PENERAPAN ALGORITMA *SUPPORT VECTOR MACHINE* PADA PLATFORM *QUANTUM COMPUTING* DALAM PENDETEKSIAN *MALICIOUS SOFTWARE*

TUGAS AKHIR

Diajukan Untuk Melengkapi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

OLEH :

Reza Mahesa Azandi
09011281924034

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2023

LEMBAR PENGESAHAN

PENERAPAN ALGORITMA *SUPPORT VECTOR MACHINE* PADA PLATFORM *QUANTUM COMPUTING* DALAM PENDETEKSIAN *MALICIOUS SOFTWARE*

TUGAS AKHIR

Program Studi Sistem Komputer

Jenjang S1

Oleh

Reza Mahesa Azandi

09011281924034

Indralaya, 7 Oktober 2023

Mengetahui,

Ketua Jurusan Sistem Komputer	Pembimbing Tugas Akhir
Dr. Ir. Sukemi, M.T.	Ahmad Heryanto, S.Kom, M.T.
NIP. 196612032006041001	NIP. 198701222015041002

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada :

Hari : Jum'at

Tanggal : 15 September 2023

Tim Penguji :

1. **Ketua : Dr. Ahmad Zarkasi, M. T.**

2. **Sekretaris : Rahmat Fadli Isnanto, M.Sc.**

3. **Penguji : Iman Saladin B. Azhar, M.MSI**

4. **Pembimbing : Ahmad Heryanto, S.Kom, M.T.**

Mengetahui,

Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.

NIP. 196612032006041001

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : Reza Mahesa Azandi

NIM : 09011281924034

Judul : Penerapan *Algoritma Support Vector Machine* pada Platform *Quantum Computing* dalam Pendeteksian *Malicious Software*

Hasil Pengecekan Software Turnitin : 5%

Menyatakan bahwa laporan tugas akhir saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila ditemukan unsur penjiplakan atau plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tanpa paksaan dari siapapun.

Indralaya, 1 Oktober 2023

Reza Mahesa Azandi

NIM. 09011281924034

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "**Penerapan Algoritma *Support Vector Machine* pada Platform *Quantum Computing* dalam Pendeteksian *Malicious Software***".

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada beberapa pihak atas ide dan saran serta bantuannya dalam menyelesaikan penulisan Tugas Akhir ini. Oleh karena itu, penulis ingin mengucapkan rasa syukur kepada Allah SWT dan terimakasih kepada yang terhormat :

1. Allah SWT, yang telah memberikan rahmat dan karunia-Nya sehingga saya dapat menyelesaikan penulisan Tugas Akhir ini dengan baik dan lancar.
2. Orang tua saya tercinta yang telah membesarkan saya dengan penuh kasih sayang dan selalu mengajarkan saya dalam berbuat hal yang baik. Terimakasih untuk segala do'a, motivasi dan dukungannya baik moril, materil maupun spritual selama ini.
3. Alm. Bapak Jaidan Jauhari, S.Pd., M.T., selaku mantan Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Prof. Dr. Erwin, S.Si, M.Msi, selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Bapak Dr. Ir. Sukemi, M.T., selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
6. Bapak Ahmad Heryanto, S.Kom, M.T., selaku Dosen Pembimbing Tugas Akhir yang telah berkenan meluangkan waktunya untuk membimbing, memberikan saran dan motivasi kepada penulis dalam menyelesaikan Tugas Akhir ini.
7. Iman Saladin B. Azhar S.Kom, M.MSI, selaku Dosen Pembimbing Akademik dan Dosen Penguji Tugas Akhir penulis yang telah berkenan meluangkan waktunya untuk membimbing, memberikan saran dan

motivasi kepada penulis dari awal hingga akhir kuliah penulis.

8. Mbak Renny selaku admin Jurusan Sistem Komputer yang telah membantu mengurus seluruh berkas.
9. Serta semua pihak yang telah membantu.

Penulis menyadari bahwa laporan ini masih sangat jauh dari kata sempurna. Untuk itu kritik dan saran yang membangun sangatlah diharapkan penulis. Akhir kata penulis berharap, semoga tugas akhir ini bermanfaat dan berguna bagi orang lain.

Wassalamu'alaikum Wr. Wb.

Indralaya, 1 Oktober 2023

Penulis,

Reza Mahesa Azandi

NIM. 09011281924034

Penerapan Algoritma Support Vector Machine pada Platform Quantum Computing dalam Pendeteksian Malicious Software

Reza Mahesa Azandi (09011281924034)
Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya
Email : reza.mahesa.azandi@gmail.com

ABSTRAK

Kebutuhan komputasi yang lebih efisien akan meningkat seiring perkembangan teknologi, dengan quantum computing dapat menyediakan kekuatan komputasi yang lebih efisien dibandingkan komputasi konvensional. Perkembangan teknologi juga akan mempengaruhi jumlah serangan cyber, serangan cyber yang paling umum ditemui pada saat ini adalah serangan malware. Salah satu algoritma yang cocok untuk mengklasifikasi adalah Support Vector Machine. Penggunaan Support Vector Machine dengan quantum computing dibantu dengan metode Stochastic Gradient Descent untuk mengoptimisasi parameter. Pada penelitian ini dilakukan optimisasi sumber daya sirkuit kuantum dan pengujian skenario untuk mendapatkan model dan hasil yang optimal. Sumber daya sirkuit kuantum yang akan dioptimisasi adalah rangkaian gerbang logika kuantum dan jumlah qubit yang digunakan. Total skenario yang akan diuji ada sembilan yang terdiri dari pembagian rasio data train dengan data test dan variasi nilai parameter learning rate. Dataset yang digunakan adalah CIC-MalMem-2022 yang memiliki dua label yaitu malware dan benign. Performa model terbaik dari penelitian ini menghasilkan nilai *precision* 97.42%, *recall* 98.69%, *specificity* 96.98%, f1-*score* 98.05%, dan *accuracy* 97.9%.

Kata Kunci : Pendeteksian Malware, *Quantum Computing, Stochastic Gradient Descent, Support Vector Machine* (SVM).

Support Vector Machine Implementation on Quantum Computing Platform in Detecting Malicious Software

Reza Mahesa Azandi (09011281924034)
Computer System Department, Computer Science Faculty, Sriwijaya University
Email : reza.mahesa.azandi@gmail.com

ABSTRACT

The necessity for more efficient computing will increase as technology develops, with quantum computing able to provide more efficient computing power than conventional computing. Technological developments will also influence the number of cyber attacks, the most common cyber attacks currently encountered are malware attacks. One of the algorithm that is suitable for classification is the Support Vector Machine. The use of Support Vector Machine with quantum computing is assisted by the Stochastic Gradient Descent method to optimize parameters. In this research, quantum circuit resource optimization and scenario testing were carried out to obtain optimal models and results. The quantum circuit resources that will be optimized are the quantum logic gate circuit and the number of qubits used. There are a total of nine scenarios that will be tested, consisting of dividing the ratio of train data to test data and variations in the learning rate parameter values. The dataset used is CIC-MalMem-2022 which has two labels, namely malware and benign. The best model performance from this research produced precision values of 97.42%, recall 98.69%, specificity 96.98%, f1-score 98.05%, and accuracy 97.9%.

Keywords : Malware Detection, Quantum Computing, Stochastic Gradient Descent, Support Vector Machine (SVM).

DAFTAR ISI

LEMBAR PENGESAHAN ..... **ii**
HALAMAN PERSETUJUAN ..... **iii**
HALAMAN PERNYATAAN ..... **iv**
KATA PENGANTAR ..... **v**
ABSTRAK ..... **vii**
ABSTRACT ..... **viii**
DAFTAR ISI ..... **ix**
DAFTAR GAMBAR ..... **xii**
DAFTAR TABEL ..... **xiv**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan 4
1.5 Manfaat 4
1.6 Sistematika Penulisan 5
BAB II TINJAUAN PUSTAKA **6**
2.1 Penelitian Terdahulu 6
2.2 *Malicious Software (Malware)* 13
2.2.1 Worm 13
2.2.2 Trojan 14
2.2.3 Botnet 14
2.2.4 Memanfaatkan kelemahan layanan jaringan 15
2.2.5 *Drive-by Downloads* 15
2.2.6 Rekayasa sosial 16
2.3 *Quantum Computing* 17
2.3.1 Mekanika Kuantum 17
2.3.1.1 Superposition dan Interferensi 18
2.3.1.2 Uncertainty 19
2.3.1.3 Entanglement 19
2.3.1.4 Notasi Dirac 20

2.3.2 Komputer Kuantum 21
2.3.3 Qubit 22
2.3.4 Bloch Sphere 23
2.3.5 Gerbang Logika Kuantum 24
2.4 Support Vector Machine 26
2.5 Principal Component Analysis 26
2.6 Confusion Matrix 27
2.6.1 *Precision* 28
2.6.2 *Recall* 28
2.6.3 *Specificity* 29
2.6.4 F1-*Score* 29
2.6.5 *Accuracy* 29
BAB III METODOLOGI PENELITIAN 30
3.1 Pendahuluan 30
3.2 Kerangka Kerja 30
3.3 Persiapan Data 31
3.4 Pra Pengolahan Data 32
3.4.1 *Encoding* 33
3.4.2 *Feature Selection* 33
3.4.3 *Principal Component Analysis (PCA)* 34
3.5 Pembagian Data Uji dan Latih 34
3.6 Pengujian Model 34
3.7 Percobaan Skenario 35
3.8 Evaluasi hasil 36
BAB IV HASIL DAN ANALISIS 37
4.1 Pendahuluan 37
4.2 Dataset 37
4.3 Pra Pengolahan Data 38
4.4 Perancangan Sirkuit Kuantum 40
4.4.1 Sirkuit A dengan Dua Qubit 41
4.4.2 Sirkuit B dengan Dua Qubit 42
4.4.3 Sirkuit A dengan Empat Qubit 42

4.4.4 Sirkuit B dengan Empat Qubit 43
4.4.5 Sirkuit A dengan Enam Qubit 44
4.4.6 Sirkuit B dengan Enam Qubit 45
4.5 Pengolahan Data 47
4.6 Evaluasi Hasil 48
4.7 Percobaan Skenario 51
4.7.1 Percobaan Skenario Satu 52
4.7.2 Percobaan Skenario Dua 53
4.7.3 Percobaan Skenario Tiga 54
4.7.4 Percobaan Skenario Empat 55
4.7.5 Percobaan Skenario Lima 56
4.7.6 Percobaan Skenario Enam 57
4.7.7 Percobaan Skenario Tujuh 58
4.7.8 Percobaan Skenario Delapan 59
4.7.9 Percobaan Skenario Sembilan 60
4.8 Analisa Hasil 61
BAB V KESIMPULAN 63
DAFTAR PUSTAKA 64
LAMPIRAN 70

DAFTAR GAMBAR

Gambar 2.1 Superposition pada qubit.....18
Gambar 2.2 Prinsip uncertainty 19
Gambar 2.3 Quantum entanglement 20
Gambar 2.4 Bloch sphere.....24
Gambar 2.5 Gerbang logika kuantum.....25
Gambar 3.1 Kerangka kerja.....31
Gambar 3.2 *Encoding*.....33
Gambar 3.3 *Feature selection* 33
Gambar 3.4 Algoritma model klasifikasi 35
Gambar 4.1 Pie chart sample data 37
Gambar 4.2 Tampilan fitur dan nilai dataset 38
Gambar 4.3 Fitur dataset sebelum *encoding* 38
Gambar 4.4 Proses *encode* pada fitur class 39
Gambar 4.5 Fitur dataset setelah proses *encoding* 39
Gambar 4.6 Proses *feature selection*.....39
Gambar 4.7 Principal Component Analysis.....40
Gambar 4.8 Perancangan sirkuit B 41
Gambar 4.9 Sirkuit A dengan dua qubit 41
Gambar 4.10 Hasil sirkuit A dengan dua qubit 42
Gambar 4.11 Sirkuit B dengan dua qubit.....42
Gambar 4.12 Hasil sirkuit B dengan dua qubit.....42
Gambar 4.13 Sirkuit A dengan empat qubit.....43
Gambar 4.14 Hasil sirkuit A dengan empat qubit 43
Gambar 4.15 Sirkuit B dengan empat qubit.....43
Gambar 4.16 Hasil sirkuit B dengan empat qubit.....44
Gambar 4.17 Sirkuit A dengan enam qubit.....44
Gambar 4.18 Hasil sirkuit A dengan enam qubit.....44
Gambar 4.19 Sirkuit B dengan enam qubit 45
Gambar 4.20 Hasil sirkuit B dengan enam qubit.....45
Gambar 4.21 Grafik hasil pengujian sirkuit 46
Gambar 4.22 Metode *Stochastic Gradient Descent*.....47
Gambar 4.23 Optimisasi parameter.....48

Gambar 4.24 Hasil data train 49
Gambar 4.25 Hasil data test 49
Gambar 4.26 Nilai confusion matrix pada data test 50
Gambar 4.27 Hasil pelatihan skenario satu 52
Gambar 4.28 Hasil pengujian skenario satu 53
Gambar 4.29 Hasil pelatihan skenario dua 53
Gambar 4.30 Hasil pengujian skenario dua 54
Gambar 4.31 Hasil pelatihan skenario tiga 54
Gambar 4.32 Hasil pengujian skenario tiga 55
Gambar 4.33 Hasil pelatihan skenario empat 55
Gambar 4.34 Hasil pengujian skenario empat 56
Gambar 4.35 Hasil pelatihan skenario lima 56
Gambar 4.36 Hasil pengujian skenario lima 57
Gambar 4.37 Hasil pelatihan skenario enam 57
Gambar 4.38 Hasil pengujian skenario enam 58
Gambar 4.39 Hasil pelatihan skenario tujuh 58
Gambar 4.40 Hasil pengujian skenario tujuh 59
Gambar 4.41 Hasil pelatihan skenario delapan 59
Gambar 4.42 Hasil pengujian skenario delapan 60
Gambar 4.43 Hasil pelatihan skenario sembilan 60
Gambar 4.44 Hasil pengujian skenario sembilan 61

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu 6
Tabel 2.2 Confusion matrix 28
Tabel 3.1 Keterangan fitur pada dataset 31
Tabel 3.2 Rasio pembagian data 35
Tabel 3.3 Nilai learning rate 36
Tabel 4.1 Hasil pengujian sirkuit 46
Tabel 4.2 Nilai confusion matrix pada data test.....50
Tabel 4.3 Hasil perhitungan manual 51
Tabel 4.4 Skenario percobaan.....51
Tabel 4.5 Hasil percobaan skenario 61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi mengarah pada kemajuan peradaban, ditemukan cara-cara baru untuk mengeksploitasi berbagai sumber daya fisik seperti material, kekuatan, dan energi. Meskipun komputer menjadi lebih kompak dan jauh lebih cepat dalam melakukan tugasnya, tugasnya tetap sama: memanipulasi dan menginterpretasikan pengkodean bit biner menjadi hasil komputasi yang berguna. Pengamatan oleh Gordon Moore pada tahun 1965 meletakkan dasar untuk apa yang kemudian dikenal sebagai "Hukum Moore", bahwa kekuatan pemrosesan komputer berlipat ganda setiap delapan belas bulan. Jika Hukum Moore diekstrapolasi secara naif ke masa depan, diketahui bahwa cepat atau lambat, setiap bit informasi harus dikodekan oleh sistem fisik berukuran subatomic [1].

Faktanya, dalam beberapa tahun terakhir jumlah transistor dalam sirkuit terpadu komputer mulai mengalami saturasi. Dari tahun 1971 sampai 2012, sirkuit terpadu yang dibuat Intel masih mengikuti hukum Moore dengan baik [2], namun pada tahun-tahun berikutnya, kenaikan jumlah transistor tiap tahunnya tidak lagi mengikuti hukum Moore. Mulai tahun 2016, industri sirkuit terpadu di seluruh dunia tidak lagi menjadikan hukum Moore sebagai landasan utama dalam rencana riset dan pengembangan mereka [3]. Kekuatan komputasi dari sebuah komputer sangat bergantung pada seberapa banyak transistor yang dapat masuk dalam sirkuit terpadu, sehingga permasalahan ini dapat menyebabkan kekuatan komputasi berhenti bertambah.

Resiko stagnasi kekuatan komputasi dan kebutuhan kekuatan komputasi untuk simulasi baik dalam dunia industri dan akademik yang terus bertambah menyebabkan dimulainya pencarian alternatif teknologi komputasi [4]. Richard Feynmann, dalam kuliah singkatnya pada Mei 1981 di California Institute of Technology, pernah mengatakan bahwa untuk dapat mensimulasikan alam yang bersifat kuantum maka dibutuhkan pula alat komputasi yang memiliki sifat kuantum [5], kata-kata ini dipercaya sebagai cikal bakal mulai digunakannya istilah komputasi kuantum, yaitu cara-cara pengolahan informasi yang dapat dilakukan menggunakan hukum-hukum dan sistem mekanika kuantum [6]. Seiring kemajuan-

kemajuan yang terjadi dalam teknologi kuantum, ketertarikan pada komputasi kuantum semakin meningkat. Pada tahun 1994, Peter Shor secara analitik menunjukkan komputasi kuantum mampu melakukan pemfaktoran bilangan prima yang besar dengan percepatan eksponensial jika dibandingkan dengan solusi komputasi klasik terbaik yang ada saat ini [7]. Penemuan ini menjadi pemicu penting yang meningkatkan perkembangan komputer kuantum secara pesat karena pemfaktoran bilangan prima yang besar sangat penting dalam aplikasi kriptografi [6].

Berbagai penelitian dilakukan untuk menyelesaikan permasalahan yang terkenal sulit dan mahal biaya komputasinya pada komputer konvensional dan meningkatkan perkembangan komputasi menggunakan *quantum computing* dengan algoritma kuantum. Salah satu ketertarikan pada komputasi kuantum adalah aplikasi komputasi kuantum dalam pembelajaran mesin. Metode pembelajaran mesin klasik mampu mengenali pola statistik dalam data sekaligus menghasilkan kembali data dengan pola tersebut, mereka mampu mengenali apa yang mereka mampu produksi. Pada tahun 2014, Lloyd et al. mengajukan algoritma versi kuantum dari Principal Component Analysis yang memiliki percepatan eksponensial jika dibandingkan apa yang bisa dilakukan komputer klasik [8].

Berdasarkan teori dan penelitian terdahulu yang sudah dijelaskan di atas, komputasi kuantum ini akan memiliki peran penting pada beberapa puluh tahun kedepan. Pada masa yang akan mendatang juga tidak menutup kemungkinan serangan cyber akan meningkat. Berdasarkan lepide, peringkat pertama pada serangan cyber yang paling umum adalah serangan malware [9]. Faktanya jumlah serangan malware meningkat pada tahun 2018 menjadi 10.5 miliar [10].

Perangkat lunak atau aplikasi telah ada di mana-mana dalam kehidupan sehari-hari, namun dari banyaknya aplikasi yang beredar terdapat aplikasi berbahaya yang dapat merugikan pengguna. Malware (aplikasi berbahaya) berjalan pada perangkat tanpa meminta izin pengguna. Secara umum, malware memiliki satu atau beberapa perilaku berikut: pemasangan paksa, pembajakan, mencuri dan mengubah data pengguna, pengumpulan informasi pengguna yang berbahaya, pemasangan berbahaya, penggabungan berbahaya, dan perilaku berbahaya lainnya [11]. Perilaku ini akan sangat melanggar hak sah pengguna, dan bahkan akan membawa kerugian besar bagi pengguna.

Berbagai algoritma *machine learning* digunakan untuk menemukan dan

mengklasifikasikan malware ke dalam jenis dan kelompoknya. Untuk mendeteksi serangan malware dapat menggunakan beberapa algoritma, seperti KNN, *Random Forest*, SVM, dan sebagainya. Berdasarkan penelitian sebelumnya untuk mendeteksi malware dengan algoritma *Random Forest* mencapai akurasi 63.49% [12], algoritma KNN mencapai akurasi 41% [13], dan algoritma naive bayes mencapai akurasi 63.53% [14].

Dari ketiga penelitian terdahulu yang mendeteksi malware, persentase akurasi yang dihasilkan masih belum maksimal. Dengan menggunakan algoritma SVM pada penelitian ini, penulis berasumsi akan mendapatkan hasil akurasi yang lebih tinggi. Kesalahan yang bisa diperbaiki oleh algoritma SVM dari penelitian yang menggunakan algoritma *Random Forest* adalah kemampuan mengatasi *overfitting*, algoritma SVM memiliki kemampuan untuk mengatasi masalah *overfitting* yang lumayan kuat [15]. Untuk penelitian yang menggunakan algoritma KNN, kesalahan yang akan diperbaiki pada penelitian ini adalah untuk memproses data sebelum mulai melatih data. Pada penelitian yang menggunakan algoritma *Naive Bayes*, hasil dari penelitian [16] menunjukkan bahwa algoritma SVM lebih akurat untuk mengklasifikasikan data. Selain dari perbaikan yang akan dilakukan pada penelitian ini, algoritma SVM memiliki beberapa kelebihan, yaitu SVM dapat melakukan generalisasi sehingga SVM dapat mengenali pola dari sebuah data dan algoritma SVM mudah untuk diimplementasikan karena dapat merumuskan *support vector* dalam masalah *quadratic programming* [17].

Dengan luasnya peluang yang dapat dieksplorasi dari *quantum computing* dan jumlah serangan malware yang kemungkinan beberapa puluh tahun kedepan akan terus meningkat, serta kelebihan dari algoritma *Support Vector Machine* untuk mengenali pola sebuah data, penelitian tugas akhir ini berfokus pada penerapan algoritma *Support Vector Machine* pada platform *quantum computing* dalam pendeteksian *malicious software*.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang sudah diuraikan di atas, maka perumusan masalah yang akan dibahas adalah sebagai berikut:

1. Bagaimana merancang model untuk mendeteksi malware dengan penerapan algoritma *Support Vector Machine* pada *quantum computing*.

2. Bagaimana mengoptimalkan penggunaan sumber daya *quantum computing*, seperti penggunaan gerbang kuantum pada sirkuit kuantum.
3. Bagaimana melakukan peningkatan kinerja model sehingga dapat menghasilkan performa yang optimal.

1.3 Batasan Masalah

Berikut batasan masalah pada Tugas Akhir ini, yaitu :

1. Menggunakan teknologi *quantum computing* dalam mendeteksi malware.
2. Algoritma *machine learning* yang digunakan adalah *support vector machine* (SVM).
3. Hasil klasifikasi atau keluaran berupa dua kelas.
4. Serangan cyber yang akan dideteksi adalah serangan malware.
5. Dataset yang digunakan adalah CIC-MalMem-2022.

1.4 Tujuan

Tujuan yang akan dicapai dari penelitian ini adalah sebagai berikut :

1. Menerapkan algoritma *Support Vector Machine* untuk mendeteksi serangan malware.
2. Mendeteksi serangan malware yang disamarkan dengan *memory dump*.
3. Merancang sirkuit kuantum yang dapat membantu mendeteksi serangan malware.
4. Menganalisa tingkat akurasi terhadap penerapan *quantum computing* untuk mendeteksi serangan malware.

1.5 Manfaat

Manfaat yang akan dicapai dari penelitian ini adalah sebagai berikut :

1. Dapat mengetahui cara menerapkan algoritma *Support Vector Machine* untuk mendeteksi serangan malware.
2. Dapat mendeteksi serangan malware yang disamarkan dengan *memory dump*.
3. Dapat merancang sirkuit kuantum yang dapat membantu mendeteksi

serangan malware.

4. Untuk mengetahui tingkat akurasi terhadap penerapan *quantum computing* untuk mendeteksi serangan malware.

1.6 Sistematika Penulisan

Sistematika yang akan digunakan dalam penulisan tugas akhir adalah :

BAB I PENDAHULUAN

Bab pertama akan memaparkan sistematis mengenai latar belakang, tujuan penelitian, rumusan masalah, serta bentuk sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Bab kedua akan menjelaskan teori-teori dasar yang akan menjadi landasan dari penelitian ini. Dasar teori yang akan dibahas pada bab ini adalah literatur mengenai serangan malware, *quantum computing*, dan algoritma *support vector machine*.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan proses dan rangkaian kegiatan dalam penelitian. Penelitian akan dimulai dari persiapan data, perancangan sirkuit kuantum, penggunaan algoritma *support vector machine* dan klasifikasi serangan malware.

BAB IV HASIL DAN ANALISIS

Bab ini akan memaparkan hasil pengujian yang diperoleh dan menjelaskan analisa terhadap hasil penelitian yang telah dilakukan.

BAB V KESIMPULAN

Bab ini akan menampung simpulan yang dapat disimpulkan dari hasil keseluruhan penelitian dan analisa.

DAFTAR PUSTAKA

[1] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.

[2] C. Y. Lee, “Transistor Degradations in Very Large-Scale-Integrated CMOS Technologies,” in *Very-Large-Scale Integration*, InTech, 2018.

[3] M. M. Waldrop, “More Than Moore,” *Nature*, vol. 530, no. 7589. Nature Publishing Group, pp. 144–147, Feb. 09, 2016, doi: 10.1038/530144a.

[4] E. Grumbling and M. Horowitz, *Quantum Computing: Progress and Prospects*, vol. 9781461418. 2018.

[5] R. P. Feynman, “Simulating physics with computers,” in *Feynman and Computation*, CRC Press, 2018, pp. 133–153.

[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2012.

[7] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999, doi: 10.1137/S0036144598347011.

[8] S. Lloyd, M. Mohseni, and P. Rebentrost, “Quantum principal component analysis,” *Nat. Phys.*, vol. 10, no. 9, pp. 631–633, 2014, doi: 10.1038/NPHYS3029.

[9] B. Jefferson, “The 15 Most Common Types of Cyber Attacks,” 2022. https://www.lepide.com/blog/the-15-most-common-types-of-cyber-attacks/.

[10] Statista Research Departement, “Annual number of malware attacks worldwide from 2015 to first half 2022,” 2022. .

[11] Y. Pan, X. Ge, C. Fang, and Y. Fan, “A Systematic Literature Review of Android Malware Detection Using Static Analysis,” *IEEE Access*, vol. 8, pp. 116363–116379, 2020, doi: 10.1109/ACCESS.2020.3002842.

[12] G. Mahajan, B. Saini, and S. Anand, “Malware classification using machine learning algorithms and tools,” Feb. 2019, doi:

10.1109/ICACCP.2019.8882965.

[13] A. R. Yogaswara, "Klasifikasi Malware Family menggunakan Metode k-Nearest Neighbor (k-NN)," *J. Repos.*, vol. 3, no. 3, Mar. 2021, doi: 10.22219/repositor.v2i3.1313.

[14] I. Anggraini, Y. N. Kunang, and Firdaus, "Penerapan Naive Bayes pada Pendeteksian Malware dengan Diskritisasi Variabel," 2019, doi: 10.35671/telematika.v13i1.886.

[15] H. Peng and X. Bai, "Comparative evaluation of three machine learning algorithms on improving orbit prediction accuracy," *Astrodynamics*, vol. 3, no. 4, pp. 325–343, Dec. 2019, doi: 10.1007/s42064-018-0055-4.

[16] M. Guia, R. R. Silva, and J. Bernardino, "Comparison of Naive Bayes, support vector machine, decision trees and random forest on sentiment analysis," in *IC3K 2019 - Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2019, vol. 1, pp. 525–531, doi: 10.5220/0008364105250531.

[17] A. M. Puspitasari, D. E. Ratnawati, and A. W. Widodo, "Klasifikasi Penyakit Gigi Dan Mulut Menggunakan Metode Support Vector Machine," *J-Ptiik*, vol. 2, no. 2, pp. 802–810, 2018, [Online]. Available: http://j-ptiik.ub.ac.id.

[18] B. Ricks and D. Ventura, "Training a quantum neural network," 2004.

[19] B. P. Lanyon *et al.*, "Experimental demonstration of a compiled version of shor's algorithm with quantum entanglement," *Phys. Rev. Lett.*, vol. 99, no. 25, 2007, doi: 10.1103/PhysRevLett.99.250505.

[20] A. Macaluso, L. Clissa, S. Lodi, and C. Satori, "Quantum Ensemble for Classification," 2007, doi: 10.48550/arXiv.2007.01028.

[21] F. Muhammad Rashid, "Quantum Computing," *J. Networks Telecommun. Syst.*, 2016, doi: 10.13140/RG.2.2.27565.23529.

[22] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," 2018, doi: 10.1103/PhysRevA.101.032308.

[23] A. Mandviwalla, K. Ohshiro, and B. Ji, "Implementing Grover's Algorithm on the IBM Quantum Computers," 2019, doi: 10.1109/BigData.2018.8622457.

[24] A. B. Mutiara, M. A. Slamet, R. Refianti, and Y. Sutanto, “Handwritten Numeric Image Classification with Quantum Neural Network using Quantum Computer Circuit Simulator,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 310–319, Oct. 2020, doi: 10.14569/IJACSA.2020.0111040.

[25] A. Chalumuri, R. Kune, and B. S. Manoj, “Training an Artificial Neural Network Using Qubits as Artificial Neurons: A Quantum Computing Approach,” in *Procedia Computer Science*, 2020, vol. 171, pp. 568–575, doi: 10.1016/j.procs.2020.04.061.

[26] A. Deb, G. W. Dueck, and R. Wille, “Exploring the Potential Benefits of Alternative Quantum Computing Architectures,” 2020, doi: 10.1109/TCAD.2020.3032072.

[27] K. Zhang and V. E. Korepin, “Depth optimization of quantum search algorithms beyond Grover’s algorithm,” *Phys. Rev. A*, vol. 101, no. 3, 2020, doi: 10.1103/PhysRevA.101.032346.

[28] K. Bertels *et al.*, “Quantum Computer Architecture Toward Full-Stack Quantum Accelerators,” *IEEE Trans. Quantum Eng.*, vol. 1, 2021, doi: 10.1109/tqe.2020.2981074.

[29] S. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. Bernal Neira, and D. Greenberg, “Formulating and Solving Routing Problems on Quantum Computers,” *IEEE Trans. Quantum Eng.*, vol. 2, 2021, doi: 10.1109/tqe.2021.3049230.

[30] H. Soeparno and A. S. Perbangsa, “Cloud Quantum Computing Concept and Development: A Systematic Literature Review,” in *Procedia Computer Science*, 2021, vol. 179, doi: 10.1016/j.procs.2021.01.084.

[31] M. Islam, M. Chowdhury, Z. Khan, and S. M. Khan, “Hybrid Quantum-Classical Neural Network for Cloud-Supported In-Vehicle Cyberattack Detection,” *IEEE Sensors Lett.*, vol. 6, no. 4, Apr. 2022, doi: 10.1109/LSENS.2022.3153931.

[32] R. M. A. El-Aziz, A. Rayan, O. R. Shahin, A. Elhadad, A. Abozeid, and A. I. Taloba, “Modified Deep Residual Quantum Computing Optimization Technique for IoT Platform,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 12,

2021, doi: 10.14569/IJACSA.2021.0121244.

[33] T. Hur, L. Kim, and D. K. Park, “Quantum convolutional neural network for classical data classification,” *Quantum Mach. Intell.*, vol. 4, no. 1, Jun. 2022, doi: 10.1007/s42484-021-00061-x.

[34] S. Choe and M. Perkowski, “Continuous Variable Quantum MNIST Classifiers —Classical-Quantum Hybrid Quantum Neural Networks,” *J. Quantum Inf. Sci.*, vol. 12, no. 02, pp. 37–51, 2022, doi: 10.4236/jqis.2022.122005.

[35] W. Li, Z. Lu, and D.-L. Deng, “Quantum Neural Network Classifiers: A Tutorial,” 2022, doi: 10.48550/arXiv.2206.02806.

[36] Y. Wang, Y. Wang, C. Chen, R. Jiang, and W. Huang, “Development of variational quantum deep neural networks for image recognition,” *Neurocomputing*, vol. 501, pp. 566–582, Aug. 2022, doi: 10.1016/j.neucom.2022.06.010.

[37] C. LeDoux and A. Lakhotia, “Malware and machine learning,” *Stud. Comput. Intell.*, vol. 563, pp. 1–42, 2015, doi: 10.1007/978-3-319-08624-8_1.

[38] G. A. N. Mohamed and N. B. Ithnin, “Survey on Representation Techniques for Malware Detection System,” *Am. J. Appl. Sci.*, vol. 14, no. 11, pp. 1049–1069, Nov. 2017, doi: 10.3844/ajassp.2017.1049.1069.

[39] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A survey on automated dynamic malware-analysis techniques and tools,” *ACM Computing Surveys*, vol. 44, no. 2. Feb. 2012, doi: 10.1145/2089125.2089126.

[40] O. Aslan and A. A. Yilmaz, “A New Malware Classification Framework Based on Deep Learning Algorithms,” *IEEE Access*, vol. 9, pp. 87936–87951, 2021, doi: 10.1109/ACCESS.2021.3089586.

[41] H. Tiirmaa-Klaar, J. Gassen, E. Gerhards-Padilla, and P. Martini, *SpringerBriefs in Cybersecurity*. 2013.

[42] A. Zimba, “A Bayesian Attack-Network Modeling Approach to Mitigating Malware-Based Banking Cyberattacks,” *Int. J. Comput. Netw. Inf. Secur.*, vol. 14, no. 1, pp. 25–39, Feb. 2022, doi: 10.5815/ijcnis.2022.01.03.

[43] O. Aslan and R. Samet, “A Comprehensive Review on Malware Detection

Approaches," *IEEE Access*, vol. 8. Institute of Electrical and Electronics Engineers Inc., pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[44] R. Komatwar and M. Kokare, "A Survey on Malware Detection and Classification," *J. Appl. Secur. Res.*, vol. 16, no. 3, pp. 390–420, 2021, doi: 10.1080/19361610.2020.1796162.

[45] M. Sikorski and A. Honig, *Practical malware analysis: the hands-on guide to dissecting malicious software. no starch press*. 2012.

[46] L. Rolf, "Is quantum mechanics useful?," *Philos. Trans. R. Soc. London. Ser. A Phys. Eng. Sci.*, vol. 353, no. 1703, pp. 367–376, 1995, doi: 10.1098/rsta.1995.0106.

[47] P. N. Telkar, P. Naik, A. Mabali, G. S H, G. S H, and S. Balikai, "Quantum Computers for Next Generation," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, 2020, doi: 10.32628/cseit2063118.

[48] S. T. Marella and H. S. K. Parisa, "Quantum Computing and Communications," 2020, doi: 10.5772/intechopen.94103.

[49] A. P, B. N. P. Alekhya, A. Malla, K. Sirisha, and M. L. Vaishali, "Quantum Computing."

[50] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemp. Phys.*, vol. 56, no. 2, pp. 172–185, Apr. 2015, doi: 10.1080/00107514.2014.964942.

[51] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine – Teori dan Aplikasinya dalam Bioinformatika," *Kuliah Umum IlmuKomputer.Com*, pp. 1303–1308, 2003, [Online]. Available: http://link.springer.com/10.1007/978-0-387-73003-5.

[52] F. Rachman and S. W. Purnami, "Perbandingan Klasifikasi Tingkat Keganasan Breast Cancer Dengan Menggunakan Regresi Logistik Ordinal Dan Support Vector Machine (SVM)," 2012.

[53] A. A. Abdillah and B. Prianto, "Pembelajaran Mesin Menggunakan Principal Component Analysis dan Support Vector Machines untuk Mendeteksi Diabetes," 2019, doi: 10.5614/jms.2019.24.1.2.

[54] H. Abdi and L. J. Williams, “Principal component analysis,” 2010, doi: 10.1002/wics.101.

[55] D. Groth, S. Hartmann, S. Klie, and J. Selbig, “Principal components analysis,” *Methods Mol. Biol.*, vol. 930, pp. 527–547, 2013, doi: 10.1007/978-1-62703-059-5_22.

[56] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman, “An Overview of Principal Component Analysis,” *J. Signal Inf. Process.*, vol. 04, no. 03, pp. 173–175, 2013, doi: 10.4236/jsip.2013.43b031.

[57] T. Carrier, P. Victor, A. Tekeoglu, and A. Lashkari, “Detecting Obfuscated Malware using Memory Feature Engineering,” Feb. 2022, pp. 177–188, doi: 10.5220/0010908200003120.

[58] M. Dener, G. Ok, and A. Orman, “Malware Detection Using Memory Analysis Data in Big Data Environment,” *Appl. Sci.*, vol. 12, no. 17, Sep. 2022, doi: 10.3390/app12178604.

[59] J. Romero, J. P. Olson, and A. Aspuru-Guzik, “Quantum autoencoders for efficient compression of quantum data,” *Quantum Sci. Technol.*, vol. 2, no. 4, Dec. 2017, doi: 10.1088/2058-9565/aa8072.