGATUR  
*LEVEL KESULITAN* PADA *GAME* EDUKASI  
BENCANA GUNUNG MELETUS**

**SKRIPSI**

**Oleh:  
DERIDA ELFRAIM FALAHIAN  
NIM. 16650082**



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

**IMPLEMENTASI *SUPPORT VECTOR MACHINE* MENGGUNAKAN  
*SEQUENTIAL MINIMUM OPTIMIZATION* SEBAGAI PENGATUR  
*LEVEL KESULITAN* PADA *GAME* EDUKASI  
BENCANA GUNUNG MELETUS**

**SKRIPSI**

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

Oleh :  
**DERIDA ELFRAIM FALAHIAN**  
NIM. 16650082

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

**HALAMAN PERSETUJUAN**

**IMPLEMENTASI *SUPPORT VECTOR MACHINE* MENGGUNAKAN  
*SEQUENTIAL MINIMUM OPTIMIZATION* SEBAGAI PENGATUR  
*LEVEL KESULITAN* PADA *GAME* EDUKASI  
*BENCANA GUNUNG MELETUS***

**SKRIPSI**

Oleh :  
**DERIDA ELFRAIM FALAHIAN**  
NIM. 16650082

Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal: 14 Juni 2023

Pembimbing I,



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

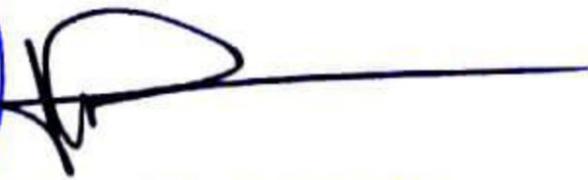
Pembimbing II,



Dr. M. Imamudin Lc, MA  
NIP. 19740602 2009011 1 010

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



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

## HALAMAN PENGESAHAN

### IMPLEMENTASI *SUPPORT VECTOR MACHINE* MENGGUNAKAN *SEQUENTIAL MINIMUM OPTIMIZATION* SEBAGAI PENGATUR LEVEL KESULITAN PADA *GAME* EDUKASI BENCANA GUNUNG MELETUS

#### SKRIPSI

Oleh:

**DERIDA ELFRAIM FALAHIAN**  
NIM. 16650082

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom )  
Tanggal: 26 Juni 2023

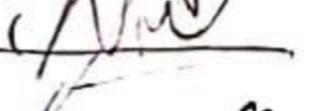
#### Susunan Dewan Penguji

Ketua Penguji : Puspa Miladin Nuraida Safitri, M. Kom  
NIP. 19930828 201903 2 018

Anggota Penguji I : Dr. Yunifa Miftachul Arif, M. T  
NIP. 19830616 201101 1 004

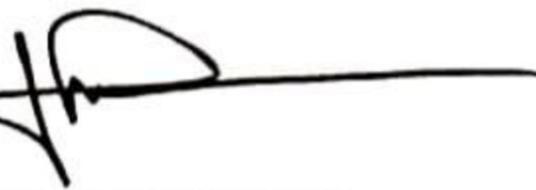
Anggota Penguji II : Dr. Fresy Nugroho, M. T  
NIP. 19710722 201101 1 001

Anggota Penguji III : Dr. M. Imamudin Lc, MA  
NIP. 19740602 2009011 1 010

()  
()  
()  
()

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 : Derida Elfrain Falahian  
NIM : 16650082  
Fakultas : Sains dan Teknologi  
Jurusan : Teknik Informatika  
Judul Skripsi : Implementasi *Support Vector Machine* Menggunakan *Sequential Minimum Optimization* Sebagai Pengatur *Level Kesulitan* Pada *Game* Edukasi Gunung Meletus

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar 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, 26 Juni 2023

Yang membuat pernyataan,



Derida Elfrain Falahian

NIM. 16650082

## **MOTTO**

“Hidup bukanlah masalah yang harus diselesaikan, tetapi sebuah realitas yang harus dijalani dengan bijaksana”– (Søren Kierkegaard)

## **HALAMAN PERSEMBAHAN**

**Puji syukur kehadiran Allah SWT, shalawat dan salam bagi Rasul-Nya.**

Saya persembahkan skripsi ini kepada:

Kedua orang tua saya Almarhum Prof. Dr. H. Mohamad Nur Yasin, SH, MA. dan Dra. Enik Faridaturrohmah, M.H. yang saya sayangi. Terima kasih banyak atas kesabarannya dan do'a yang mengiringi perjalanan saya serta memberikan dukungan juga semangat kepada saya. Semoga kebaikan dan rahmat Allah SWT selalu tercurah kepada mereka semua Amin.

Dosen pembimbing saya, Dr. Fresy Nugroho, M.T dan Dr. M. Imamudin Lc, MA. yang telah bersedia membimbing saya dengan penuh kesabaran dan memberikan masukan ilmu yang bermanfaat sehingga skripsi ini dapat terselesaikan.

Seluruh Dosen Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah membimbing dan memberikan ilmu kepada saya selama perkuliahan. Tidak lupa terima kasih juga untuk teman-teman Teknik Informatika, terutama angkatan 2016 yang telah berjuang bersama melewati skripsi ini.

## KATA PENGANTAR

سَمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

Segala puji bagi Allah SWT, karena atas rahmad, hidayah serta karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul “**Implementasi Support Vector Machine Menggunakan Sequential Minimum Optimization Sebagai Pengatur Level Kesulitan Pada Game Edukasi Bencana Gunung Meletus**” sebagai salah satu syarat untuk memperoleh gelar sarjana pada program studi di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulis mengucapkan terima kasih banyak kepada semua pihak yang telah membantu terselesaikannya skripsi ini. Maka dari itu dengan segenap kerendahan hati penulis mengucapkan terima kasih kepada:

1. Prof. Dr. M. Zainuddin, MA selaku Rektor Universitas Islam Negeri 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 ST., M.MT., IPM selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Fresy Nugroho, M.T dan Dr. M. Imamudin Lc, MA. selaku dosen pembimbing skripsi yang telah senantiasa meluangkan waktu untuk membimbing, mengarahkan penulis, dan memberikan masukan dalam penyusunan skripsi ini hingga selesai.

5. Seluruh Dosen dan Jajaran Staf Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang yang memberikan ilmu pengetahuan serta pengalaman.
6. Ayahanda dan Ibunda tercinta yang mengiringi do'a kepada penulis untuk menuntut ilmu dan mendukung serta memberikan semangat selama kuliah.
7. Teman - teman Teknik Informatika angkatan 2016 yang turut menemani proses dalam penyelesaian skripsi ini.
8. Teman – teman seperbimbingan yang membantu dan memberi motivasi dalam proses penyelesaian skripsi.

Penulis menyadari dalam penyusunan skripsi ini masih terdapat banyak sekali kekurangan. Semoga skripsi ini bermanfaat bagi para pembaca. Amin Ya Rabbal Alamin.

## 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
المخلص.....	xvi
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Tujuan Penelitian .....	5
1.4 Batasan Penelitian .....	5
1.5 Manfaat Penelitian .....	6
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1 Penelitian Terkait .....	7
2.2 Landasan Teori.....	8
2.2.1 <i>Game</i> .....	8
2.2.2 <i>Non-Player Character (NPC)</i> .....	9
2.2.3 <i>Asap Kabut</i> .....	10
2.2.4 <i>Taksonomi Bloom</i> .....	12
2.2.5 <i>Tingkat Kesulitan Level Otomatis</i> .....	16
2.2.6 <i>Support Vector Machine (SVM)</i> .....	16
2.2.7 <i>Sequential Training</i> .....	17
2.2.8 <i>Sequential Minimum Optimization (SMO)</i> .....	17
<b>BAB III DESAIN DAN RANCANGAN SISTEM.....</b>	<b>23</b>
3.1 Skenario Penelitian .....	23
3.2 Perancangan Desain Sistem <i>Game</i> .....	24
3.2.1 <i>Desain Menu Flow</i> .....	24
3.2.2 <i>Gameplay Flowchart</i> .....	26
3.3 Perancangan <i>Storyboard</i> .....	27
3.4 Perancangan Implementasi <i>Support Vector Machine (SVM)</i> .....	28
3.5 Implementasi <i>Support Vector Machine (SVM)</i> .....	30
3.6 Rancangan Pengujian.....	39
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>41</b>
4.1 Implementasi <i>User Interface</i> .....	41
4.1.1 <i>Tampilan Awal Game</i> .....	41
4.1.2 <i>Tampilan Gameplay</i> .....	42
4.1.3 <i>Tampilan Kuis</i> .....	42

4.1.4	Tampilan Ketika Menang.....	43
4.1.5	Tampilan Ketika Kalah .....	43
4.2	Implementasi <i>Support Vector Machine</i> (SVM) .....	44
4.3	Pengujian.....	46
4.3.1	<i>Dataset</i> .....	46
4.3.2	Pengujian.....	48
4.4	Integrasi dengan Islam .....	53
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>57</b>
5.1	Kesimpulan .....	57
5.2	Saran .....	58
<b>DAFTAR PUSTAKA</b>		

## DAFTAR GAMBAR

Gambar 3. 1 Blok Diagram Skenario Penelitian .....	23
Gambar 3. 2 <i>Flowchart Gameplay</i> .....	26
Gambar 3. 3 Blok Diagram Rancangan Implementasi SVM .....	29
Gambar 4. 1 Tampilan Awal <i>Game</i> .....	41
Gambar 4. 2 Tampilan <i>Gameplay</i> .....	42
Gambar 4. 3 Tampilan Kuis .....	42
Gambar 4. 4 Tampilan Menang .....	43
Gambar 4. 5 Tampilan Kalah .....	43

## DAFTAR TABEL

Tabel 2. 1 Densitas 12 Citra Uji.....	11
Tabel 2. 2 Fitur Kabut dalam <i>Game</i> .....	11
Tabel 2. 3 Kuesioner Aspek Kognitif .....	13
Tabel 2. 4 Kuesioner Aspek Afektif .....	14
Tabel 2. 5 Kuesioner Aspek Psikomotorik .....	15
Tabel 3. 1 Data Latih.....	30
Tabel 3. 2 Hasil Perhitungan Normalisasi Data .....	31
Tabel 3. 3 Nilai Variansi dari Setiap Fitur .....	31
Tabel 3. 4 Hasil Perhitungan <i>Kernel</i> .....	32
Tabel 3. 5 Desain <i>One vs All</i> Pada SMO .....	33
Tabel 3. 6 Hasil Akhir Nilai <i>ai</i> Tahap Pertama .....	37
Tabel 3. 7 Hasil Akhir Nilai <i>ai</i> Tahap Kedua.....	37
Tabel 3. 8 Data Uji .....	38
Tabel 3. 9 <i>Kernel</i> Uji.....	38
Tabel 4. 1 Penerapan SVM menggunakan <i>Library Python</i> .....	44
Tabel 4. 2 <i>Dataset</i> Pengujian .....	46
Tabel 4. 3 Hasil Pengujian SVM .....	49
Tabel 4. 4 Hasil <i>Confusion Matrix</i> antara RF dengan SMO .....	50
Tabel 4. 5 Jumlah TP, FN, FP, dan TN.....	50
Tabel 4. 6 Hasil <i>Confusion Matrix</i> antara ST dengan SMO .....	51
Tabel 4. 7 Jumlah TP, FN, FP, dan TN.....	52

## ABSTRAK

Falahian, Derida Elfraim. 2023. **Implementasi *Support Vector Machine* Menggunakan *Sequential Minimum Optimization* Sebagai Pengatur Level Kesulitan pada *Game* Edukasi 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) Dr. M. Imamudin Lc, MA.

---

*Kata Kunci: Game, Support Vector Machine, SMO.*

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*. 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* dengan model pelatihan *Sequential Minimum Optimization (SMO)*. Dikarenakan metode ini terbukti mampu memprediksi dan menghasilkan *level* dengan tingkat akurasi yang cukup tinggi. Metode ini di gunakan sebagai penyesuaian tingkat kesulitan *level* yang di gambarkan dengan intensitas kabut yang semakin menebal dan pertanyaan secara otomatis yang disesuaikan dengan kemampuan pemainnya. Data masukan pada *game* ini terdapat 7 variabel diantaranya *skor*, darah pemain, waktu, jumlah barang, tipe barang, jumlah musuh dan tipe musuh. Setelah didapatkan nilai dari masing-masing variabel kemudian data tersebut di olah menggunakan metode *Support Vector Machine* yang di bantu dengan *algoritma Sequential Minimum Optimization* untuk mendapatkan *level* kesulitan pada *game*.

## ABSTRACT

Falahian, Derida Elfraim. 2023. **The Implementation Of Support Vector Machine Utilizing Sequential Minimum Optimization As Regulator of Difficulty Level In The Educational Game Of Volcanic Eruption.** Thesis. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisor: (I) Dr. Fresy Nugroho, M.T (II) Dr. M. Imamudin Lc, MA.

---

A volcanic eruption is a phenomenon that is caused by magma deposits in the bowels of the earth and is ejected by gasses that have high strength. The volcanic eruptions are one of the most devastating disasters. In Indonesia, the volcanic eruptions are one of the fearest disasters therefore to train the management of volcanic disaster, the community is given education in the form of. The advantages of the *game* as a socialization place providing practical experience in order to comprehend the materials easily. This *game* applies the Support Vector Machine method with the Sequential Minimum Optimization (SMO) training model. It is because this model is capable of predicting and generating *levels* with a high accuracy. This model is applied as an adjustment of the difficulty *level* described by the fog intensity which gets thicker and the questions are automatically adjusted according to the *player's* abilities. There are seven variables of input data consisting of *skor*, *player* health, time, number of items, type of items, number of enemies and type of enemies. After obtaining the value of each variable, the data is processed with the Support Vector Machine method which is assisted by the Sequential Minimum Optimization algorithm to obtain the difficulty *level* in the *game*.

*Keywords: Educational Game, Support Vector Machine, SMO.*

## المخلص

فلاحيان ، دريدا آلفرايم. ٢٠٢٣ . تطبيق آلة المتجهات الداعمة باستخدام التحسين المتسلسل الأدنى كجهاز تحكم في مستوى الصعوبة في لعبة بركان تعليمية طروحه. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانج. مشرف: (١) د. فريزي نوجروهو، الماجستير (٢) د. محمد إمام الدين لسانسي ، الماجستير

الكلمات الرئيسية: الألعاب التعليمية ، سوبورت فيكتور ماجهيني، SMO

الجبال المتفجرة هي ظواهر طبيعية تحدث بسبب رواسب الصهارة في أحشاء الأرض ويتم إخراجها بواسطة الغازات ذات القوة العالية. الانفجارات البركانية هي واحدة من أكثر الكوارث الطبيعية تدميرا. في إندونيسيا، تعد الجبال المتفجرة واحدة من الكوارث التي يخشاها الكثير من الناس، وبالتالي للتدريب على إدارة الكوارث، يتم إعطاء المجتمع التعليم في شكل لعبة. مزايا الألعاب كوسيلة للتنشئة الاجتماعية للمجتمع يمكن أن توفر خبرة عملية حتى يتمكن المشاركون من فهم المادة بسهولة. تستخدم هذه اللعبة طريقة آلة المتجه الداعم *Support Vector Machine* مع نموذج تدريب الحد الأدنى المتسلسل للتحسين *Sequential Minimum Optimization (SMO)*. لأن هذه الطريقة أثبتت قدرتها على التنبؤ وإنتاج المستويات بمستوى عال من الدقة. استخدمت هذه الطريقة كتعديل لصعوبة المستوى الذي يتم تصويده مع شدة الضباب الذي يتخون والأسئلة التي يتم تعديلها تلقائيا لقدرة اللاعب. تحتوي بيانات الإدخال في هذه اللعبة على ٧ متغيرات بما في ذلك النتيجة ودم اللاعب والوقت وعدد العناصر ونوعها وعدد الأعداء ونوعها. بعد الحصول على قيمة كل متغير، تمت معالجة البيانات باستخدام طريقة آلة المتجه الداعم التي تساعدها خوارزمية الحد الأدنى المتسلسل للتحسين للحصول على مستوى صعوبة اللعبة.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Bencana alam adalah sebuah peristiwa yang mengganggu dan merugikan kehidupan masyarakat yang di sebabkan oleh alam antara lain berupa gempa bumi, tsunami, gunung meletus, dsb. Gunung Meletus termasuk dalam salah satu bencana alam yang paling mengerikan yang di sebabkan oleh gunung berapi. Gunung berapi adalah lubang kepundan atau rekahan dalam kerak bumi tempat keluarnya cairan magma atau gas atau cairan lainnya ke permukaan bumi. Material yang diteruskan ke permukaan bumi umumnya membentuk kerucut terpancung (Christie Gosal et al., 2018).

Bukti yang menunjukkan kerentanan yang signifikan dari gunung berapi di Indonesia dapat disimpulkan dari lokasi dan posisi geografisnya, karena terletak di sepanjang jalur pertemuan lempeng tektonik, membentuk apa yang biasa disebut sebagai cincin api. Perlu dicatat bahwa Indonesia adalah rumah bagi 13% gunung berapi dunia yang luar biasa, yang terdiri dari 129 gunung berapi yang tergolong aktif dan 500 yang tergolong tidak aktif. (Christie Gosal et al., 2018).

Di Indonesia penanggulangan akan sebuah bencana alam sudah di jelas pada Undang-Undang Republik Indonesia nomor 24 tahun 2007, sebagai warga negara Indonesia, penting bagi kita untuk mengambil langkah-langkah dalam meminimalkan potensi risiko bencana. Salah satu langkah tersebut melibatkan fokus pada pendidikan dasar, terutama mulai dari masa kanak-kanak, karena dapat berfungsi sebagai alat yang berharga untuk membekali individu dengan

pengetahuan dan keterampilan yang diperlukan untuk menangani bencana secara efektif. Hal ini dapat dicapai, sebagian, melalui penggunaan permainan edukatif yang berfungsi sebagai contoh praktis kesiapsiagaan bencana. (Muyasaroh & Sudarmilah, 2019).

Pengembangan *game* edukasi adalah usaha yang menawan. *Game* edukasi menawarkan banyak keuntungan dibandingkan pendekatan pendidikan tradisional. Manfaat utama terletak pada kemampuan mereka untuk menyajikan masalah dunia nyata secara visual. Hal ini ditunjukkan oleh *Massachusetts Institute of Technology* (MIT) melalui proyek *game* mereka, *Scratch* pada dasarnya secara efektif meningkatkan kemampuan berpikir logis dan pemecahan masalah pemain. Temuan penelitian sebelumnya tidak diragukan lagi bahwa permainan edukatif memiliki potensi untuk meningkatkan proses pendidikan. Permainan ini unggul dalam berbagai aspek, terutama melalui penyertaan animasi yang dapat meningkatkan retensi memori. Akibatnya, anak-anak dapat mempertahankan materi pelajaran untuk waktu yang lebih lama dibandingkan dengan metode pengajaran konvensional. (Haryanto & Sari, 2012). Menurut Islam hal ini sangat sesuai dengan hadist Rasulullah SAW tentang manfaat pembelajaran menggunakan media.

مَنْ أَرَادَ الدُّنْيَا فَعَلَيْهِ بِالْعِلْمِ، وَمَنْ أَرَادَ الْآخِرَةَ فَعَلَيْهِ بِالْعِلْمِ، وَمَنْ أَرَادَهُمَا فَعَلَيْهِ بِالْعِلْمِ

“Barangsiapa yang hendak menginginkan dunia, maka hendaklah ia menguasai ilmu. Barangsiapa yang hendak menginginkan akhirat, maka hendaklah ia menguasai ilmu, dan barangsiapa yang menginginkan keduanya (dunia dan akhirat) hendaklah ia menguasai ilmu” (HR. Ahmad).

Di Indonesia, permainan edukatif mayoritas berbentuk permainan balok, permainan yang menampilkan figur makhluk hidup, dan permainan yang berfokus pada pengenalan benda sehari-hari. (Rima Ariona N, 2014).

Konsep dari penelitian ini sesuai dengan konsep pada penelitian terdahulu (Syafri Pratama, 2022). Terdapat beberapa penelitian tentang pengaturan *level* pada *game* yang salah satunya, yaitu dengan menerapkan metode *Random Forest*. Penerapan metode *Random Forest* ini melibatkan penggabungan berbagai teknik pohon keputusan bersama untuk membuat model yang lebih terperinci. (Adrian et al., 2021). Metode Random Forest secara khusus dibentuk dengan menggabungkan beberapa teknik pohon keputusan menjadi model terpadu. Metode ini umumnya digunakan dalam *game* untuk mengklasifikasikan volume data yang substansial secara efektif. (Syafri Pratama, 2022).

Dalam studi ini, peneliti memanfaatkan metode *Support Vector Machine* (SVM) untuk mengatur tingkat keluaran yang meliputi tingkat kabut (dalam rentang level 1-10). Dipilihnya metode SVM dikarenakan metode tersebut sanggup memecahkan masalah klasifikasi *multiclass*. Namun SVM sendiri memiliki kekurangan yaitu, SVM membutuhkan waktu yang cukup lama dalam proses pelatihannya, hal ini dikarenakan adanya proses *Quadratic Problem* (QP) sehingga untuk mengatasi hal tersebut di gunakanlah algoritma *Sequential Minimum Optimization* (SMO) supaya proses pelatihan menjadi lebih *efisien* (Afrizal Laksita Akbar, 2015).

Jika dibandingkan dengan metode *Random Forest*, metode SVM mampu mengklasifikasi dan memprediksi jauh lebih baik. Hal ini di sampaikan oleh

(Adrian et al., 2021) pada penelitiannya berjudul “Perbandingan Metode Klasifikasi *Random Forest* dan *Support Vector Machine* pada Analisis Sentimen PSBB”, Usai mengevaluasi setiap model, terdapat pengamatan bahwa model *Random Forest* mencapai skor yang lebih tinggi, namun gagal mengenali *tweet* yang ditandai sebagai "Positif". Sebaliknya, model *Support Vector Machine* memperoleh skor yang lebih rendah namun berhasil mengidentifikasi *tweet* berlabel "Positif" dengan baik. Oleh karena itu, kesimpulannya adalah *Support Vector Machine* lebih unggul dibandingkan dengan model *Random Forest* karena kemampuannya dalam mengenali *tweet* dengan label "Positif".

Dengan mempertimbangkan penjelasan latar belakang yang telah disampaikan, penulis memilih judul penelitian. **“Implementasi *Support Vector Machine* Menggunakan *Sequential Minimum Optimization* Sebagai Pengatur Level Kesulitan Pada *Game* Edukasi Bencana Alam”**.

## 1.2 Rumusan Masalah

Dengan mempertimbangkan penjelasan latar belakang penelitian yang telah dijabarkan sebelumnya, rumusan masalah yang dapat diajukan adalah apakah metode *Support Vector Machine* dengan menggunakan model pelatihan *Sequential Minimum Optimization* dapat diterapkan untuk menyesuaikan tingkat kesulitan pada permainan edukasi yang melibatkan variabel *input* seperti skor, waktu, darah, jumlah musuh, tipe musuh, jumlah *item*, dan tipe *item*, dengan keluaran berupa tingkat kesulitan dalam bentuk *level* kabut dan pertanyaan.

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menerapkan metode *Sequential Minimum Optimization* pada *Support Vector Machine* dalam mengatur tingkat kesulitan pada permainan edukasi tentang gunung meletus.

## 1.4 Batasan Penelitian

Setelah mengidentifikasi masalah yang telah disebutkan sebelumnya, dalam penelitian ini akan ditetapkan batasan masalah sebagai berikut:

1. *Software* yang digunakan adalah Unity 3D.
2. *Game* berjenis edukasi.
3. Terdapat tujuh variabel masukan yang digunakan pada *game* ini, yaitu skor, waktu, darah pemain, barang yang dikumpulkan, nilai barang, jenis musuh, dan jumlah musuh.
4. Pada *game* ini menghasilkan empat jenis keluaran, yaitu level kabut, pertanyaan yang mengandung aspek afektif, kognitif, dan psikomotorik.

### **1.5 Manfaat Penelitian**

Dampak positif dari penelitian ini adalah meningkatnya daya tarik dalam permainan melalui pembuatan tingkatan yang lebih menantang. Dengan demikian, para pemain akan merasa terasah dan tertantang karena semakin tinggi level yang semakin sulit.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

*Support Vector Machine* (SVM) adalah sebuah teknik pembelajaran yang sering digunakan untuk mengklasifikasikan data dan cocok untuk *dataset* yang hanya memiliki dua kelas. Untuk memisahkan data antara kelas, SVM menggunakan *hyperplane* (bidang pemisah) yang dibangun berdasarkan data yang ada. *Hyperplane* yang baik harus memiliki margin (batasan) yang paling besar, bukan hanya dapat memisahkan data. Mencari *hyperplane* terbaik adalah inti dari SVM. Namun, dalam mencari *hyperplane* tersebut, terdapat permasalahan baru yaitu *Quadratic Programming*, sebuah formula yang sulit untuk dipecahkan (Platt, 1999).

Jajang dkk menjelaskan dalam penelitiannya metode *Support Vector Machine* digunakan untuk mengatasi keterbatasan data, prediksi, dan klasifikasi karena ukuran sampel yang terbatas, dan untuk meningkatkan akurasi dan meminimalkan kesalahan, model *kernel* digunakan untuk memvisualisasikan data siswa. Visualisasi ini meliputi penggunaan diagram batang, histogram, dan diagram peningkatan. Metode ini merupakan salah satu pendekatan yang digunakan untuk memenuhi kriteria kelulusan minimum. (Purnama et al., 2020).

## 2.2 Landasan Teori

### 2.2.1 *Game*

Dalam bahasa Indonesia *game* diartikan sebagai “permainan” atau kegiatan bermain. Permainan adalah aktivitas kompleks yang mencakup faktor menang, kalah, aturan, tantangan, dan berbagai faktor lain sesuai dengan *game* yang dibuat. Dalam hal ini *game* adalah sebuah sistem dimana pemain bisa berpartisipasi dalam sebuah konflik yang di rekayasa dari sebuah kejadian nyata. Pemain dapat berinteraksi dengan sistem *game* dan terlibat dengan konflik yang disajikan dalam *game*. Ketentuan dalam *game* didesain untuk mengatur perilaku para *player* dan menentukan seberapa jauh perkembangan mereka dalam menyelesaikan sebuah konflik atau masalah dalam *game*. (Asmiatun dan Putri, 2017).

Tujuan *game* adalah untuk memberikan hiburan kepada setiap individu dan untuk aspek ini *game* dapat dinikmati oleh berbagai kalangan umur dari anak-anak hingga orang dewasa. *Game* sendiri menawarkan berbagai manfaat karena dapat membantu perkembangan kemampuan dalam hal kognitif dan meningkatkan konsentrasi, hal ini memungkinkan setiap individu memecahkan masalah secara efisien dan cepat. Faktor tersebut disebabkan oleh adanya suatu konflik dan unsur pemecahan masalah di dalam *game*. Namun, *game* juga bisa menimbulkan dampak yang negatif, terutama ketika seseorang menjadi terlalu sering memainkannya dapat menimbulkan sebuah rutinitas buruk yang dapat mengganggu aktifitas seseorang sehari-hari, dalam kasus tersebut merupakan dampak negative dari bermain *game*, yaitu kecanduan. Kecanduan seperti itu dapat menyebabkan pengabaian waktu dan

berbagai aktifitas terjadwal lainnya terganggu bahkan mungkin untuk tertunda. (Asmiatun dan Putri, 2017).

### **2.2.2 *Non-Player Character (NPC)***

Dalam sebuah *game* terdapat banyak sekali faktor atau unsur yang dapat mempengaruhi sebuah *gameplay* dari para *player*. Salah contoh faktor yang mempengaruhi jalanya permainan dalam sebuah *game* adalah *Non Player Character* atau NPC yang biasa terdengar familiar oleh seseorang yang sering bermain *game*. NPC sendiri adalah istilah yang digunakan untuk memvisualisasikan karakter dalam video *game* yang dikendalikan oleh desain sistem komputer dan tidak dapat di interupsi oleh *player*. Konsep ini ada sebelum era *game* digital. Dalam *game* digital terdapat beragam NPC, mulai dari karakter biasa hingga fantastik, dari yang kecil hingga besar, bahkan karakter *non-manusia*. NPC adalah sebuah entitas yang dirancang menyerupai pemain dan mampu menunjukkan *skill* perlawanan yang dapat menyesuaikan dari *skill player* jika dilengkapi dengan *artificial intelligence (AI)*. Oleh karena itu NPC secara aktif terlibat dalam memerankan sebuah peran mereka dan dapat berperilaku sesuai kondisi yang ditentukan agar dapat meyakinkan *player* bahwa keberadaan mereka cukup realistis dalam dunia *game*. Hal ini seolah-olah menyatakan bahwa NPC perlu menampilkan beberapa aspek dari apa yang sering disebut sebagai kepribadian mereka (Warpefelt, 2016)

### 2.2.3 Asap Kabut

Di dunia ini terdapat banyak sekali fenomena alam yang membuat setiap individu sangat takjub dan penasaran bagaimana suatu fenomena alam bisa muncul dan menghilang dengan sendirinya. Kabut merupakan salah satu fenomena alam yang sering di jumpai, terutama di daerah dengan curah hujan yang cukup tinggi. Kabut sendiri adalah sebuah fenomena alam yang terjadi pada atmosfer bumi yang terlihat seperti asap dan mempunyai durasi yang lama dengan cakupan yang luas di suatu wilayah karena berbagai faktor yang berkontribusi. Umumnya kabut tersusun oleh beberapa senyawa yang terurai menjadi beberapa bentuk oksidasi nitrogen, termasuk nitrogen dioksida, ozon troposfer, dan senyawa organik *volatile*. Senyawa organik *volatile* adalah sebuah zat atau senyawa hasil ekstraksi dari bahan bakar minyak, cat, pestisida dan berbagai bahan kimia yang menguap. Sedangkan sebagian besar nitrogen oksida terbentuk oleh aktifitas pembakaran bahan bakar fosil yang sering digunakan sebagai bahan dasar mesin mobil atau mesin listrik (Suryani, 2012).

Metode *Perlin Noise* pada penelitian sebelumnya digunakan sebagai perhitungan untuk mendapatkan sebuah nilai *density* (kedalaman) dari setiap tipe kabut agar menghasilkan simulasi pendekatan terhadap setiap tipe kabut yang lebih nyata. Aplikasi yang digunakan untuk perhitungan matematis pada penelitian tersebut, yaitu MATLAB. Penggunaan MATLAB di khususkan untuk menghasilkan citra sintesi kabut heterogen dua dimensi (2D) yang akan diterapkan pada *Unity 3D* sehingga dapat menghasilkan kabut sintesis heterogen tiga dimensi (3D) (Giantara et al., 2020).

Tabel 2. 1 Densitas 12 Citra Uji

<b>Time</b>	<b>Density</b>
06.01 WIB	3,55
07.01 WIB	3,38
08.01 WIB	2,4
09.01 WIB	3,4
10.01 WIB	4,11
11.01 WIB	3,6
12.01 WIB	2,8
13.01 WIB	4,21
14.01 WIB	3,1
15.01 WIB	8,2
16.01 WIB	5,1
17.01 WIB	4,15

Sumber : (Cahyani, 2020)

Estimasi *Transmission Map* digunakan pada penelitian tersebut karena sangat di perlukan sebagai perkiraan transmisi citra kabut untuk menciptakan *scene* kabut. Untuk menunjukkan tingkatan dari transparansi citra digunakanlah *Transmission map*. Tabel 2.2 merupakan penjelasan perkiraan dari citra kabut yang akan digunakan pada penelitian tersebut.

Tabel 2. 2 Fitur Kabut dalam *Game*

<b>Kabut</b>	<b>Ketebalan</b>
1	Sangat Cerah
2	Sangat Bening
3	Bening
4	Kabut Tipis
5	Kabut Samar
6	Kabut Lembut
7	Kabut Ringan
8	Kabut Sedang
9	Kabut Tebal
10	Kabut Pekat

#### 2.2.4 *Taksonomi Bloom*

Dalam dunia pendidikan terdapat banyak sekali faktor yang mempengaruhi keberhasilan dalam membuat pelajar dapat memahami konteks pembelajaran yang telah di sediakan. Adapun salah satu metode yang membahas tentang upaya dalam membentuk model pembelajaran yang efektif bagi pelajar, yaitu *taksonomi Bloom*. Metode tersebut dapat memberikan pengetahuan, pemahaman, aplikasi, analisis, sintesis dan evaluasi bagi para pelajar. Pada tahap pengembangannya, *taksonomi bloom* mengalami perubahan yang melibatkan penggabungan berbagai elemen, termasuk perubahan istilah untuk setiap tingkat, struktur tingkat yang lebih rumit, dan penambahan jumlah variabel. Perubahan ini telah diimplementasikan dalam desain *e-learning* yang mengandung berbagai perintah secara *online*. *E-learning* ini menggunakan objek dan hasil pembelajaran sebagai dasar dalam pengembangannya. Dalam desain yang diajukan, terdapat tiga aspek pembelajaran yang saling terkait, yaitu aspek kognitif, aspek psikomotorik, dan aspek afektif. (Nugroho et al., 2019). Dalam konteks pendidikan, tujuan Taksonomi Bloom adalah membantu setiap pengajar dalam membuat klasifikasi materi yang harus dipelajari oleh para pelajar dalam periode waktu tertentu. Untuk menjawab tuntutan pendidikan, *taksonomi bloom* mempertimbangkan berbagai perkembangan yang saat ini efektif dan terpercaya digunakan dalam bidang 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. Kognitif pada dasarnya meliputi berbagai hal yang berkaitan dengan kemampuan psikis atau mental dalam menganalisis sebuah pembelajaran (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. Tabel 2.3 merupakan pertanyaan dari aspek kognitif berdasarkan hasil wawancara dari Lutfi Arisatun Niswah sebagai guru SDN 01 Klemunan, Kecamatan Wlingi, Kabupaten Blitar yang menyatakan bahwa tingkatan pertanyaan kognitif dibagi menjadi 3 tingkatan yaitu *low*, *medium*, dan *high*.

Tabel 2. 3 Kuesioner Aspek Kognitif

No.	Kuesioner	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. Dalam ranah ini peserta didik dinilai sejauh mana ia mampu menginternalisasikan nilai – nilai pembelajaran ke dalam dirinya (Nurhidayati & Sunarsih, 2013). Ranah ini erat kaitannya dengan tata nilai dan konsep diri. Tabel 2.4 merupakan pertanyaan dari aspek afektif.

Tabel 2. 4 Kuesioner Aspek Afektif

No.	Kuesioner	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?	A2-3
8	Apa yang kamu rasakan jika kamu kehilangan keluargamu karena bencana erupsi gunung meletus?	A3-1
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 dari panca indra (Nurhidayati & Sunarsih, 2013). Jika pelajar telah dirasa dapat menginternalisasikan, maka tahap selanjutnya adalah bagaimana peserta didik mampu mengaplikasikan pemahamannya dalam kehidupan sehari – hari melalui perbuatan atau tindakan.

Tabel 2.5 merupakan pertanyaan dari aspek psikomotorik.

Tabel 2. 5 Kuesioner Aspek Psikomotorik

No.	Kuesioner	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

### **2.2.5 Tingkat Kesulitan Level Otomatis**

Dalam dunia game terlepas dari tujuan dan pemanfaatannya penerapan konsep tingkat kesulitan level otomatis pada sebuah game sangat banyak dan mudah untuk ditemukan. Konsep tersebut sering disebut sebagai *Auto Dynamic Difficulty*, *Dynamic Game Balancing* ataupun *Dynamic Difficulty Adjustment*. Dalam penerapannya konsep tersebut berfungsi untuk menyesuaikan kemampuan pemain terhadap skenario dan perilaku di dalam *game* yang bekerja secara *realtime* atau dinamis sesuai *skill* dan *ability* pemain. Saat di implementasikan dengan benar konsep tersebut dapat memberikan pemain keseluruhan pengalaman dalam bermain *game*, dimana hal tersebut dapat menjamin pemain mendapat pengalaman baru dan memberikan nuansa bermain *game* yang tidak cenderung jenuh saat *game* terlalu mudah sekaligus menghindari dampak negatif seperti frustrasi saat *game* terlalu sulit untuk diselesaikan (Colwell & Glavin, 2018). Konsep tingkat kesulitan level otomatis pada dasarnya bertujuan untuk membuat pemain tertarik dengan konten *game* yang diberikan sehingga pemain tidak mudah bosan di awal dan akhir permainan.

### **2.2.6 Support Vector Machine (SVM)**

Metode *Support Vector Machine* (SVM) merupakan sebuah metode yang sering digunakan untuk menyelesaikan sebuah kasus pengklasifikasian data. Metode tersebut bekerja dengan cara mengolah sebuah dataset menjadi sebuah data latih yang akan diolah menggunakan model pelatihan yang di tentukan dan data uji untuk menguji kemampuan dari model pelatihan yang di tentukan dalam

mengklasifikasi sebuah data-*i* (Kasim & Sudarsono, 2019). Adapun beberapa model pelatihan SVM yang populer dan sering digunakan antara lain:

### **2.2.7 Sequential Training**

*Sequential Training* adalah salah satu algoritma *Support Vector Machine* yang umum di gunakan pada pembelajaran *semi-supervised*. Algoritma dapat diterapkan untuk sebuah kumpulan data dalam skala yang besar. Selain itu *Sequential Training* juga memberikan keuntungan dalam memproses data dalam jumlah besar yang dimana algoritma ini menggunakan *numerical quadratic programming* sebagai *inner loop* nya untuk mencari nilai  $a$  , model pelatihan ini juga sering di gunakan dalam situasi pembelajaran di mana jumlah *dataset* pelatihan terus meningkat (Huang & Sun, 2012).

### **2.2.8 Sequential Minimum Optimization (SMO)**

*Sequential Minimum Optimization* (SMO) salah satu algoritma dalam model pelatihan *Support Vector Machine* yang cukup sederhana dalam proses pelatihannya dan dapat memberikan solusi pada permasalahan yang terjadi di *Quadratic Problem* (QP). Dalam hal ini SMO bekerja dengan cara yang berbeda di karenakan SMO menggunakan *analytic quadratic programming* sebagai *inner loop* nya. Maka dengan itu SMO akan menggunakan dua buah data untuk setiap literasinya agar mendapat pencarian solusi yang optimal (Willy Sutina, 2010).

Namun fokus pada penelitian ini adalah penerapan *Support Vector Machine* menggunakan model pelatihan *Sequential Minimum Optimization*. Berikut adalah formula matematis yang di gunakan untuk perhitungan:

## 1. Normalisasi

Untuk menyederhanakan sebuah data di perlukanlah langkah normalisasi. *Min-Max Normalization* merupakan sebuah metode normalisasi yang sering di gunakan untuk mengatasi permasalahan nilai antar fitur yang memiliki jarak yang terlalu jauh (Larose, 2005).

$$X^* = \frac{X - \min(x)}{\max(x) - \min(x)} \quad (2.1)$$

Keterangan:

$X$  = data yang belum di normalisasi.

$\min(x)$  = nilai *minimum* dari semua data.

$\max(x)$  = nilai *maximum* dari semua data.

## 2. *Kernel Trick*

Dalam penerapannya *kernel* digunakan untuk membuat sebuah *matrix* yang terbentuk dari pemetaan rentetan fitur lama ke dalam rentetan fitur baru. Pemanfaatan fungsi kernel pada *Support Vector Machines* (SVM) juga dikenal sebagai *kernel trick*. Di bawah ini merupakan beberapa pilihan dalam menentukan fungsi *kernel* yang Akan di gunakan:

### a. *Kernel linier*

$$K(x, y) = x \cdot y \quad (2.2)$$

Keterangan:

$x$  = data ke- $x$

$y$  = data ke- $y$

b. *Kernel Polynomial*

$$K(x, y) = (x \cdot y + C)^d \quad (2.3)$$

Keterangan:

$x$  = data ke- $x$

$y$  = data ke- $y$

$C$  = *Complexity*

c. *Kernel Radial Basis Function (RBF)*

$$K(x, y) = \prod_{i=1}^n \exp\left(-\frac{\|x - y\|}{2\sigma^2}\right) \quad (2.4)$$

Keterangan:

$x$  = data ke- $x$

$Y$  = data ke- $y$

$\sigma$  = nilai variansi

3. *Sequential Minimum Optimization (SMO)*

Pada langkah ini SMO akan mengoptimasi setiap tahapan terkecil dengan memilih dua buah *Langrange Multiplier* ( $a_i$ ) yang akan di optimasi secara bersamaan dan mempebaruinya pada SVM.

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

$a_i$  = *Langrange Multiplier*.

$C$  = *complexity*, bertujuan untuk meminimalkan nilai *error*

$\varepsilon$  = *epsilon*, digunakan sebagai sebuah kondisi.

- b. Mencari  $E_i$

Mencari nilai *Error* pada data latih ke-i.

$$Error_i = \left( \sum_{i=1}^n a_i y_i K(x_i x_j) \right) + b - y_i \quad (2.5)$$

Keterangan:

$a_i$  = *Langrange Multiplier*

$y_i$  = kelas data ke-i

$b$  = nilai bias

- c. Setelah mendapatkan nilai  $Error_i$  langkah selanjutnya adalah pengecekan *Kash-Kuhn-Tucker* (KKT) dengan syarat  $E_i < -\epsilon$  dan  $a_i < c$  atau  $E_i > \epsilon$  dan  $a_i > 0$  jika memenuhi akan *true* dan jika tidak maka akan *false* dan melanjutkan ke data-i selanjutnya.
- d. Langkah selanjutnya mencari nilai *Langrange Multiplier* ke-2 ( $a_j$ )/kelas data latih dengan cara random.
- e. Mencari Nilai  $Error_j$  dengan persamaan 2.1
- f. Mencari nilai  $a_j^{lama} = a_j^{bssru.potongan}$  dan  $a_i^{lama} = a_i^{bssru.potongan}$
- g. Kemudian mencari nilai L dan H sebagai batas *maximum* dan *minimum* dengan persamaan:

$$L = \begin{cases} \max(0, a_j - a_i) & \text{jika } y_i \neq y_j \\ \max(0, a_i + a_j - C) & \text{jika } y_i = y_j \end{cases} \quad (2.6)$$

$$H = \begin{cases} \min(C, C + a_j - a_i) & \text{jika } y_i \neq y_j \\ \min(C, a_i + a_j) & \text{jika } y_i = y_j \end{cases} \quad (2.7)$$

- h. Melakukan pengecekan terhadap kondisi L dan H dengan menggunakan syarat L=H.

- i. Setelah pengecekan kondisi L dan H selanjutnya adalah menghitung nilai  $\eta$  (*eta*) menggunakan persamaan 2.14.

$$\eta = 2K(x^i, x^j) - (x^i, x^i) - (x^j, x^j) \quad (2.8)$$

- j. Melakukan pengecekan kondisi  $\eta$  dimana  $\eta \geq 0$ .
- k. Mencari nilai  $a_j$  baru dengan menggunakan persamaan 2.15.

$$a_j^{baru} = a_j^{lama} - \frac{y_i(E_i - E_j)}{\eta} \quad (2.9)$$

- l. Selanjutnya mencari nilai  $a_j^{baru}$  baru yang berpotongan antara L dan H dengan menggunakan persamaan:

$$a_j^{baru.potongan} = \begin{cases} H, & \text{jika } a_j^{baru} > H \\ a_j^{baru}, & \text{jika } L < a_j^{baru} < H \\ L, & \text{jika } a_j^{baru} < L \end{cases} \quad (2.10)$$

- m. Mencari nilai  $a_i$  baru yang akan di gunakan pada data *kernel* ke-i selanjutnya menggunakan persamaan:

$$a_i^{baru} = a_i^{lama} + y_i * y_j (a_j^{lama} - a_j^{baru.potongan}) \quad (2.11)$$

- n. Setelah nilai  $a_i^{baru}$  di temukan langkah selanjutnya mencari nilai bias1 dan bias2 untuk menentukan nilai bias baru yang akan di gunakan pada data *kernel* ke-i selanjutnya menggunakan persamaan 2.18.

$$b_1 = b^{lama} - E_i - y_i(a_i^{baru} - a_i^{lama})K(x^i, x^i) - y_i(a_j^{baru.potongan} - a_j^{lama})K(x^i, x^i) \quad (2.12)$$

$$b_2 = b^{lama} - E_i - y_i(a_i^{baru} - a_i^{lama})K(x^i, x^j) - y_i(a_j^{baru.potongan} - a_j^{lama})K(x^j, x^j) \quad (2.13)$$

#### 4. Mencari nilai bias

Rumus untuk mencari nilai bias sebagai berikut:

$$b = \begin{cases} b_1, & \text{jika } 0 < a_i < C \\ b_2, & \text{jika } 0 < a_j < C \\ \frac{(b_1 + b_2)}{2}, & \text{Lainnya} \end{cases} \quad (2.14)$$

### 5. *Testing SVM*

Rumus yang digunakan untuk menghitung *testing SVM* adalah sebagai berikut:

$$f(x) = \sum_{i=1}^m a_i y_i k(x_i, x) + b \quad (2.15)$$

Keterangan:

$a_i$  = alfa ke-i

$y_i$  = kelas data latih ke-i

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

$b$  = nilai bias

## BAB III

### DESAIN DAN RANCANGAN SISTEM

#### 3.1 Skenario Penelitian



Gambar 3. 1 Blok Diagram Skenario Penelitian

Dari *blok diagram* pada gambar 3.1 penelitian memiliki alur sebagai berikut:

1. Studi ini diawali dengan pengumpulan data yang akan digunakan sebagai *dataset* yang nanti akan dibagi menjadi data latih dan data uji.
2. Langkah selanjutnya pada studi ini, yaitu merancang desain sistem *game*. Desain sistem *game* disusun dalam bentuk alur menu dan *gameplay flowchart*.
3. Setelah alur *game* tergambar, tahap selanjutnya adalah menyusun jalan cerita pada *game*. Penyusunan jalan cerita atau plot dalam *game* ini dibentuk dalam model *storyboard game*.
4. Pada tahap ini *game* sudah bisa dimainkan. Namun, perubahan *level* hanya mengacu pada *skor*, sehingga dengan banyaknya skor yang diperoleh oleh pemain maka *level* akan semakin meningkat.

5. Langkah selanjutnya adalah merancang pengatur level kesulitan dengan mengimplementasikan *support vector machine*.
6. Langkah terakhir adalah melakukan pengujian terhadap hasil level dari pengatur kesulitan tanpa *support vector machine* dan pengatur kesulitan menggunakan *support vector machine* dengan metode *Random Forest* dan model pelatihan *Sequential Training* menggunakan *confusion matrix* untuk menghasilkan nilai akurasi.

### **3.2 Perancangan Desain Sistem Game**

Proses pembuatan *game* pada penelitian ini melibatkan beberapa desain awal, yaitu:

#### **3.2.1 Desain Menu *Flow***

Untuk memudahkan pembuatan *User Interface* pada penelitian ini, game akan di rancang dalam bentuk menu flow. Desain menu flow yang di tawarkan dalam studi ini dijelaskan sebagai berikut:

##### **1. *Title Screen***

Dalam panel ini akan ada beberapa *button* yang umum di gunakan pada berbagai *game*, yaitu:

- a. *Button play with SVM*, berfungsi untuk menghubungkan *game* dengan pengatur *level* kesulitan menggunakan SVM.
- b. *Button play without SVM*, berfungsi untuk menghubungkan *game* dengan pengatur *level* kesulitan tanpa menggunakan SVM.

c. *Button setting*, untuk mengatur tampilan.

d. *Button quit*, untuk keluar dari *game*.

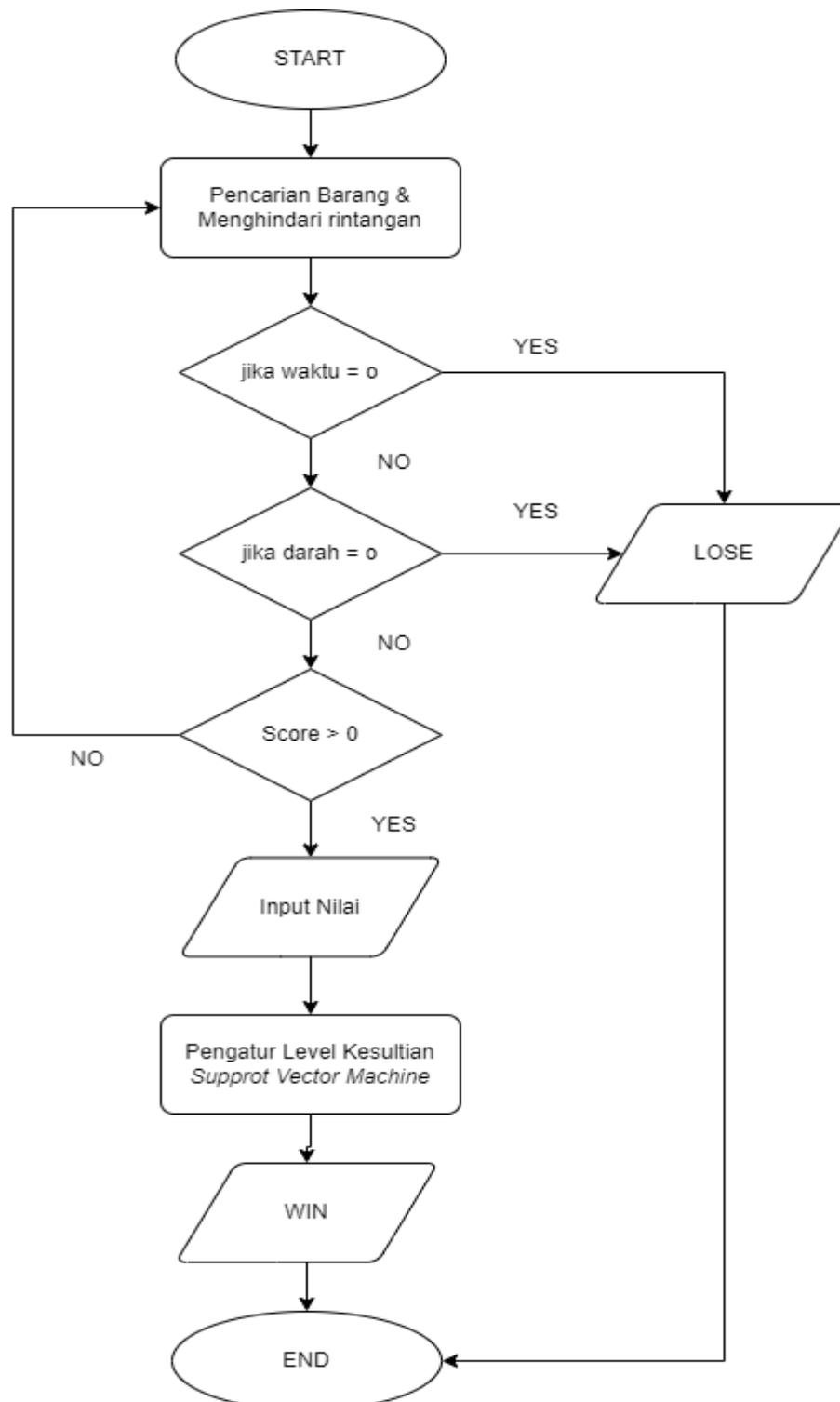
## 2. *Game Playing*

Desain *game* yang di buat pada penelitian ini menggunakan media berupa *terrain* yang digunakan agar *main character* leluasa dalam menjalankan *game* petualangan, yang di dalamnya sudah terdapat beberapa asset *game* berupa barang dan rintangan.

## 3. *Button Game Pause*

*Button* ini berfungsi untuk menghentikan *game* sementara sesuai keinginan *player*.

### 3.2.2 *Gameplay Flowchart*



Gambar 3. 2 *Flowchart Gameplay*

Berdasarkan pada gambar 3.2 maka alur *gameplay* pada penelitian sebagai berikut

1. Mulai.
2. Pemain wajib mengambil barang sebanyak mungkin dan menghindari rintangan untuk mendapatkan poin tambahan.
3. Dalam *game* ini akan di ambil tujuh parameter yang akan di jadikan sebagai nilai *input*.
4. Mengumpulkan barang dan menghindari rintangan akan menjadi nilai input untuk variabel Jumlah barang, nilai barang, jumlah musuh, dan nilai musuh.
5. *Skor*, waktu, dan darah berfungsi sebagai kondisi dimana *player* dapat menyelesaikan *game* dengan baik atau tidak. Jika tidak maka akan *lose* , dan jika berhasil *survive* maka akan *win* dan melanjut ke *level* sesuai poin yang di capai.
6. Nilai-nilai yang di dapatkan dari langkah 4 dan 5 nantinya akan menjadi inputan baru yang akan di proses menggunakan SVM untuk menghasilkan pengatur *level* kesulitan.
7. Jika semua kondisi terpenuhi maka pengatur *level* kesulitan hasil dari SVM akan bekerja secara otomatis dan dinamis.
8. Selesai.

### **3.3 Perancangan *Storyboard***

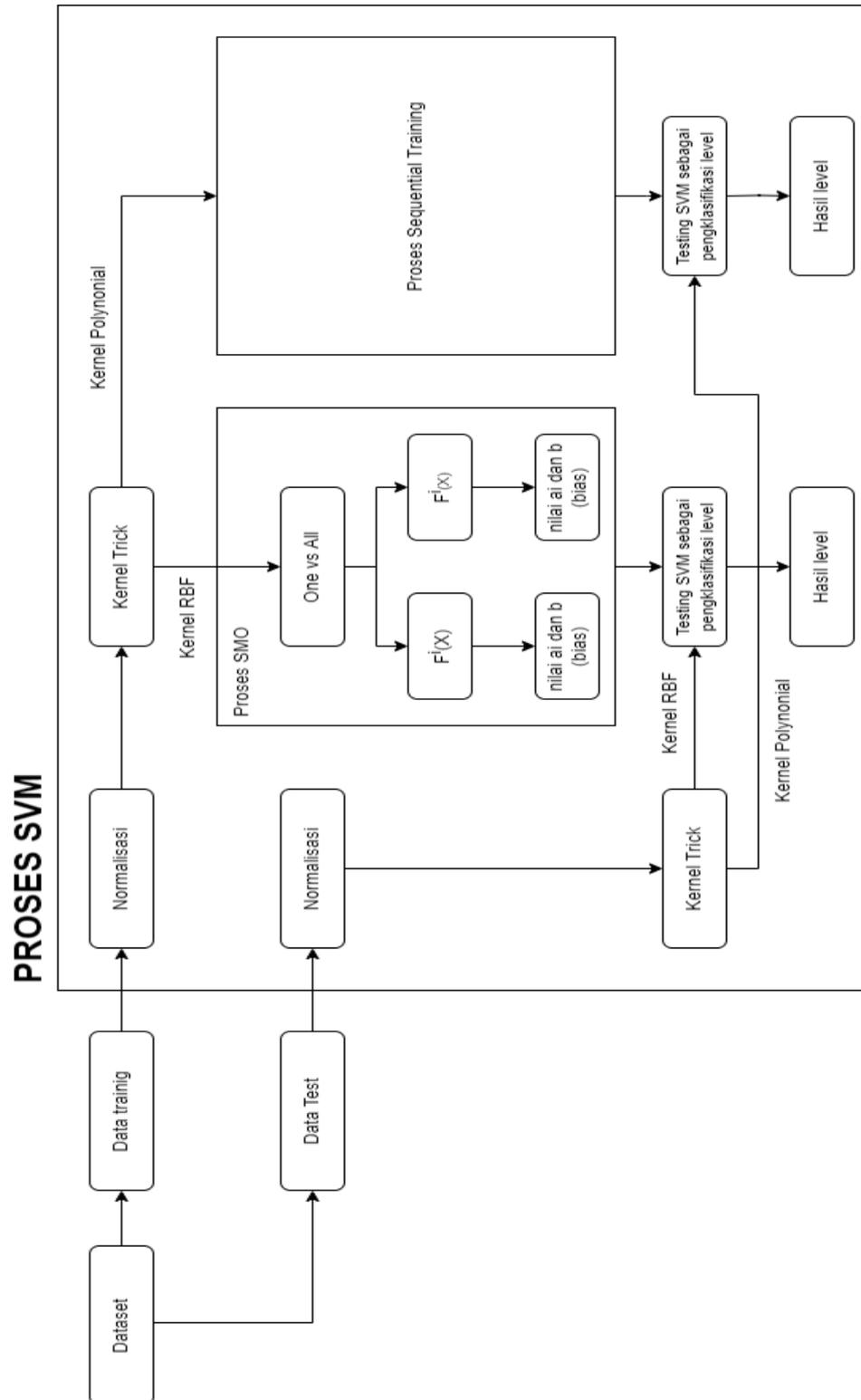
*Game* ini menceritakan pendaki yang akan bersiap-siap untuk mendaki gunung kelud, pendaki tersebut perlu mengumpulkan barang-barang yang akan digunakan untuk mendaki di hutan. Di dalam hutan pendaki akan berhadapan dengan hewan-hewan berbahaya. Apabila menyentuh hewan tersebut, pendaki akan

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.

### 3.4 Perancangan Implementasi *Support Vector Machine* (SVM)

*Support vector machine* dirancang sebagai algoritma yang akan di implementasikan pada pengatur kesulitan, supaya pengatur kesulitan lebih dinamis di banding tanpa SVM. Rancangan Implementasi SVM dalam penelitian ini terdapat pada gambar 3.3 dengan penjelasan sebagai berikut:

1. *Dataset* yang akan digunakan pada penelitian ini di ambil dari penelitian sebelumnya yang menggunakan *Random Forest* sebagai pengatur *level* kesulitan. Dimana *dataset* yang di gunakan akan di bagi menjadi dua, yaitu data latih dan dat uji.
2. Selanjutnya setiap fitur utama yang di jadikan data latih akan di normalisasi agar mendapatkan nilai yang lebih sederhana.
3. Langkah berikutnya yaitu melakukan normalisasi yang hasilnya nanti akan di gunakan sebagai perhitungan *kernel*.
4. *Kernel Trick* melakukan pemilihan *kernel* untuk perhitungan pada model *training* SVM. Yang dimana pada penelitian ini akan menggunakan *kernel Radial Basis Function* (RBF).
5. Model *Training* yang akan digunakan pada penelitian adalah model *training* menggunakan *Sequential Minimum Optimization* untuk mengklasifikasi *level*, dan hasil klasifikasi dari model *Sequrntial Training* di gunakan untuk perbandingan pada *confusion matrix*.



Gambar 3. 3 Blok Diagram Rancangan Implementasi SVM

### 3.5 Implementasi *Support Vector Machine* (SVM)

Pada implementasi algoritma *support vector machine* dibutuhkan pelatihan data latih terlebih dahulu dimana data latih yang akan di gunakan pada implementasi ini adalah hasil dari level dari penelitian sebelumnya yang menggunakan metode *Random Forest*.

#### 1. *Dataset*

Tabel 3. 1 Data Latih

No.	Masukan							Level
	Skor	Waktu	Darah	Tipe Musuh	Jumlah Musuh	Jumlah Barang	Nilai Barang	
1	5	400	100	1	2	5	2	1
10	15	300	90	1	4	10	2	2
20	25	280	80	2	5	15	3	3
30	30	260	70	2	6	30	3	4
50	45	240	60	3	6	40	3	5
60	50	220	50	3	7	45	4	6
70	65	200	40	4	7	50	4	7
80	75	180	30	4	8	60	5	8
90	80	160	20	5	9	60	5	9
100	99	140	15	5	10	65	5	10

#### 2. Normalisasi

Selanjutnya adalah melakukan normalisasi pada setiap nilai yang digunakan sebagai data latih agar mendapatkan nilai yang lebih sederhana. Data yang di gunakan ialah data pada tabel 3.1 dan rumus yang digunakan terdapat pada rumus (3.1).

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

Selanjutnya dilakukan perhitungan normalisasi data berdasarkan rumus 3.1 yang akan menghasilkan nilai pada tabel 3.2.

Tabel 3. 2 Hasil Perhitungan Normalisasi Data

Skor	Waktu	Darah	Tipe Musuh	Jumlah Musuh	Jumlah Barang	Nilai Barang
0.21	1.00	0.70	0.22	0.25	0.29	0.75
0.84	0.33	1.00	0.44	0.25	0.59	1.00
0.58	0.75	0.15	0.44	0.50	0.18	1.00
1.00	0.33	0.60	0.56	1.00	0.59	0.75
0.68	0.33	1.00	0.56	0.25	0.29	1.00
0.26	0.33	0.40	0.56	1.00	0.69	0.75
1.00	0.33	0.35	0.56	0.75	0.59	1.00
0.89	0.50	0.50	0.67	0.00	0.08	1.00
0.37	1.00	0.35	0.67	0.25	0.59	0.75
0.58	0.63	0.35	0.56	0.25	0.08	0.75

### 3. Perhitungan *Kernel*

Setelah normalisasi data di dapatkan langkah selanjutnya ialah menghitung nilai *kernel*. *Kernel* berfungsi untuk memetakan fitur yang lama pada fitur yang baru. Berikut formula yang di gunakan untuk menghitung nilai *kernel*.

$$K(x, y) = \prod_{i=1}^n \exp\left(-\frac{\|x - y\|}{2\sigma^2}\right) \quad (3.2)$$

Sebelum menghitung nilai *kernel* terlebih dahulu untuk mencari nilai variansi dari setiap variable menggunakan rumus 3.3.

$$\sigma^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n} \quad (3.3)$$

$$\sigma = \frac{(0.19 - 0.64)^2 + (0.04 - 0.64)^2 + (0 - 0.64)^2 + (0.13 - 0.64)^2 + (0 - 0.64)^2 + (0.14 - 0.64)^2 + (0.13 - 0.64)^2 + (0.06 - 0.64)^2 + (0.07 - 0.64)^2 + (0 - 0.64)^2}{10} = 0.77$$

Dari perhitungan manual di atas maka di dapatkan nilai variansi dari setiap variable seperti pada table 3.3.

Tabel 3. 3 Nilai Variansi dari Setiap Fitur

Skor	Waktu	Darah	Tipe Musuh	Jumlah Musuh	Jumlah Barang	Nilai Barang
0.077	0.071	0.073	0.015	0.110	0.051	0.016

Langkah selanjutnya melakukan perhitungan *kernel*, di bawah ini merupakan proses perhitungan *kernel* menggunakan rumus 3.2.

$$\begin{aligned}
 K(1,1) = & \left( \exp \left( -\frac{(0.21 - 0.21)}{2 * 0.077} \right) \right) * \left( \exp \left( -\frac{(1 - 0.21)}{2 * 0.071} \right) \right) * \left( \exp \left( -\frac{(0.7 - 0.21)}{2 * 0.073} \right) \right) \\
 & * \left( \exp \left( -\frac{(0.22 - 0.21)}{2 * 0.015} \right) \right) * \left( \exp \left( -\frac{(0.25 - 0.21)}{2 * 0.11} \right) \right) \\
 & * \left( \exp \left( -\frac{(0.29 - 0.21)}{2 * 0.051} \right) \right) * \left( \exp \left( -\frac{(0.75 - 0.21)}{2 * 0.016} \right) \right) \\
 K(1,1) = & 1
 \end{aligned}$$

Pada proses perhitungan diatas merupakan proses perhitungan  $k(1,1)$  artinya hasil dari perhitungan antara kolom 1 dengan baris 1 Perhitungan tersebut dilanjutkan sampai perhitungan antara maximal kolom yaitu  $k(10,10)$  sehingga terbentuk sebuah matrik seperti pada tabel 3.4.

Tabel 3. 4 Hasil Perhitungan *Kernel*

<b>x,y</b>	1	2	3	4	5	6	7	8	9	10
1	1.00	1.08	1.05	1.16	1.06	1.12	1.13	1.07	1.02	1.03
2	1.08	1.00	1.09	1.08	1.01	1.12	1.06	1.04	1.09	1.06
3	1.05	1.09	1.00	1.08	1.08	1.07	1.05	1.05	1.03	1.01
4	1.16	1.08	1.08	1.00	1.09	1.05	1.01	1.14	1.14	1.10
5	1.06	1.01	1.08	1.09	1.00	1.12	1.07	1.03	1.08	1.04
6	1.12	1.12	1.07	1.05	1.12	1.00	1.05	1.18	1.10	1.10
7	1.13	1.06	1.05	1.01	1.07	1.05	1.00	1.08	1.10	1.06
8	1.07	1.04	1.05	1.14	1.03	1.18	1.08	1.00	1.06	1.02
9	1.02	1.09	1.03	1.14	1.08	1.10	1.10	1.06	1.00	1.03
10	1.03	1.06	1.01	1.10	1.04	1.10	1.06	1.02	1.03	1.00

#### 4. *Sequential Minimum Optimization (SMO)*

Pada tahap selanjutnya yaitu proses perhitungan *SMO* yaitu proses *training* dari data latih yang ada. Terdapat 2 tahapan pada proses perhitungan menggunakan *SMO*, Dimana *SMO* akan menggunakan algoritma *One vs All*

untuk mendapatkan 2 fungsi yang akan di gunakan pada *testing SVM*. Tabel 3.4 merupakan gambaran konsep dari algoritma *One vs All* dalam proses perhitungan SMO.

Tabel 3. 5 Desain *One vs All* Pada SMO

	Tipe Kelas									
	+1	-1								
KELAS / LEVEL	1	2	3	4	5	6	7	8	9	10
	2	1	3	4	5	6	7	8	9	10
	3	1	2	4	5	6	7	8	9	10
	4	1	2	3	5	6	7	8	9	10
	5	1	2	3	4	6	7	8	9	10
	6	1	2	3	4	5	7	8	9	10
	7	1	2	3	4	5	6	8	9	10
	8	1	2	3	4	5	6	7	9	10
	9	1	2	3	4	5	6	7	8	10

Tahapan pertama membandingkan data kelas 1 dengan data kelas lainnya. Dimana data kelas 1 di anggap sebagai tipe data kelas (+1) dan data kelas yang lainnya di anggap sebagai tipe data kelas (-1). Pertama SMO akan membuat nilai awal  $a_i = 0$  dan nilai bias  $b = 0$ . Kemudian mencari nilai  $E_i(Error_i)$  menggunakan persamaan 3.4.

$$Error_i = \left( \left( \sum_{i=1}^n a_i y_i K(x_i^i, x_i^j) + b \right) - y_i \right) \quad (3.4)$$

$$Error_i = \left( ((0 * 1 * 1) + 0) + ((-10 * (-1) * 1) + 0) \right) \\ + \left( (-18.3 * (-1) * 1) + ((136 * -1) * 1) + 0 \right)$$

Setelah di ketahui nilai dari  $Error_i$  maka langkah selanjutnya adalah pengecekan kondisi *Karush-Kahn-Tucker* (KKT). Persyaratan yang di gunakan

untuk melakukan pengecekan KKT , yaitu  $E_i < -e(\textit{epsilon})$  dan  $a_i > C$  atau  $E_i > \textit{epsilon}$  dan  $a_i > 0$ , Dimana  $e = 0.01$  dan  $C = 10$ . Jika memenuhi akan *true* dan melanjutkan proses pencarian nilai  $a_i$  yang baru, jika tidak memenuhi akan *false* dan melanjutkan ke data- $i$  selanjutnya, yaitu menentukan nilai penggali kedua ( $a_j$ ) secara random (acak 1 sampai 10).

Setelah di dapatkan nilai  $j$  maka langkah selanjutnya mencari, yaitu nilai  $E_j$  menggunakan persamaan 3.5 dimana  $i = j$ . Berikut adalah hasil perhitungan untuk pencarian nilai  $E_j$ .

$$\textit{Error}_j = \left( \left( \sum_{i=1}^n a_i y_i K(x_i^i, x_i^j) + b \right) - y_i \right) \quad (3.5)$$

$$\begin{aligned} \textit{Error}_j = & \left( (0 * 1 * 1) + 0 \right) + \left( (-10 * (-1) * 1) + 0 \right) \\ & + \left( (-18.3 * (-1) * 1) + ((136 * -1) * 1) + 0 \right) \end{aligned}$$

Setelah mendapat nilai  $E_j$  maka selanjutnya adalah memperbarui nilai  $a_j^{\textit{lama}} = a_j^{\textit{baru.potongan}}$  dan  $a_i^{\textit{lama}} = a_i^{\textit{baru}}$ . Kemudian mencari nilai L dan H menggunakan persamaan.

$$L = \begin{cases} \textit{max}(0, a_j - a_i) & \textit{jika } y_i \neq y_j \\ \textit{max}(0, a_i + a_j - C) & \textit{jika } y_i = y_j \end{cases} \quad (3.6)$$

$$H = \begin{cases} \textit{min}(C, C + a_j - a_i) & \textit{jika } y_i \neq y_j \\ \textit{min}(C, a_i + a_j) & \textit{jika } y_i = y_j \end{cases} \quad (3.7)$$

Maka di dapatkan nilai L dan H sebagai berikut:

$$L(1, 1) = \textit{max}(0, 0 - 0) = 0$$

$$H(1,1) = \textit{min}(10, 10 + 0 - 0) = 10$$

Setelah nilai L dan H di ketahui maka selanjutnya adalah mencari nilai  $n(eta)$  menggunakan persamaan 3.8.

$$\eta = 2K(x^i, x^j) - (x^i, x^i) - (x^j, x^j) \quad (3.8)$$

Dengan menggunakan persamaan 3.8 maka di lakukan proses perhitungan manual untuk mencari nilai  $\eta(eta)$  sebagai berikut:

$$\begin{aligned} \eta &= 2 * 1.18 - 1 - 1 \\ \eta &= 0.356 \end{aligned}$$

Setelah di dapatkan nilai  $\eta$  maka selanjutnya adalah melakukan pengecekan terhadap kondisi  $\eta$ , dengan persyaratan  $\eta \geq 0$  jika memenuhi akan *true* dan melanjutkan ke data-*i* selanjutnya, jika tidak memenuhi maka akan *false* dan melanjutkan pencarian nilai dari  $a_j$  baru. Untuk mendapatkan nilai dari  $a_j$  yang baru di gunakanlah persamaan 3.9.

$$\begin{aligned} a_j^{baru} &= a_j^{lama} - \frac{y_i(E_i - E_j)}{\eta} \quad (3.9) \\ a_5^{baru} &= 0 - \frac{1 * ((-1) - 1)}{-0.038} \\ &= 52.48 \end{aligned}$$

Setelah mendapatkan nilai  $a_j$  baru maka langkah selanjutnya adalah mencari  $a_j$  baru yang berpotongan antara L hingga H menggunakan persamaan 3.10.

$$a_j^{baru.potongan} = \begin{cases} H, & \text{jika } a_j^{baru} > H \\ a_j^{baru}, & \text{jika } L < a_j^{baru} < H \\ L, & \text{jika } a_j^{baru} < L \end{cases} \quad (3.10)$$

$$a_5^{baru.potongan} = \begin{cases} 10, & \text{jika } 52.48 > 10 \\ 52.48, & \text{jika } 0 < 52.48 < 10 \\ 0, & \text{jika } 52.48 < 0 \end{cases}$$

Setelah diketahui nilai dari  $a_j^{baru.potongan}$  langkah selanjutnya adalah mencari nilai  $a_i^{baru.potongan}$  menggunakan persamaan 3.11.

$$a_4^{baru} = a_i^{lama} + y_i * y_j (a_j^{lama} - a_j^{baru.potongan}) \quad (3.11)$$

$$a_4^{baru} = 0 + 1 * (-1) * (52.48 - 10)$$

$$a_4^{baru} = 52.48$$

Langkah selanjutnya adalah mencari nilai  $b$  (*bias*). Untuk mendapatkan nilai  $b$  maka perlu mencari nilai dari  $B1$  dan  $B2$ . Persamaan (3.12) dan (3.13) berikut digunakan untuk mencari nilai dari  $B1$  dan  $B2$ .

$$b_1 = b^{lama} - E_i - y_i (a_i^{baru} - a_i^{lama}) * K(x^i, x^i) - y_i (a_j^{baru.potongan} - a_j^{lama}) K(x^i, x^i) \quad (3.12)$$

$$b_1 = 0 - 1 - 1 * (52.48 - 0) * 1.13 - 1 * (10 - 0) * 1.3$$

$$b_1 = 588$$

$$b_2 = b^{lama} - E_i - y_i (a_i^{baru} - a_i^{lama}) K(x^i, x^j) - y_i (a_j^{baru.potongan} - a_j^{lama}) K(x^j, x^j) \quad (3.13)$$

$$b_2 = 0 - 1 - 1 * (52.48 - 0) * 1.3 - 1 * (10 - 0) * 1$$

$$b_2 = 336$$

Setelah  $B1$  dan  $B2$  di dapatkan maka selanjutnya adalah mencari nilai  $b$  menggunakan persamaan.

$$b = \begin{cases} b_1, & \text{jika } 0 < a_i < C \\ b_2, & \text{jika } 0 < a_j < C \\ \frac{(b_1 + b_2)}{2}, & \text{Lainnya} \end{cases} \quad (3.14)$$

$$b = \begin{cases} 588, & \text{jika } 0 < 52.48 < 10 \\ 336, & \text{jika } 0 < 52.48 < 10 \\ \frac{(588 + 336)}{2} = 461.8 \end{cases}$$

Langkah pada proses optimasi menggunakan SMO di atas di lakukan pada setiap *kernel*. Untuk hasil akhir dari nilai  $a_i$  yang memenuhi kondisi KKT dengan syarat  $a_i < C$  atau  $a_i > 0$ . Tabel 3.6 dan 3.7 adalah hasil akhir nilai  $a_i$  tahap pertama dan tahap kedua, sedangkan untuk nilai bias tahap pertama adalah -667.35 dan tahap kedua -552.17.

Tabel 3. 6 Hasil Akhir Nilai  $a_i$  Tahap Pertama

	$a_i$ & $a_j$
$a_1$	93
$a_2$	80
$a_3$	54
$a_4$	74
$a_5$	99
$a_6$	89
$a_7$	65
$a_8$	94
$a_9$	87
$a_{10}$	64

Tabel 3. 7 Hasil Akhir Nilai  $a_i$  Tahap Kedua

	$a_i$ & $a_j$
$a_1$	54
$a_2$	85
$a_3$	75
$a_4$	65
$a_5$	33
$a_6$	89
$a_7$	90
$a_8$	44
$a_9$	12
$a_{10}$	50

Tabel 3. 8 Data Uji

Skor	Waktu	Darah	Tipe Musuh	Jumlah Musuh	Jumlah Barang	Nilai Barang
15	192	96	7	16	1	1

Tabel 3. 9 Kernel Uji

$K(x_i, x_j)$	$x_1, x_1$	$x_1, x_2$	$x_1, x_3$	$x_1, x_4$	$x_1, x_5$	$x_1, x_6$	$x_1, x_7$	$x_1, x_8$	$x_1, x_9$	$x_1, x_{10}$
$x_1$	1.02	13.1	5.35	4.89	3.64	1.00	1.03	1.71	1.99	2.53

Data uji di ambil dari percobaan data latih, data uji digunakan sebagai pengujian, sedangkan data variansi berasal dari proses *training* sebelumnya. *Kernel* adalah data baru yang dihasilkan dari pengolahan data uji. Untuk mencari *kernel*, langkah-langkah berikut dapat dilakukan: data uji dikurangi dengan target (data *training*), kemudian hasilnya dipangkatkan dua dan dibagi -2, lalu dikalikan dengan nilai variansi sebelumnya untuk setiap fitur. Hasil pencarian *kernel* dijumlahkan sebanyak jumlah fitur. Setelah mendapatkan data *kernel* dari data uji dan nilai alfa serta bias dari dua tahapan sebelumnya, langkah selanjutnya adalah membuat fungsi dari persamaan pada setiap tahapan.

### 5. Testing SVM

$$f(x_i) = \text{sign} \left( \sum_{i=1}^n a_i y_i K(x_i, x_j) + b \right) \quad (3.15)$$

$$\begin{aligned} f(x_1) &= \text{sign} ((93 * 1 * 1.02) + (80 *) + (54 *) + (74 *) + (99 *) + (89 *) \\ &\quad + (65 *) + (34 *) + (87 *) + (64 *) + (-667.35)) \\ &= \text{sign}(-371.45) = -1 \end{aligned}$$

$$\begin{aligned} f(x_2) &= \text{sign} ((93 * 1 * 1.02) + (80 *) + (54 *) + (74 *) + (99 *) + (89 *) \\ &\quad + (65 *) + (34 *) + (87 *) + (64 *) + (-667.35)) \\ &= \text{sign}(-552.67) = 1 \end{aligned}$$

Formula pada *testing* SVM ini di gunakan saat data uji telah di tentukan dengan mengamil nilai  $a_i$  dan bias dari setiap kombinasi tahapan. Proses ini berguna untuk mengklasifikasi data uji termasuk dalam kelas  $f(x_1)$  atau  $f(x_2)$  kelas 2. Dan berdasarkan perhitungan *testing* SVM menggunakan persamaan (3.15) maka data uji termasuk dalam kelas 2.

### 3.6 Rancangan Pengujian

Pada tahap akhir penelitian ini, dilakukan proses untuk mendapatkan *value accury, presisi, recall, dan F-measure*. Data yang digunakan dibagi menjadi 2 jenis, yaitu data latih dan data uji. Data data latih digunakan untuk melatih model pelatihan yang telah dibangun, sedangkan data *testing* digunakan untuk menguji model pelatihan yang telah dilatih. Untuk menghitung nilai *True Positive, True Negative, False Positive, dan False Negative* dari hasil uji data *testing*, digunakan *confusion matrix*. Kemudian nilai-nilai tersebut dihitung dengan menggunakan formula yang telah ditentukan.

$$\frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (3.16)$$

$$\frac{TP}{TP+FP} \times 100\% \quad (3.17)$$

$$\frac{TP}{TP+FN} \times 100\% \quad (3.18)$$

$$\frac{precision \times recall}{precision+recall} \times 100\% \quad (3.19)$$

Formula 3.16 – 3.19 adalah proses perhitungnan untuk mendapatkan nilai *accury, presisi, recall, dan F-measure*.

Untuk skenario pengujian pada studi ini akan menggunakan skenario pengujian 80 : 20 dimana *dataset* akan di bagi menjadi sebuah *raw data* yang acak dengan persentase 80% sebagai data *training* dan 20% sebagai data *testing*.

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan hasil dan pembahasan dari sistem pengaturan *level* dinamis menggunakan metode *support vector machine*, lalu hasil dari metode *support vector machine* akan dihitung menggunakan *confusion matrix* untuk mendapatkan *accuracy*, presisi, *recall*, dan *F-measure*.

#### 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.

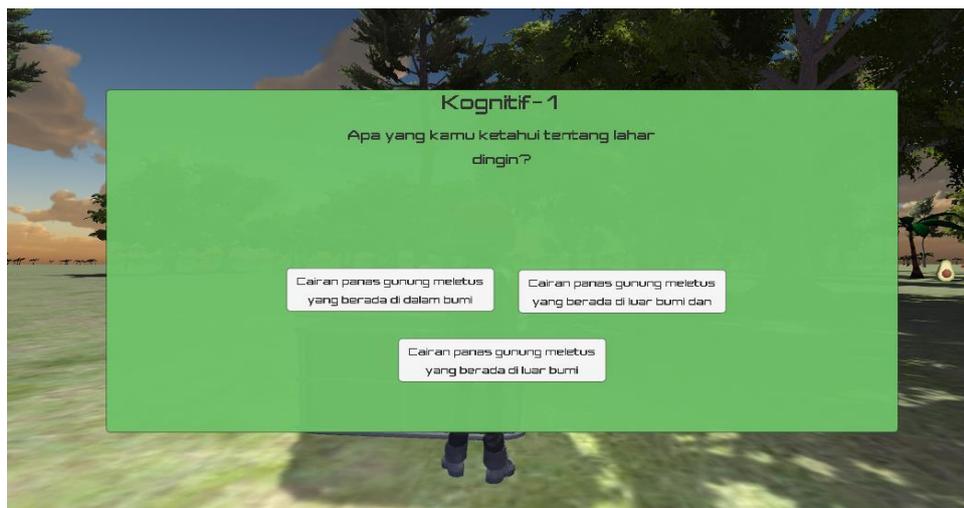
### 4.1.2 Tampilan *Gameplay*



Gambar 4. 2 Tampilan *Gameplay*

Pada saat memasuki *game* dengan tujuan untuk mencari barang-barang yang ada di dalam *game*. Pada tahap ini peneliti menggunakan *controller* agar *player* bias leluasa dalam menjalankan *game*. Di perhatikan dalam *game* terdapat kondisi yang harus terpenuhi agar *level* kabut bisa berubah.

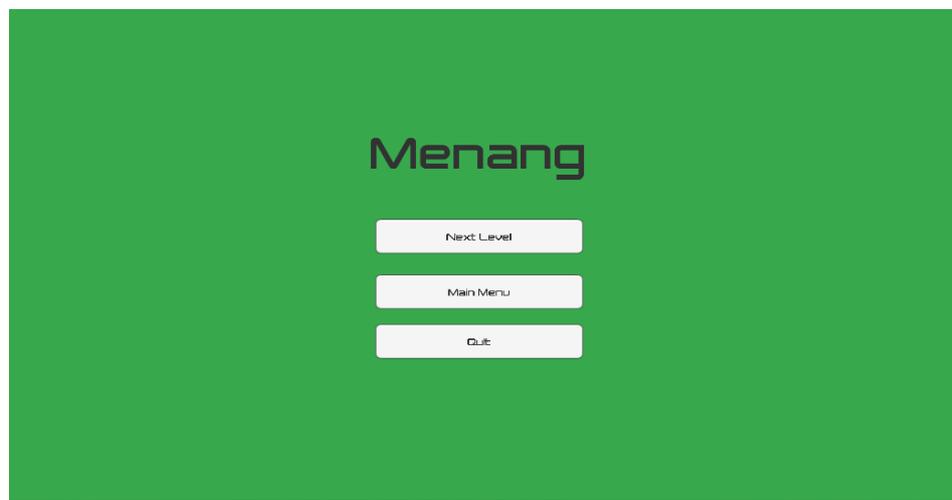
### 4.1.3 Tampilan Kuis



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.

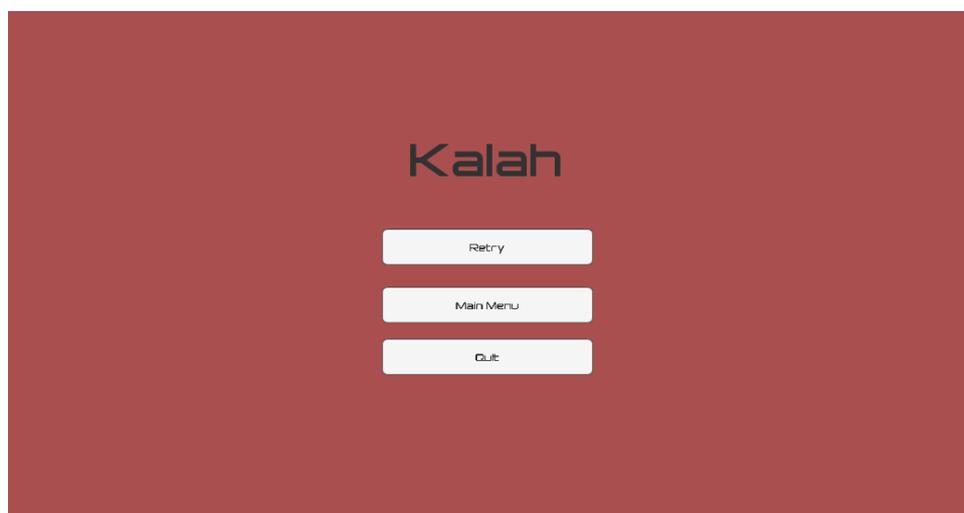
#### 4.1.4 Tampilan Ketika Menang



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 Ketika 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* (SVM)

Penerapan SVM pada penelitian ini menggunakan Bahasa pemrograman *python*. pemilihan bahasa pemrograman *python* dilakukan karena terdapat *library* SVM sehingga mempermudah proses penulisan *code*. Penerapan algoritma *Sequential Minimum Optimization* sebagai pengatur *level* kesulitan sudah include di *library* SVM. Tabel 4.1 merupakan penerapan SVM menggunakan *library python*.

Tabel 4. 1 Penerapan SVM menggunakan *Library Python*

```
# Import library
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import matplotlib.pyplot as plt
import joblib

# Muat dataset
df = read_csv('svm.csv')

# Praproses dataset
target = []
for i in range(len(df['level'])):
    target.append(df['level'][i])
df['target'] = target
X = df.drop(['level', 'target'], axis=1)
y = df.target

# Bagi dataset menjadi set data latih dan uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Buat dan latih model
model = SVC(kernel='rbf')
```

```
model.fit(X_train, y_train)

# Lakukan prediksi pada set data uji
pred = model.predict(X_test)

# Hitung dan cetak akurasi
akurasi = accuracy_score(pred, y_test)
print("Akurasi menggunakan {}: {}".format('rbf', akurasi))
```

Pada tabel 4.1 terdapat beberapa langkah untuk melatih dan menggunakan model *Support Vector Machine* (SVM) dengan kernel RBF (*Radial Basis Function*) menggunakan *library scikit-learn (sklearn)* di *python*. Kode tersebut dapat digunakan untuk memuat *dataset*, melakukan praproses pada *dataset*, membagi *dataset* menjadi set data latih dan uji, melatih model SVM, melakukan prediksi pada set data uji, dan menghitung akurasi dari model yang dilatih.

### 4.3 Pengujian

#### 4.3.1 Dataset

*Dataset* yang akan digunakan sebagai langkah dalam pengujian di ambil dari skripsi sebelumnya dengan topik serupa menggunakan metode *Random Forest* sebagai pengatur *level* kesulitan. *Dataset* pada tabel 4.1 di peroleh dari percobaan sebanyak jumlah *dataset*.

Tabel 4. 2 *Dataset* Pengujian

Skor	Waktu	Darah	Tipe Musuh	Jumlah Musuh	Jumlah Barang	Nilai Barang	<i>Level</i>
5	400	100	1	2	5	2	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
15	300	90	1	4	10	2	2
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
25	280	80	2	5	15	3	3
51	168	84	5	8	1	3	2
63	160	80	9	6	4	3	3
87	144	72	5	16	3	3	3
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
30	260	70	2	6	30	3	4

72	154	77	10	15	1	4	3
93	140	70	7	20	4	4	4
108	130	65	7	6	4	4	4
114	126	63	9	19	3	4	4
120	122	61	7	17	4	4	4
132	114	57	10	12	3	4	5
135	112	56	9	10	2	4	5
156	98	49	7	19	4	4	6
165	92	46	10	13	3	4	6
300	2	1	5	17	2	2	10
207	64	32	7	7	5	4	7
219	56	28	9	10	6	4	8
258	30	15	5	10	5	4	9
273	20	10	9	19	1	4	10
291	8	4	9	18	3	4	10
81	148	74	5	16	5	5	3
90	142	71	9	8	3	5	3
99	136	68	9	8	1	5	4
162	94	47	7	17	3	5	6
168	90	45	7	20	3	5	6
189	76	38	9	5	6	5	7
45	240	60	3	6	40	3	5
225	52	26	9	7	2	5	8
240	42	21	5	14	2	5	8
261	28	14	9	10	4	5	9
276	18	9	7	6	2	5	10
282	14	7	9	15	3	5	10
21	188	94	9	16	6	6	1
33	180	90	5	8	1	6	2
45	172	86	7	7	2	6	2
60	162	81	7	15	5	6	2
84	146	73	7	6	5	6	3
102	134	67	7	18	1	6	4
111	128	64	10	18	4	6	4
144	106	53	7	20	2	6	5
171	88	44	7	8	6	6	6
186	78	39	5	11	6	6	7
198	70	35	7	6	3	6	7
249	36	18	5	19	1	6	9
252	34	17	9	12	3	6	9
12	194	97	9	18	6	7	1
48	170	85	7	6	3	7	2
69	156	78	7	15	1	7	3
105	132	66	7	19	5	7	4

138	110	55	5	10	3	7	5
192	74	37	10	8	6	7	7
216	58	29	9	7	1	7	8
264	26	13	9	20	1	7	9
267	24	12	5	16	1	7	9
42	174	87	9	20	5	8	2
96	138	69	10	19	1	10	4
57	164	82	9	6	2	8	2
117	124	62	9	11	6	8	4
177	84	42	7	12	4	8	6
294	6	3	9	5	5	8	10
297	4	2	10	16	2	8	10
75	152	76	5	18	1	9	3
126	118	59	7	20	1	9	5
180	82	41	9	19	1	9	6
237	44	22	9	10	5	9	8
279	16	8	5	16	4	9	10
9	196	98	7	18	3	10	1
18	190	95	7	13	4	10	1
27	184	92	5	20	1	10	1
54	166	83	10	16	5	10	2
96	138	69	10	19	1	10	4
159	96	48	5	15	2	10	6
195	72	36	7	16	6	10	7
213	60	30	5	14	2	10	8
234	46	23	10	11	3	10	8

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

#### 4.3.2 Pengujian

Pada pengujian menggunakan 80% dari *dataset* sebagai data *training* dan menggunakan 20% dari *dataset* sebagai data *testing*. Hasil perhitungan *level* dari data *testing* dalam implementasi *Support Vector Machine* (SVM) menggunakan *Sequential Minimum Optimization* (SMO) terdapat pada tabel 4.2.

Tabel 4. 3 Hasil Pengujian SVM

Skor	waktu	Darah <i>Player</i>	Nilai barang	Jumlah Barang	Tipe Musuh	Jumlah Musuh	<i>level</i>		
							RF	SVM	
								ST	SMO
159	96	48	5	15	2	10	6	6	6
9	196	98	7	18	3	10	1	1	1
246	38	19	7	11	6	3	9	9	9
279	16	8	5	16	4	9	10	10	8
156	98	49	7	19	4	4	6	5	5
39	176	88	9	20	6	3	2	2	2
237	44	22	9	10	5	9	8	8	8
285	12	6	7	20	5	10	10	10	10
60	162	81	7	15	5	6	2	2	2
66	158	79	9	16	1	2	3	3	3
72	154	77	10	15	1	4	3	3	3
261	28	14	9	10	4	5	9	9	9
210	62	31	7	15	6	1	7	7	7
42	174	87	9	20	5	8	2	2	2
234	46	23	10	11	3	10	8	8	8
192	74	37	10	8	6	7	7	7	7
147	104	52	9	13	4	3	5	5	5
222	54	27	7	15	5	1	8	8	8
180	82	41	9	19	1	9	6	7	7
84	146	73	7	6	5	6	3	4	4

Tabel 4.2 merupakan *level* yang di hasilkan dari implementasi SVM-SMO yang menggunakan 20% *dataset* yang di jadikan sebagai data uji. Terdapat hasil *level* dari metode *Random Forest* yang di gunakan pada penelitian terdahulu dan SVM yang menggunakan model *training Sequential Training*. Hasil dari setiap metode yang di terapkan nantinya akan di bandingkan dengan SVM yang menggunakan *Sequential Minimum Optimization*. *Confusion Matrix* digunakan untuk mendapatkan nilai *accury*, *presisi*, *recall*, dan *F-measure*.

Perhitungan *Confusion Matrix* metode *Random Forest* (RF) dengan *Sequential Minimum Optimization* (SMO).

Tabel 4. 4 Hasil *Confusion Matrix* antara RF dengan SMO

<i>Level</i>		Prediksi									
		1	2	3	4	5	6	7	8	9	10
Actual	1	1	0	0	0	0	0	0	0	0	0
	2	0	3	1	0	0	0	0	0	0	0
	3	0	0	2	0	0	0	0	0	0	0
	4	0	0	0	3	0	0	0	0	0	0
	5	0	0	0	0	2	0	0	0	0	0
	6	0	0	0	0	1	1	0	0	0	0
	7	0	0	0	0	0	0	1	0	0	0
	8	0	0	0	0	0	0	0	2	0	0
	9	0	0	0	0	0	0	0	0	2	0
	10	0	0	0	0	0	0	0	0	0	3

Berdasarkan table 4.3 maka dapat di ketahui jumlah TP, FN, FP, dan TN dari *confusion matrix* antara RF dan ST, yaitu:

Tabel 4. 5 Jumlah TP, FN, FP, dan TN

<i>Level</i>	TP	FN	FP	TN
1	1	0	0	21
2	3	1	0	18
3	2	0	1	19
4	3	0	0	19
5	2	0	1	19
6	1	1	0	20
7	1	0	0	21
8	2	0	0	20
9	2	0	0	20
10	3	0	0	19
TOTAL	20	2	2	196

Langkah selanjutnya yaitu, menghitung nilai *accuracy*, *presisi*, *recall*, dan *F-measure*.

$$\text{Akurasi} = \frac{20+196}{20+196+2+2} \times 100\% = 0.9$$

$$\text{Presisi} = \frac{20}{20+2} \times 100\% = 0.83$$

$$\text{Recall} = \frac{20}{20+2} \times 100\% = 0.93$$

$$F\text{-measure} = \frac{0.83 \times 0.93}{0.83+0.93} \times 100\% = 0.83$$

Perhitungan *Confusion Matrix* metode *Sequential Training* (ST) dengan *Sequential Minimum Optimization* (SMO).

Tabel 4. 6 Hasil *Confusion Matrix* antara ST dengan SMO

Level		Prediksi										
		1	2	3	4	5	6	7	8	9	10	
Actual	1	1	0	0	0	0	0	0	0	0	0	0
	2	0	3	1	0	0	0	0	0	0	0	0
	3	0	1	1	1	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	2	0	0	0	0	0	0
	6	0	0	0	0	1	1	1	0	0	0	0
	7	0	0	0	0	0	0	2	0	0	0	0
	8	0	0	0	0	0	0	0	3	0	0	0
	9	0	0	0	0	0	0	0	0	2	0	0
	10	0	0	0	0	0	0	0	0	1	0	1

Berdasarkan table 4.3 maka dapat di ketahui jumlah TP, FN, FP, dan TN dari *confusion matrix* antara ST dan SMO, yaitu:

Tabel 4. 7 Jumlah TP, FN, FP, dan TN

<i>Level</i>	TP	FN	FP	TN
1	1	0	0	21
2	3	1	1	17
3	1	2	1	18
4	0	0	1	21
5	2	0	1	19
6	1	2	0	19
7	2	0	1	19
8	3	0	1	18
9	2	0	0	19
10	1	1	0	20
Total	16	6	6	191

Langkah selanjutnya yaitu, menghitung nilai *accuracy*, *presisi*, *recall*, dan *F-measure*.

$$Akurasi = \frac{20+196}{20+196+2+2} \times 100\% = 0.85$$

$$Presisi = \frac{20}{20+2} \times 100\% = 0.83$$

$$Recall = \frac{20}{20+2} \times 100\% = 0.93$$

$$F\text{-maesure} = \frac{0.83 \times 0.93}{0.83+0.93} \times 100\% = 0.83$$

Hasil dari kedua perhitungan confusion matrix di atas dengan RF yang di gunakan sebagai data actual dan di bandingkan dengan hasil *level* dari model pelatihan SVM (SMO dan ST) di peroleh hasil tingkat akurasi sebagai berikut : Random Forest dengan SVM (Sequential Minimum Optimization) memperoleh nilai akurasi mencapai 90%. Sedangkan SVM-SMO dengan SVM-ST memperoleh nilai akurasi sebesar 85%.

#### 4.4 Integrasi dengan Islam

##### a. *Mu'amallah ma'a Allah*

Di dalam Al-Quran ada banyak sekali ayat-ayat yang berkenaan dengan bencana alam. Allah SWT memberikan banyak peringatan kepada manusia tentang bencana dalam berbagai ayat-ayatnya. Allah menyampaikan bahwa salah satu ujian dalam kehidupan manusia di muka bumi adalah bencana atau musibah agar manusia sentiasa bersabar dalam menghadapinya. Hal ini terdapat dalam surah *Al-Baqarah* ayat 155-157 sebagai berikut:

وَلَنَبْلُوَنَّكُمْ بِشَيْءٍ مِّنَ الْخَوْفِ وَالْجُوعِ وَنَقْصٍ مِّنَ الْأَمْوَالِ وَالْأَنْفُسِ وَالثَّمَرَاتِ ۗ وَبَشِّرِ الصَّابِرِينَ (155)  
الَّذِينَ إِذَا أَصَابَتْهُمُ مُّصِيبَةٌ قَالُوا إِنَّا لِلَّهِ وَإِنَّا إِلَيْهِ رَاجِعُونَ (156) أُولَٰئِكَ عَلَيْهِمْ صَلَوَاتٌ مِّن رَّبِّهِمْ وَرَحْمَةٌ  
وَأُولَٰئِكَ هُمُ الْمُهْتَدُونَ (157)

*“Dan kami pasti akan menguji kamu dengan sedikit ketakutan, kelaparan, kekurangan harta, jiwa, dan buah-buahan. Dan sampaikanlah kabar gembira kepada orang-orang yang sabar (ayat 155), (yaitu) orang-orang yang apabila ditimpa musibah, mereka berkata "Innā lillāhi wa innā ilaihi rāji'ūn" <sup>1</sup> (sesungguhnya kami milik Allah dan kepada-Nyalah kami kembali) (ayat !56), Mereka itulah yang memperoleh ampunan dan rahmat dari Tuhannya, dan mereka itulah orang-orang yang mendapat petunjuk.”*

Berdasarkan Tafsir Jalalain Surat Al Baqarah Ayat 155-157 menjelaskan bahwasannya Allah akan memberimu cobaan berupa sedikit ketakutan) terhadap musuh, (kelaparan) paceklik, (kekurangan harta) disebabkan datangnya malapetaka, (dan jiwa) disebabkan pembunuhan, kematian dan penyakit, (serta buah-buahan) karena bahaya kekeringan, artinya Kami akan menguji kamu, apakah kamu bersabar atau tidak. (Dan

sampaikanlah berita gembira kepada orang-orang yang sabar) bahwa mereka akan menerima ganjaran kesabaran itu berupa surga.

Sesungguhnya kita (orang-orang yang sabar) ini milik Allah; maksudnya menjadi milik dan hamba-Nya yang dapat diperlakukan-Nya sekehendak-Nya, ('*wa innaa ilaihi raaji`uun*') artinya dan sesungguhnya kepada-Nyalah kita akan kembali, yakni ke akhirat, di sana kita akan diberi-Nya balasan.

Kolerasi dari ayat-ayat tersebut pada penelitian ini, yaitu bahwasannya bencana alam dan berbagai musibah merupakan bagian dari pada kehidupan manusia di muka bumi ini, dimana dengan menelaah ayat-ayat tersebut kita harus bersikap bijak dan bersabar dengan apa yang terjadi saat kita (manusia) tertimpa sebuah musibah, dan senantiasa ingat terhadap Allah SWT.

Sebagaimana konsep dari pada *game* ini dimana kita di haruskan untuk melewati sebuah rintangan (cobaan) dengan keberanian sehingga kita dapat melewati semua rintangan tersebut dan mencapai tujuan dengan hasil yang baik.

b. *Mu'amallah ma'a An-nas*

Tema yang di rancang pada penelitian ini adalah mengenai bencana alam. Dimana penelitian ini menerapkan *game* sebagai media dalam mensimulasikan salah satu bencana alam gunung meletus agar manusia mengetahui apa penyebab dari datangnya sebuah bencana alam dan tindakan yang di perlukan saat terjadi bencana alam. Pembuatan *game* ini bertujuan untuk memberikan pengetahuan terhadap pemain berupa kuis dan



alam dengan menggunakan contoh bencana gunung meletus. Selain itu, fitur kuis dalam permainan ini secara tidak langsung mengajarkan kepada pemain bahwa bencana alam sering kali disebabkan oleh perilaku manusia. Konsep ini juga ditegaskan dalam Al-Qur'an, Surah Ar-Ruum: 41 sebagai berikut:

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ

*“Telah tampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia; Allah menghendaki agar mereka merasakan sebagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar).”*

Merujuk pada az-Zuhaili, Tafsir Al-Munir volume 11 menjelaskan bahwa ayat di atas mengungkapkan bahwa bencana-bencana di dunia, seperti kelaparan, disebabkan oleh tindakan jahat manusia, seperti perampokan, pencurian, dan penindasan terhadap lingkungan. Akibatnya, lingkungan yang dulunya subur menjadi tandus dan tidak dapat dihuni, bahkan dapat membahayakan kehidupan manusia.

*Game* ini menawarkan serangkaian pertanyaan kuis yang mengajak pemain untuk menjaga lingkungan. Tujuan dari *game* ini adalah memberikan pemahaman yang lebih kepada pemain tentang pentingnya setiap individu dalam menjaga alam. Dengan menjaga alam tempat tinggal kita, diharapkan alam akan memberikan manfaat dan kebaikan bagi manusia, menciptakan kehidupan yang damai dan melindungi dari bencana seperti kelaparan dan tanah longsor.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Hasil penelitian yang telah dilakukan menyatakan, bahwa penerapan metode *Support Vector Machine* (SVM) menggunakan *Sequential Minimum Optimization* (SMO) sebagai pengaturan *level* kesulitan pada *game* edukasi gunung meletus untuk menentukan kesulitan permainan, dapat disimpulkan bahwa:

1. Hasil dari pengukuran akurasi metode SVM-SMO pada percobaan yang dilakukan pada *game* yang menerapkan metode SVM-SMO dengan *game* yang menerapkan metode *Random Forest*, menghasilkan akurasi sebesar 90%.
2. Hasil dari pengukuran akurasi metode SVM-SMO pada percobaan yang dilakukan pada *game* yang menerapkan metode SVM-SMO dengan *game* yang menerapkan metode *Support Vector Machine* yang menggunakan model *training Sequential Training*, menghasilkan akurasi sebesar 85%.
3. Dari penelitian di simpulkan bahwa metode SVM-SMO dapat di gunakan sebagai pengatur *level* kesulitan dimana SMO bekerja dengan cara mengklasifikasi 20% *dataset* yang di gunakan sebagai data uji untuk di klasifikasi.

## 5.2 Saran

Dalam penelitian ini penulis yakin dan penuh kesadaran bahwa penelitian ini terdapat banyak yang nantinya sangat perlu untuk dilakukan pengembangan terhadap ilmu pengetahuan diantaranya sebagai berikut:

1. Perlunya penambahan keluaran sehingga *game* bisa lebih menarik lagi, yang mana pada penelitian ini hanya terdapat satu keluaran berupa *level* kabut saja, karena keluaran seputar aspek kognitif, afektif, dan psikomotorik sudah di kategorikan dalam setiap *levelnya*.
2. Dalam pembuatan *level* kesulitan pada *game* diharapkan musuh bisa menyerang dari jarak jauh sehingga akan menghasilkan nilai yang berbeda yang dapat mempengaruhi gaya permainan dan proses perhitungan.

Pada penelitian ini hanya terdapat 10 tingkat kesulitan sehingga diharapkan pada peneliti selanjutnya dapat membuat tingkat kesulitan lebih banyak lagi sehingga proses berjalannya permainan menjadi lebih menarik.

## DAFTAR PUSTAKA

- Abdurakhman, O., & Rusli, R. (2015). Teori Belajar dan Pembelajaran Inovatif. *Didakti Tauhidi: Jurnal Pendidikan Guru Sekolah Dasar*, 2(1), 33.
- Adrian, M. R., Putra, M. P., Rafialdy, M. H., & Rakhmawati, N. A. (2021). Perbandingan Metode Klasifikasi Random Forest dan SVM Pada Analisis Sentimen PSBB. *Jurnal Informatika Upgris*, 7(1). <https://doi.org/10.26877/jiu.v7i1.7099>
- Afrizal Laksita Akbar. (2015). *Implementasi Algoritma Support Vector Machine Untuk Mengetahui Tingkat Risiko Penyakit Stroke*. Universitas Brawijaya.
- B. S. Avi Shena, B. Sitohang and S. A. Rukmono, "Application of Dynamic Difficulty Adjustment on Evidence-centered Design Framework for Game Based Learning," 2019 International Conference on Data and Software Engineering (ICoDSE), Pontianak, Indonesia, 20, doi: 10.1109/ICoDSE48700. 2019. 9092725. (2020). *Penerapan Penyesuaian Kesulitan Dinamis pada Kerangka Desain yang Berpusat pada Bukti untuk Game Pembelajaran Berbasis*.
- Cahyani, B. (2020). *Implementasi Perlin Noise Pada Simulasi Kabut Skripsi* (Issue 1969).
- Christie Gosal, L., Ch Tarore, R., & Karongkong, H. H. (2018). Analisis Spasial Tingkat Kerentanan Bencana Gunung Api Lokon Di Kota Tomohon. In *Jurnal Spasial* (Vol. 5, Issue 2).
- Colwell, A. M., & Glavin, F. G. (2018). Colwell's castle defence: A custom game using dynamic difficulty adjustment to increase player enjoyment. *ArXiv, December*.
- 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>
- Huang, R., & Sun, S. (2012). Sequential training of semi-supervised classification based on sparse Gaussian process regression. *Proceedings - International Conference on Machine Learning and Cybernetics*, 2, 702–707. <https://doi.org/10.1109/ICMLC.2012.6359010>
- 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.
- Muyasaroh, S. M., & Sudarmilah, E. (2019). Game Edukasi Mitigasi Bencana Kebakaran Berbasis Android. *PROtek : Jurnal Ilmiah Teknik Elektro*, 6(1),

31–35. <https://doi.org/10.33387/protk.v6i1.1029>

- Nugroho, F., Yuniarno, E. M., & Hariadi, M. (2019). Penerapan materi ilmu pengetahuan alam pada serious game sosialisasi mitigasi bencana berbasis model teori aktivitas dan taksonomi bloom. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 5(2), 106. <https://doi.org/10.26594/register.v5i2.1479>
- Nurhidayati, A., & Sunarsih, E. S. (2013). Peningkatan Hasil Belajar Ranah Afektif Melalui Pembelajaran Model Motivasional. *Jurnal Ilmiah Pendidikan Teknik Dan Kejuruan*, 6(2), 112–116. <https://doi.org/10.20961/jiptek.v6i2.12614>
- Platt, J. (1999). Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel Methods-Support Vector learning*. Cambridge, MA: MIT Press, July 1998, 185–208.
- Rima Ariona N. (2014). Game Edukasi Pengenalan Warna Sebagai Media-Pembelajaran Anak Usia Prasekolah Berbasis Android. *Jurnal Teknik Elektro*, 6(1), 30. <https://doi.org/10.15294/jte.v6i1.3573>
- Ruwaida, H. (2019). Proses Kognitif dalam Taksonomi Bloom Revisi : Analisis Kemampuan Mencipta (C6) Pada Pembelajaran Fikih Di MI Miftahul Anwar Desa Banua Lawas. *Al-Madrasah: Jurnal Pendidikan Madrasah Ibtidaiyah*, 4(1), 51. <https://doi.org/10.35931/am.v4i1.168>
- Sekar, B., & Shena, A. (2019). *Application of Dynamic Difficulty Adjustment on Evidence-centered Design Framework for Game Based Learning*.
- Suryani, A. S. (2012). Penanganan Asap Kabut Akibat Kebakaran Hutan di Wilayah Perbatasan Indonesia. *Aspirasi: Jurnal Masalah-Masalah Sosial*, 59–76.
- Syafri Pratama, K. (2022). *Pengaturan Level Kesulitan Pada Game Edukasi Bencana Gunung Meletus Menggunakan Metode Random Forest*.
- Warpefelt, H. (2016). *The Non-Player Character: Exploring the believability of NPC presentation and behavior* (Issue 16).
- Willy Sutina. (2010). *Pengaruh Algoritma Sequential Minimal Optimization pada Support Vector Machine untuk Klasifikasi Data* [Universitas Telkom]. <https://repository.telkomuniversity.ac.id/pustaka/95237/pengaruh-algoritma-sequential-minimal-optimization-pada-support-vector-machine-untuk-klasifikasi-data-influence-of-sequential-minimal-optimization-algorithm-on-support-vector-machine-for-data-classif>