Implementasi Algoritma Kuantum Shor pada Platform IBM Quantum

Malik Abdurrasyid dan Wirawan
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: wirawan@ee.its.ac.id

***Abstrak*—Di era komputasi modern, komputasi kuantum telah muncul sebagai teknologi revolusioner dengan potensi untuk menyelesaikan masalah yang tidak dapat dipecahkan oleh komputer klasik dalam waktu yang cepat. Salah satu algoritma kuantum yang paling menonjol adalah Algoritma Shor, yang dapat memfaktorkan bilangan bulat menjadi faktor-faktor primanya secara eksponensial lebih cepat dibandingkan dengan algoritma klasik terbaik yang ada. Namun, implementasi Algoritma Shor dihadapkan pada berbagai tantangan, seperti keterbatasan algoritma klasik dalam pemfaktoran bilangan besar, potensi ancaman terhadap keamanan kriptografi modern, dan tantangan teknis dalam implementasi algoritma kuantum. Penelitian ini bertujuan untuk mengimplementasikan Algoritma Shor pada platform IBM Quantum untuk memverifikasi keefektifan dan efisiensinya dalam pemfaktoran bilangan bulat, serta untuk mengatasi tantangan teknis yang muncul selama proses implementasi. Evaluasi performa dilakukan dengan menguji algoritma pada berbagai bilangan bulat untuk mengukur kecepatan dan akurasi hasil pemfaktoran. Hasil yang telah didapatkan dapat ditunjukan bahwa kompleksitas dari algoritma shor yaitu $big\ O(n^3)$. Semakin besar nilai N yang ingin difaktorkan maka waktu yang dibutuhkan semakin lama. Didapatkan juga waktu rata – rata yang dibutuhkan algoritma shor untuk menemukan kunci yaitu sekitar 0.01 detik – 0.09 detik dan waktu algoritma faktorisasi klasik sekitar .0.001 detik – 0.008 detik. Perbedaan ini dikarenakan adanya tambahan program pada algoritma shor agar menghasilkan output pada excel.**

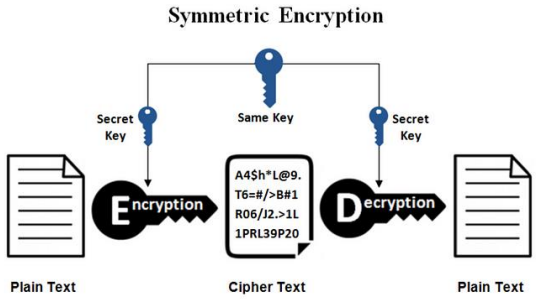
***Kata Kunci*—Komputasi Kuantum, Algoritma Shor, Pemfaktoran Bilangan Bulat, Sirkuit Kuantum, Kriptografi**

I. PENDAHULUAN

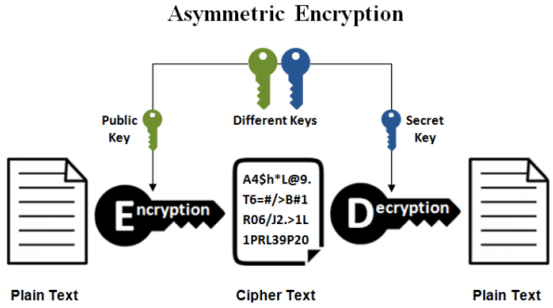
PADA era yang dipenuhi dengan peningkatan eksponensial dalam data dan transaksi online, menjaga keamanan data dalam dunia maya menjadi hal yang sangat penting. Oleh karena itu, sistem enkripsi berperan sebagai elemen yang krusial dalam hal ini. Cara yang digunakan untuk menyelesaikan masalah keamanan data adalah dengan menggunakan algoritma yang andal. Pada saat ini, terdapat berbagai algoritma enkripsi yang sulit di bobol secara pendekatan menggunakan komputer. Disisi lain terdapat penelitian yang menunjukan bahwa komputasi kuantum dapat memecahkan algoritma terbaik secara cepat. Komputasi kuantum diusulkan pada tahun 1980-an oleh Richard Feynman dan Yuri Manin.Komputasi kuantum berdasarkan dari fenomena mekanika kuantum berserta teknik superposisi dan keterkaitan. Pada tahun 1982, Richard Feynman mengusulkan bahwa menggunakan prinsip-prinsip kuantum dalam komputasi dapat mengatasi batasan fisik dalam mensimulasikan sistem fisik yang kompleks, seperti mekanika kuantum. Kemudian pada tahun 1985, David Deutsch memperkenalkan konsep mesin turing universal kuantum. Teori inilah yang menjadi dasar konseptual dari pengembangan komputer kuantum. Menurut teorema fundamental aritmatika, setiap bilangan bulat positif yang lebih besar dari satu dikatakan prima jika faktor positifnya hanya satu dan dirinya sendiri. Tetapi teorema tersebut tidak memberikan petunjuk mengenai cara mencarinya dan hanya membuktikan bahwa faktorisasi tersebut pasti ada. Sejauh ini, telah ada banyak algoritma yang dikembangkan untuk melakukan faktorisasi bilangan bulat. Namun sampai saat ini, faktorisasi bilangan bulat besar masih tetap menjadi masalah yang sulit bagi komputer konvensional untuk dipecahkan. Saat ini, algoritma konvensional tercepat untuk faktorisasi bilangan bulat yaitu *General Number Field Sieve* yang membutuhkan waktu sub-exponensial.

Pada tahun 1994, Peter Shor mencetuskan algoritma shor yang memiliki kemampuan untuk secara efisien memecahkan masalah faktorisasi dengan menggunakan komputasi kuantum sekaligus membuktikan potensi yang dimiliki komputasi kuantum dalam mengubah ketahanan keamanan pada bidang kriptografi. Dengan memanfaatkan sifat – sifat yang ada pada komputer kuantum, algoritma shor sanggup melakukan perhitungan faktorisasi bilangan bulat dalam waktu polinomial. Pada masa sebelumnya, perkembangan bidang tersebut bergantung pada penyelesaian permasalahan matematika yang rumit, seperti faktorisasi bilangan. Dalam sistem komputasi konvensional hanya memiliki dua kemungkinan keadaan yaitu 0 dan 1 sedangkan komputer kuantum terdiri dari *qubit* dimana masing - masing bit terdiri dari dua kemungkinan kondisi. Faktorisasi bilangan bulat merupakan proses memecah sebuah bilangan menjadi perkalian dari 2 bilangan prima. Sebagai contoh, faktorisasi dari 15 adalah 3 dan 5. Meskipun kelihatan sederhana untuk bilangan kecil, faktorisasi bilangan besar sangat sulit dan memakan waktu dengan menggunakan algoritma klasik. Saat ini, algoritma klasik terbaik untuk faktorisasi bilangan besar adalah sub-eksponensial dalam waktu berjalan, yang berarti waktu yang diperlukan untuk memfaktorkan bilangan besar meningkat sangat cepat seiring dengan meningkatnya ukuran bilangan.

Keamanan banyak sistem kriptografi modern, termasuk RSA (Rivest-Shamir-Adleman), bergantung pada kesulitan faktorisasi bilangan besar. RSA adalah salah satu algoritma kriptografi kunci publik yang paling banyak digunakan, dan keamanannya didasarkan pada asumsi bahwa faktorisasi bilangan besar (yang merupakan kunci publik) menjadi dua faktor prima besar (yang merupakan kunci privat) adalah masalah yang sangat sulit dan memakan waktu untuk dipecahkan dengan komputer klasik. Dengan algoritma klasik, tidak ada cara efisien untuk memfaktorkan bilangan besar dalam waktu polinomial. Algoritma faktorisasi klasik terbaik yang ada, seperti algoritma pengayakan kuadrat dan metode ECM (Elliptic Curve Method), masih memerlukan waktu eksponensial untuk memecahkan bilangan yang sangat



Gambar 2. Enkripsi simetris.



Gambar 3 Enkripsi asimetris.

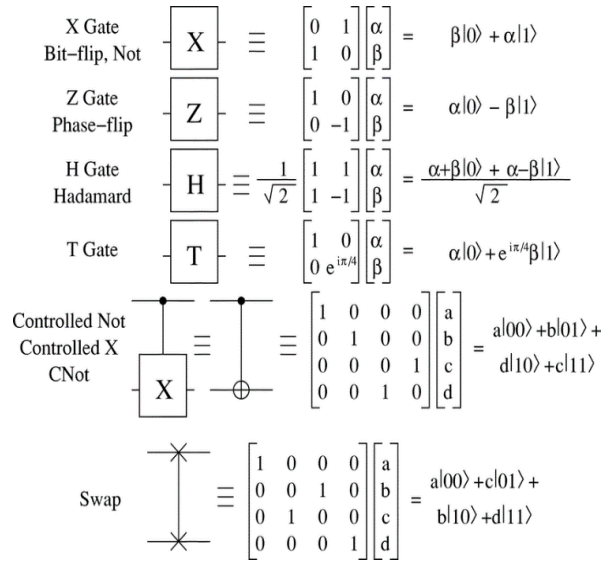
besar. Ini menciptakan kebutuhan untuk pendekatan baru yang dapat memecahkan masalah ini lebih efisien. Komputasi kuantum menawarkan paradigma baru dalam pemrosesan informasi, yang didasarkan pada prinsip-prinsip mekanika kuantum. Komputer kuantum dapat mengeksploitasi superposisi dan keterikatan (entanglement) untuk melakukan komputasi dengan cara yang tidak mungkin dilakukan oleh komputer klasik. Algoritma Shor menunjukkan bahwa komputer kuantum dapat memecahkan masalah faktorisasi bilangan bulat secara eksponensial lebih cepat dibandingkan dengan algoritma klasik terbaik.

II. URAIAN PENELITIAN

Enkripsi merupakan hal yang vital pada saat ini, dengan meningkatnya pertumbuhan data secara exponensial. Saat ini masih ada banyak algoritma enkripi yang sulit di bobol menggunakan komputer klasik. Tetapi terdapat penelitian yang menunjukan bahwa algoritma terbaik pada komputer klasik dapat dipecahkan oleh komputer kuantum dalam waktu yang sangat singkat [1]. Kriptografi kunci publik saat ini yang paling dikenal dan banyak digunakan adalah RSA. RSA merupakan kunci publik *cryptosistem* yang berdasarkan teori bilangan dan merupakan sistem cipher blok [2].

A. Kriptografi Kunci Publik

Sejak jaman dahulu, manusia sudah menggunakan berbagai teknik untuk menyembunyikan pesan – pesan rahasia agar tidak dapat dibaca oleh orang yang tidak memiliki akses yang sah. Walaupun tidak dianggap kriptografi modern tetapi teknik – teknik seperti penggeseran huruf atau pergantian huruf sudah digunakan untuk melindungi informasi. Kriptografi mengalami evolusi yang cukup signifikan dengan munculnya algoritma kriptografi simetris, seperti DES (data encryption standard) pada tahun 1970 – an. Kemudian pada tahun 1977, alogoritma asimetris pertama muncul yaitu RSA (rivest – shamir – adleman) yang menggunakan pasangan kunci privat dan kunci publik [2].



Gambar 1. Quantum logic gates.

Kriptografi diklasifikasikan menjadi 2 katergori utama yaitu enskripsi simetris dan asimetris. Enskripsi simetris merupakan metode enkripsi yang mana antara pengirim dan penerima menggunakan kunci yang sama untuk melakukan enkripsi maupun dekripsi. Hal ini tertera pada Gambar 1.

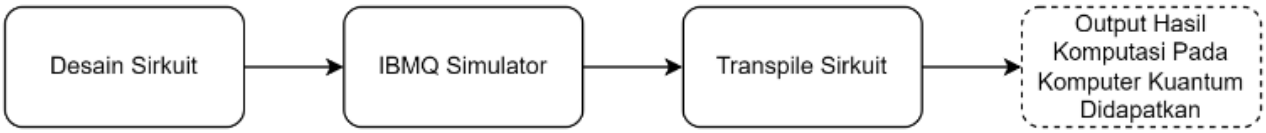
Berbeda halnya dengan enkripsi simetris, enkripsi asimetris merupakan kriptografi kunci publik yang menggunakan 2 pasang kunci yang disebut kunci publik dan kunci privat. Kunci publik digunakan untuk enkripsi pesan sedangkan kunci privat digunakan untuk menerjemahkan pesan [3]. Enskripsi asimetris tertera pada Gambar 2.

B. Algoritma RSA

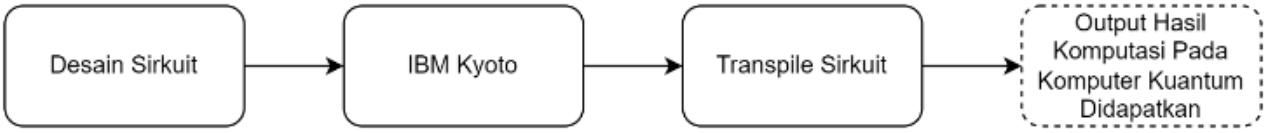
Algoritma RSA tergolong kedalam kriptografi dengan kunci asimetris atau kriptografi kunci publik yang didasarkan pada algoritma pertukaran kunci Diffie – Hellman. Pada algoritma RSA terdapat 3 proses dalam algoritma RSA yaitu pembangkitan kunci, enkripsi dan dekripsi. Algoritma RSA standar memiliki kelemahan keamanan yang berasal dari pemilihan dua bilangan prima saat menghasilkan kunci. Pada tahap ini, dilakukan operasi matematika berupa perkalian dua bilangan prima acak, p dan q, yang menghasilkan nilai n. Tingkat keamanan RSA dapat ditingkatkan dengan memilih bilangan prima yang lebih besar, yang akan membuat faktorisasi nilai n menjadi lebih sulit bagi penyerang. Semakin besar ukuran n, semakin tinggi tingkat keamanannya. Pemilihan bilangan prima yang besar juga membuat lebih sulit bagi penyerang untuk menemukan nilai faktor bilangan prima p dan q. Selain itu, untuk meningkatkan keamanan algoritma RSA, bisa dilakukan penambahan jumlah bilangan prima dalam pembangkitan kunci publik maupun kunci privat [4].

C. Komputer Kuantum

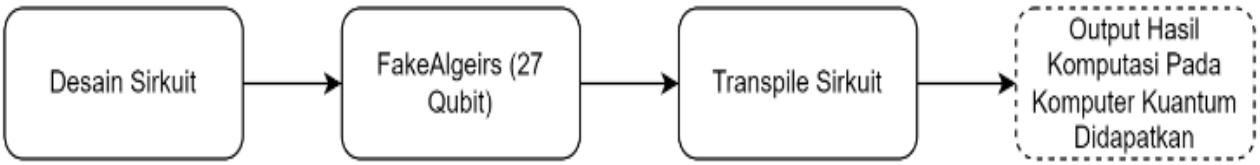
Komputer kuantum merupakan teknologi baru yang menggunakan prinsip fisika mekanika kuantum untuk menyelesaikan masalah yang tidak dapat diatasi oleh komputer klasik. Tujuan utama pengembangan komputer kuantum adalah untuk menangani jumlah data yang sangat besar dengan menerapkan prinsip-prinsip *entanglement* dan superposisi, *qubit* serta menjalankannya secara bersamaan. Teknologi ini tidak seperti komputer standar yang telah ada selama 50 tahun, algoritma kuantum mengambil perspektif



Gambar 4. blok diagram pengujian hasil penelitian menggunakan IBMQ simulator.



Gambar 5. Blok diagram pengujian hasil penelitian menggunakan IBM kyoto.



Gambar 6. Blok diagram pengujian hasil penelitian menggunakan fakealgeirs 27 qubit.

yang sesuai terhadap masalah-masalah ini dengan menciptakan ruang multidimensional di mana pola yang menghubungkan titik data individu muncul. Komputer klasik tidak dapat menemukan pola-pola ini karena tidak dapat menciptakan wilayah komputasi baru. Sebagai contoh nyata, komputer klasik tidak dapat menemukan pola lipatan dalam protein, sedangkan algoritma kuantum awal dapat menemukannya dengan cara yang lebih efektif dan jauh lebih cepat, tanpa memerlukan pemeriksaan yang memakan waktu seperti yang diperlukan oleh komputer klasik [5].

D. Komputer Klasik vs Komputer Kuantum

Bagian ini memperkenalkan secara fenomenologis perbedaan paling mendasar antara komputer klasik dan komputer kuantum :

1) Qubit

Qubit (Quantum Bit) merupakan satuan unit informasi pada komputer kuantum. Berbeda hal nya dengan bit yang merupakan satuan unit pada komputer klasik. Qubit dapat berada pada 0 atau 1 disaat yang bersamaan atau biasa disebut dengan superposisi. Pada komputer klasik, suatu infomasi atau data dikodekan dalam bit state yaitu 1 atau 0 sedangkan pada komputer kuantum, informasi dikodekan dalam unit qubit yang dapat berada dalam kondisi 1 atau 0 disaat yang bersamaan yang disebut superposisi. Gambar 2 merupakan qubit dalam keadaan superposisi yang direpresentasikan dalam bentuk bloch sphere dan memiliki vektor besar dan arah [6].

2) Keterkaitan Kuantum

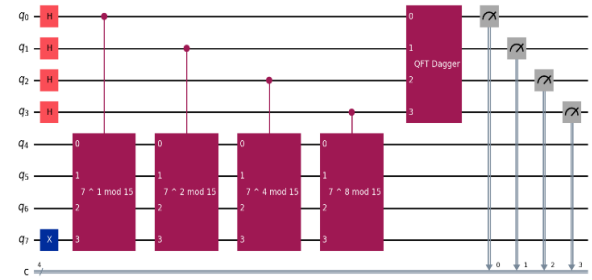
Teori tentang entanglement kuantum menjelaskan bahwa sistem partikel kuantum di setiap titik saling terkait dan tidak dapat dipisahkan. Jika keadaan kuantum dari sistem gabungan tidak dapat langsung diungkapkan sebagai hasil perkalian langsung dari keadaan kuantum dari kedua subsistem gabungan, maka keadaan murni dari sistem tersebut dapat berupa keadaan kuantum murni atau keadaan terkait. Teori ini juga diperluas ke dalam kasus keadaan campuran, di mana keadaan campuran dari seluruh sistem tidak sepenuhnya ditentukan, tetapi ada dalam bentuk probabilitas kuantum yang sesuai. Keadaan campuran ini dijelaskan oleh matriks densitas kuantum. Ketika keadaan campuran tidak dapat diungkapkan secara akurat sebagai berbagai bentuk keadaan langsung yang dapat diintegrasi, dan tidak satupun dari bentuk tersebut dapat dianggap sebagai keadaan terkait nonlinear, maka keadaan campuran tersebut disebut sebagai keadaan non-terkait. Ketika suatu subsistem gabungan tidak dapat memiliki tiga atau lebih subsistem, dan dua subsistem tidak dapat digabungkan dalam bentuk fungsi produk langsung dari masing-masing subsistem, keadaan murni atau campuran dari keadaan gabungan tersebut adalah keadaan terkait [7].

3) Reversibiltas Komputasi

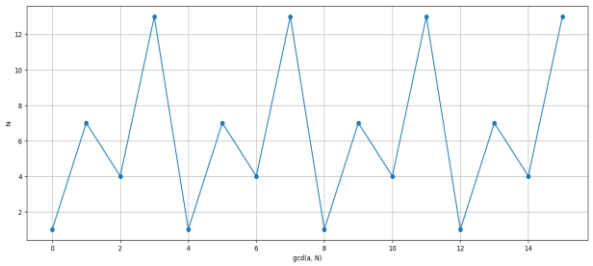
Masalah utama dalam miniaturisasi komputer klasik adalah disipasi panas, dan pendinginan konstan dari semua komponen diperlukan. Hal ini dicapai dengan menghubungkan termal sirkuit ke suatu reservoir panas seperti udara sekitarnya. Bagi komputer kuantum, pendinginan melalui koneksi panas bukanlah pilihan karena keadaan logisnya secara langsung direpresentasikan oleh keadaan kuantum umum dari registernya. Setiap koneksi panas akan menyebabkan entanglement dari keadaan ini dengan dunia luar dan menghancurkan kohesi perhitungan [8]. Hukum kedua termodinamika menyatakan bahwa setiap perubahan keadaan yang tidak dapat dibalikkan dari suatu sistem harus mendispersikan panas. Banyak operasi logis umum seperti AND, OR, atau mengatur ulang bit ke 0 atau 1 bersifat tidak dapat dibalikkan dalam arti bahwa input tidak dapat dihitung dari output. Oleh karena itu, operasi-operasi ini tidak dapat diimplementasikan secara langsung dalam komputer kuantum [8].

4) Gerbang Kuantum

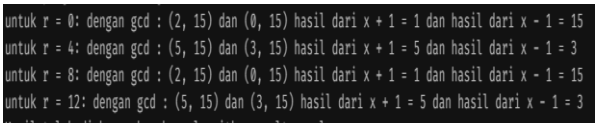
Quantum network adalah perangkat komputasi kuantum yang terdiri dari gerbang logika kuantum yang langkah-langkah komputasinya disinkronkan terhadap waktu. Output dari beberapa gerbang terhubung oleh kabel ke input gerbang lainnya. Ukuran jaringan ditentukan oleh jumlah gerbangnya. Ukuran input jaringan diatur oleh jumlah *qubit* input-nya, yaitu *qubit* yang disiapkan dengan tepat di awal setiap



Gambar 9. Quantum circuit 4 qubit.



Gambar 10. Hasil faktorisasi N = 15 dengan periode r = 4 pada komputer klasik.



Gambar 11. Hasil pengujian dengan N = 15 dan a = 7 dengan sirkuit 4 qubit pada fakealgiers provider.

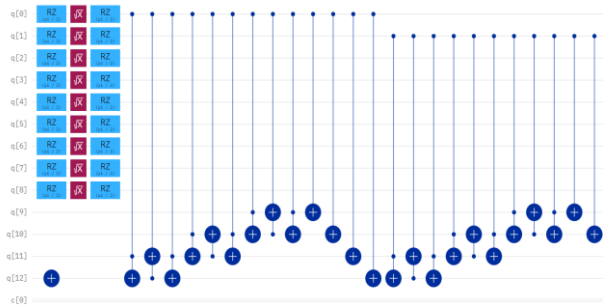
komputasi yang dilakukan oleh jaringan. Input dienkripsi dalam bentuk biner dalam basis komputasi dari *qubit* yang dipilih, sering disebut sebagai register kuantum, atau register. Seluruh komputasi kuantum ini adalah transformasi unitaris, di mana pengukuran dilakukan pada akhir untuk mengekstrak hasil. Hal ini akan dijelaskan lebih detail dalam subseksi berikutnya melalui beberapa contoh. Namun, transformasi unitaris sendiri adalah reversible; oleh karena itu, kita harus menggunakan gerbang reversible yang telah diperkenalkan sebelumnya agar dapat mengimplementasikan gerbang kuantum. Perbedaan antara komputasi klasik dan komputasi kuantum adalah bahwa gerbang kuantum beroperasi pada superposisi dari berbagai keadaan dasar *qubit*, sementara secara klasik hanya dapat berada dalam salah satu state saja yaitu 0 atau 1 [8]. Gambar 3 merupakan contoh gerbang logika kuantum.

E. Teori Bilangan Pada Algoritma Shor

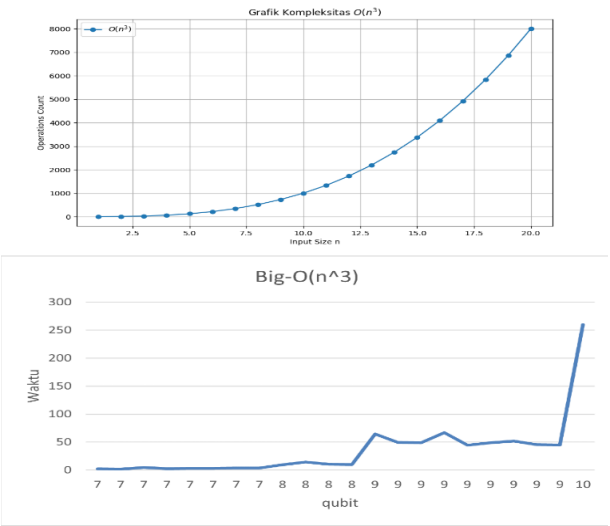
Teori bilangan merupakan cabang dari matematika murni yang berkaitan dengan sifat – sifat bilangan. Teori bilangan ditujukan untuk mempelajari bilangan bulat (interger) atau fungsi bernilai bilangan bulat. Bilangan bulat sendiri adalah bilangan yang tidak memiliki pecahan desimal. Contohnya 8, 22, 853, -35, 0. Bilangan bulat memiliki sifat pembagian misalnya a dan b adalah bilangan bulat, $a \neq 0$, a habis membagi b (a divided b) jika terdapat bilangan bulat c sedemikian sehingga b = a * c. Dapat dinotasikan sebagai $a \mid b$ jika $b = ac$, $c \in Z$ dan a != 0.

F. Algoritma Shor

Pada tahun 1978, Rivest, Shamir, dan Adleman mengembangkan algoritma kriptografi dengan memanfaatkan sifat satu arah dalam perkalian dua bilangan prima yang signifikan, biasanya melibatkan lebih dari 100



Gambar 7. Quantum circuit setelah di deploy pada IBMQ simulator.



Gambar 8. Kompleksitas waktu shor dalam memecahkan pesan.

digit desimal. Teknik ini dikenal sebagai algoritma RSA (diambil dari inisial penciptanya) dengan cepat menjadi sistem kunci publik yang paling populer dan penerapannya banyak dalam program komunikasi. Sementara itu, terdapat suatu keyakinan umum bahwa faktorisasi prima yang efisien pada komputer klasik dianggap tidak mungkin. Tetapi pada tahun 1994, algoritma yang efisien untuk komputer kuantum diusulkan oleh P.W. Shor. Algoritma shor didasarkan pada teori bilangan : fungsi $F(a) = x^a mod\ n$ merupakan fungsi periodik apabila nilai dari x saling relatif prima dengan n. Dalam algoritma shor, bilangan n merupakan bilangan bulat yang akan difaktorkan menjadi 2 bilangan prima. Apabila menghitung fungsi ini pada komputer klasik dengan jumlah eksponensial maka diperlukan waktu yang eksponensial juga. Oleh karena itu, masalah ini dapat diselesaikan dengan algoritma shor dengan memanfaatkan *pararellisme quantum* dengan cukup 1 langkah saja. Oleh karena $F(a)$ memiliki sifat periodik maka terdapat sebuah periode r. Sebagai contoh $x^0\ mod\ n = 1$ maka $x^r\ mod\ n = 1$ begitupun seterusnya. Dari informasi yang telah didapatkan maka persamaan dibawah ini dapat dimanipulasi.

$$X^r \equiv 1\ mod\ n \quad (1)$$

$$(X^{r/2})^2 \equiv 1\ mod\ n \quad (2)$$

$$(X^{r/2})^2 - 1 \equiv 0\ mod\ n \quad (3)$$

Dengan mengasumsikan bahwa r merupakan angka genap maka :

$$\left(x^{r/2} - 1\right)\left(x^{r/2} + 1\right) \equiv 0\ mod\ n \quad (4)$$

Tabel 1.
Perbandingan Waktu Algoritma

No	qubits	a	N	waktu (Shor)	waktu (ROT)	Av Time
1	4	7	15	0.036824	0.001003	0.035821
2	5	4	21	0.024782	0.002401	0.022381
3	5	8	21	0.048617	0.004117	0.044499
4	6	4	35	0.036332	0.00372	0.032611
5	6	9	55	0.045669	0.008557	0.037112
6	6	7	39	0.061318	0.001008	0.060310
7	7	12	77	0.067711	0.00657	0.061141
8	7	17	65	0.055255	0.002037	0.053218
9	7	8	91	0.064632	0.00214	0.062492
10	8	25	143	0.073385	0.001004	0.072381
11	8	14	221	0.077185	0.001993	0.075191
12	9	12	323	0.093215	0.002001	0.091214
13	9	12	437	0.075841	0.002002	0.073839
14	10	3	667	0.016338	0.001732	0.014606

Tugas dari algoritma Shor adalah menemukan periode dari fungsi periodik. Algoritma enkripsi yang didasarkan pada konsep teknik faktorisasi ini menggunakan operator *Quantum fourier transform*ation untuk pelaksanaan. Algoritma ini berfungsi baik dengan algoritma faktorisasi dan hingga saat ini dapat menemukan faktor dari persamaan hingga 21 *qubit*. Langkah-langkah implementasinya dimulai dengan menginisialisasi register dalam bentuk superposisi dari keadaan Q bersama dengan produk tensor. Dengan menggunakan keadaan tersebut, fungsi kuantum f(x) dibuat untuk register-input tersebut. Selanjutnya, dilakukan Inverse *Quantum fourier transformation*, diikuti dengan operasi-operasi irreduksi lainnya. Tugas utamanya adalah mengurangi masalah faktorisasi.

G. Transformasi Fourier Kuantum

Dalam komputasi kuantum, *Quantum fourier transform* (QFT) merupakan transformasi linear pada bit kuantum dan merupakan analogi kuantum dari transformasi Fourier diskrit. QFT dapat dilakukan dengan efisien pada komputer kuantum dengan mendekomposisi menjadi hasil perkalian matriks unitaris yang lebih sederhana. Transformasi Fourier diskrit pada 2^n amplitudo dapat diimplementasikan sebagai rangkaian kuantum yang terdiri dari hanya $O(n^2)$ gerbang Hadamard dan gerbang pergeseran fasa terkendali, di mana n adalah jumlah *qubit*. Hal ini dapat dibandingkan dengan transformasi Fourier diskrit klasikal yang memerlukan $O(n \times 2^n)$ gerbang (di mana n adalah jumlah bit), yang secara eksponensial lebih banyak daripada $O(n^2)$. QFT dapat dirumuskan sebagai berikut :

$$| x >= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{2^n}} |y > \quad (5)$$

$$y = \sum_{k=1}^{n} y_k \quad (6)$$

$$| x >= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x \sum_{k=1}^{n} y_k}{2^n}} |y > \quad (7)$$

$$|x >= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^{n} e^{\frac{2\pi i x y_k}{y_k}} |y > \quad (8)$$

H. Estimasi Fase Kuantum

Quantum Phase Estimation merupakan salah satu blok penting untuk berbagai algoritma kuantum. QPE bertujuan untuk menghitung eigenvalue dari operator unitary. Secara singkat estimasi phase dapat diartikan memperkirakan nilai dari ϕ eigenvalue. Pada algoritma shor penggunaan dari quantum phase estimation ini untuk mencari periode r menggunakan beberapa fungsi modulo [5]. QPE dapat dirumuskan sebagai berikut :

$$|x > |\omega > = |x > |\omega \oplus f_{a,N}(x) > \quad (9)$$

I. Transformasi Fourier Kuantum Invers

Jika Quantum Fourier Transform mengubah dari komputasional basis ke fourier basis agar dapat berada dalam kondisi superposisi, maka Quantum Fourier Transform Invers merupakan proses dimana mengubah dari fourier basis ke komputasional basis. Transformasi ini bertujuan untuk dapat menghitung probabilitas suatu bit dalam klasikal [5]. QFT^+dapat dirumuskan sebagai berikut :

$$QFT^+|x >= |x >= | x >= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{-2\pi i x y}{2^n}} |y > \quad (10)$$

J. Pengukuran

Dalam komputer klasik, deskripi formal dari *inner state* dan pengukuran keadaaan ini sama dan diberikan oleh nilai biner dari bit yang bersangkutan. Selain itu *inner state* tidak terpengaruh oleh proses pengukuran. Menurut interpretasi kopenhagner fisika kuantum, hasil pengukuran pada sistem kuantum bit harus dirumuskan dalam istilah klasik bit biner. Keadaan kuantum dari sistem tersebut dengan demikian berkurang. Jika bit pertama dalam keadaan 2 *qubit* yang disebutkan diatas diukur, dan pada nilai 1 diamati, maka keadaan akan berkurang menjadi $b'|1,0 > +d'|1,1 >$ dengan $|b'|^2 + |d'|^2 = 1$ sehingga semua basevektor yang 0 dalam bit pertama ($|0,0 >$ dan $|0,1 >$) akan diatur ke amplitudo 0. Oleh karena itu pada dasarnya tidak mungkin untuk mengukur keadaan dari register kuantum itu sendiri. Namun, memperkirakan nilai harapan dari *qubit* dapat dilakukan dengan pengukuran berulang setelah perhitungan yang sama [8]. langkah terakhir yaitu dilakukan pengukuran terhadap hasil tersebut. Pengukuran dapat dilakukan dengan metode modular exponentiation. Jika r = genap, maka :

$$x \equiv a^{r/2} \, (mod \, N) \quad (11)$$

$$\gcd\left(a^{r/2} \pm 1, N\right) = p \; dan \; q \quad (12)$$

III. HASIL PENELITIAN

Penelitian ini tidak hanya mengeksplorasi kemampuan algoritma shor dalam memfaktorkan suatu bilangan N yang

merupakan perkalian antara dua bilangan prima, tetapi juga mengevaluasi kinerja dan akurasi implementasi pada komputer kuantum. Pada bab ini akan diuraikan secara detail langkah – langkah implementasi algoritma shor, hasil eksperimen yang diperoleh dan analisis terhadap data yang didapatkan. Hasil penelitian yang telah dilakukan dapat dilihat melalui blok diagram pada Gambar 4.

Pada Gambar 4 merupakan blok diagram proses ketika pengujian dilakukan menggunakan IBMQ Simulator. Pada hasil pengujian ini, proses yang dilalui hanya sampai pada tahap transpile sirkuit. Meskipun sirkuit berhasil dieksekusi sampai tahap mendapatkan tranpile sirkuit tetapi output yang diharapkan tidak tercapai.

Pada Gambar 5 merupakan blok diagram proses ketika pengujian dilakukan menggunakan komputer IBM Kyoto. Pada hasil pengujian ini, proses yang dilalui hanya sampai pada tahap implementasi sirkuit pada IBM Kyoto sedangkan transpile sirkuit dan output yang diharapkan tidak tercapai.

Pada Gambar 6 merupakan blok diagram proses ketika pengujian dilakukan pada Komputer Kuantum lokal yaitu FakeAlgeirs. Proses implementasi algoritma shor menggunakan Fake Algeirs berhasil dilakukan hingga tahap pengumpulan output dan analisis data pengukuran untuk mendapatkan nilai p dan q. Dengan menggunakan perangkat simulasi lokal Fake Algeirs, sirkuit algoritma shor dapat diuji dan divalidasi untuk memastikan algoritma berjalan dengan benar dan memberikan hasil yang akurat dalam faktorisasi N.

A. Perencanaan Sirkuit Algoritma Shor

Dalam implementasi algoritma shor pada platform IBM Quantum, perencanaan sirkuit kuantum merupakan langkah yang sangat penting. Algoritma shor menggunakan prinsip dasar dari Quantum Fourier Transform dan modular eksponentiation untuk mencari periode suatu fungsi yang kemudian digunakan untuk faktorisasi. Ada beberapa tahapan yang perlu direncanakan sebelum membuat sirkuit kuantum yaitu sebagai berikut:

1. Pemilihan bilangan yang ingin difaktorkan
2. Pemilihan Bilangan Acak
3. Menentukan Register Kuantum dan Klasik

B. Desain Algoritma Shor menggunakan Python pada VSCode

Ketika dilakukan pengujian perlu diketahui terlebih dahulu nilai N yang ingin difaktorkan, jika ingin memfaktorkan 15, minimal register *qubit* yang dibutuhkan adalah 4 dan angka 15 tersebut merupakan perkalian antara dua bilangan prima. Sehingga dari Langkah – Langkah yang telah disebutkan diatas maka didapatkan hasil sirkuit kuantum seperti pada Gambar 7. Terdapat 4 proses dalam merancang sirkuit kuantum. Berdasarkan hasil pengujian $N = 15$ dan $qubit = 4$. berikut adalah 4 proses yang dilakukan untuk mewujudkan algoritma shor.

1) Transformasi Komputasional Basis menjadi Fourier Basis

4 *qubit* menandakan terdapat 4 register, yang dapat kita lihat pada Gambar 7 ditandai dengan simbol $q0, q1, q2\ dan\ q3$. Simbol ini merepresentasikan qubit 1, qubit 2, qubit 3 dan qubit 4. Kemudian, "H" direpresentasikan sebagai quantum fourier transform dimana "H" juga merupakan one qubit logic gate yaitu Hadamard Gate. Apabila persamaan 8, diterapkan pada $n = 1$ dan $x = 0$. Dimana $N = 2^n$ dan x merupakan register qubit sehingga :

$$QFT|0> = \frac{1}{\sqrt{2}}\left[(|0> + e^{\frac{2\pi i 0}{2^1}}|1>)\right]$$

$$QFT|0> = \frac{1}{\sqrt{2}}[(|0> + |1>)]$$

$$QFT|0> = |+>$$

2) Menemukan Periode

Pada proses kedua pada Gambar 7 terdapat sirkuit untuk melakukan *period finding* yang ditandai dengan $a^x mod\ N$. Dari persamaan 9 jika diterapkan pada sirkuit 4 *qubit*, $N = 15$ dan $a = 7$ maka sebagai berikut :

$$\begin{aligned}\frac{1}{4}[&|0>|7^0(mod\ 15)> + |1>|7^1(mod\ 15)> + |2\\ &> |7^2(mod\ 15)> + |3> |7^3(mod\ 15)\\ &> + |4> |7^4(mod\ 15)> + |5\\ &> |7^5(mod\ 15)> + |6> |7^6(mod\ 15)\\ &> + |7> |7^7(mod\ 15)> + |8\\ &> |7^8(mod\ 15)>] + |9> |7^9(mod\ 15)\\ &> + |10> |7^{10}(mod\ 15)> |11\\ &> |7^{11}(mod\ 15)> + |12\\ &> |7^{12}(mod\ 15)> + |13\\ &> |7^{13}(mod\ 15)> + |14\\ &> |7^{14}(mod\ 15)> + |15\\ &> |7^{15}(mod\ 15)>]\end{aligned}$$

Kemudian dari hasil diatas dilakukan penyederhanaan sehingga :

$$\begin{aligned}\frac{1}{4}[&|0>_4 |1>_4 + |1>_4 |7>_4 + |2>_4 |4>_4\\ &+ |3>_4 |13>_4 + |4>_4 |1>_4\\ &+ |5>_4 |7>_4 + |6>_4 |4>_4\\ &+ |7>_4 |13>_4 + |8>_4 |1>_4]\\ &+ |9>_4 |7>_4 + |10>_4 |4>_4\\ &+ |11>_4 |13>_4 + |12>_4 |1>_4\\ &+ |13>_4 |7>_4 + |14>_4 |4>_4\\ &+ |15>_4 |13>_4]\end{aligned}$$

Dari hasil perhitungan diatas didapatkan periode seperti pada Gambar 8. Pada Gambar 8 dimana angka yang ingin di faktorisasi yaitu N = 15 dan a merupakan angka yang dipilih secara acak dengan syarat $1 < a < N$ yaitu a = 7, menunjukan hasil periode r = 4. sumbu y merepresentasikan nilai N yaitu 0 - 15 dan sumbu x merepresentasikan periode dari hasil gcd $(a^x\ mod\ N)$.

3) Transformasi Fourier Basis ke Komputasional Basis

Setelah periode dari tiap register sudah dilakukan perhitungan, maka Langkah selanjutnya adalah dilakukan proses transformasi dari fourier basis ke komputasional basis. Apabila persamaan 10 diterapkan pada masing – masing register *qubit* yang telah dilakukan *period finding* maka diapatkan hasil sebagai berikut:

$$QFT^+|3>_4 = \frac{1}{\sqrt{16}}\sum_{y=0}^{16-1} e^{\frac{-2\pi i 3y}{16}}|y>$$

$$QFT^{+}|7>_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{16-1} e^{\frac{-2\pi i 7y}{16}} |y>$$

$$QFT^{+}|11>_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{16-1} e^{\frac{-2\pi i 11y}{16}} |y>$$

$$QFT^{+}|15>_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{16-1} e^{\frac{-2\pi i 15y}{16}} |y>$$

$$QFT^{+}|x>= \frac{1}{8} \sum_{y=0}^{15} \left[e^{-i\frac{3\pi}{8}y} + e^{-i\frac{7\pi}{8}y} + e^{-i\frac{11\pi}{8}y} + e^{-i\frac{15\pi}{8}y} \right] |y>$$

Kemudian menerapkan teorema euler pada hasil diatas, sehingga :

$$\frac{1}{8}[4|0>_4 + 4i\,|4>_4 - 4|8>_4 - 4i\,|12>_4]$$

Seperti yang dapat dilihat dari hasil diatas, amplitudenya terdapat pada $|x>= [0, 4, 8, 12]$ dengan periode antar bilangannya adalah $r = 4$

4) Pengukuran

Kemudian dari persamaan 11 dan 12 diterapkan pada hasil yang telah didapatkan dari Quantum Fourier Transform Invers, maka hasilnya seperti berikut:

$$7^{0}/_{2}\ (mod\ 15) = 1$$

$$7^{4}/_{2}\ (mod\ 15) = 4$$

$$7^{8}/_{2}\ (mod\ 15) = 1$$

$$7^{12}/_{2}\ (mod\ 15) = 4$$

Kemudian menggunakan persamaan 12 dari hasil nilai x diatas yang telah didapatkan, sehingga didapatkan :

$$gcd(1 \pm 1, 15) = 1, 15$$

$$gcd(4 \pm 1, 15) = 5, 3$$

$$gcd(1 \pm 1, 15) = 1, 15$$

$$gcd(4 \pm 1, 15) = 5, 3$$

Didapatkanlah hasil faktor dari N = 15 yaitu [1, 15] dan [5, 3]. Atau jika dari hasil pengujian yang dilakukan pada program yang telah dibuat maka output nya seperti pada Gambar 9.

C. Implementasi dan Analisis Kinerja Algoritma Shor pada Platform IBM Quantum

Pengujian algoritma shor ini dilakukan dengan menggunakan resource komputer kuantum yang telah disediakan oleh IBM Quantum. Pada percobaan pertama dilakukan pengujian dengan men-*deploy* sirkuit kuantum yang telah dibuat pada IBM Quantum dengan menggunakan IBMQ Simulator, IBM Kyoto dan FakeAlgeirs sebagai sistem yang digunakan. Hasil dari pengujian pada IBMQ Simulator tertera pada Gambar 10.

Gambar 10 merupakan hasil setelah komputer kuantum di *deploy* pada komputer kuantum dapat dilihat bahwa hasil sirkuit sudah tidak dalam bentuk seperti ketika dikembangkan pada VSCode seperti pada Gambar 7. Hasil yang didapatkan pada komputer kuantum IBM direpresentasikan sebagai *quantum logic gate*. Hasil pengujian dari IBM Kyoto hanya dapat tercapai sampai pada implementasi sirkuit pada system IBM Kyoto. Kemudian pengujian dilakukan dengan menggunakan *Fake Quantum Computer* (Fake Algiers 27 *Qubit*) yang telah disediakan oleh IBM untuk pengembangan secara lokal.

D. Memecahkan Pesan

Pengujian ini dilakukan untuk memecahkan pesan yang digenerate oleh algoritma RSA dengan membangkitkan pasangan kunci secara acak menggunakan fungsi *randint*. Hasil pengujiannya adalah sebagai berikut :

Kunci publik (e, N): (13, 899)

Kunci privat (d, N): (517, 899)

pesan asli : INI ADALAH PESAN KE 19 YANG AKAN DIPECAHKAN, DENGAN QUBIT = 11

Pesan terenkripsi: [114, 190, 114, 280, 489, 316, 489, 617, 489, 350, 280, 671, 50, 239, 489, 190, 280, 104, 50, 280, 857, 57, 280, 246, 489, 190, 328, 280, 489, 104, 489, 190, 280, 316, 114, 671, 50, 129, 489, 350, 104, 489, 190, 259, 280, 316, 50, 190, 328, 489, 190, 280, 169, 15, 250, 114, 416, 280, 309, 280, 857, 857]

Pesan terenkripsi dalam text : ['r', '¾', 'r', 'Ę', 'ǩ', 'ļ', 'ǩ', 'ı', 'ǩ', 'Ş', 'Ę', 'ʟ', '2', 'ï', 'ǩ', '¾', 'Ę', 'h', '2', 'Ę', '¸', '9', 'Ę', 'ö', 'ǩ', '¾', 'ň', 'Ę', 'ǩ', 'h', 'ǩ', '¾', 'Ę', 'ļ', 'r', 'ʟ', '2', '\x81', 'ǩ', 'Ş', 'h', 'ǩ', '¾', 'ă', 'Ę', 'ļ', '2', '¾', 'ň', 'ǩ', '¾', 'Ę', '©', '\x0f', 'ú', 'r', 'O', 'Ę', 'j', 'Ę', '¸', '¸']

Pesan terdekripsi: [73, 78, 73, 32, 65, 68, 65, 76, 65, 72, 32, 80, 69, 83, 65, 78, 32, 75, 69, 32, 49, 57, 32, 89, 65, 78, 71, 32, 65, 75, 65, 78, 32, 68, 73, 80, 69, 67, 65, 72, 75, 65, 78, 44, 32, 68, 69, 78, 71, 65, 78, 32, 81, 85, 66, 73, 84, 32, 61, 32, 49, 49]

Hasil akhir setelah di cracking : INI ADALAH PESAN KE 19 YANG AKAN DIPECAHKAN, DENGAN QUBIT = 11

Hasil dari pengujian tersebut didapatkan kompleksitas waktu $big\ O(n^3)$. Seperti pada Gambar 11. Seperti yang terlihat pada Gambar 11 grafik kompleksitas bahwa ketika nilai N dan qubit semakin besar maka dibutuhkan waktu yang semakin lama juga untuk memecahkan pesan yang digenerate oleh algoritma RSA.

E. Perbandingan Waktu Algoritma Kuantum Shor dengan Algoritma Faktorisasi Komputer Klasik

Dari percobaan yang telah dilakukan, waktu komputasi algoritma shor cenderung lebih lambat dikarenakan terdapat proses compile sirkuit pada komputer yang digunakan, sedangkan algoritma ROT komputasi nya lebih cepat beberapa mikrodetik. Seperti yang dapat dilihat pada Tabel 1.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab sebelumnya dapat disimpulkan bahwa algoritma shor menggunakan prinsip dasar fisika kuantum untuk mengubah komputasional basis menjadi forier basis, kemudian mencari periode menggunakan quantum phase estimation, dilakukan quantum fourier transform invers untuk mengembalikan lagi ke bentuk klasikal basis dan pengukuran

untuk mendapatkan probabilitas bit 0 dan bit 1. Kemudian dilakukan perhitungan menggunakan persamaan modular exponentiation $x \equiv a^{r}/_{2} \ (mod\ N)$ kemudian nilai x tersebut akan dilakukan gcd $(x \pm 1\ , N)$ untuk mendapatkan p dan q. Nilai dari p dan q kemudian digunakan pada algoritma RSA untuk mendapatkan kunci privat (d), yang kemudian kunci privat (d) tersebut digunakan untuk menerjemahkan kembali dari pesan yang terenkripsi menjadi pesan asli dengan persamaan $M = C^{d} mod\ N$.

Dalam pengimplementasian algoritma kuantum shor pada platform IBM kuantum diperlukan pembuatan sirkuit dengan menerapkan quantum fourier transform, quantum phase estimation, quantum fourier transform invers dan measurement pada sirkuit. Setelah itu, program menggunakan fungsi qiskit_ibm_runtime dengan menspesifikasikan sistem yang akan digunakan. Dari hasil penelitian yang telah dilakukan, sistem yang digunakan yaitu qasm simulator dan ibm Kyoto. Tetapi dikarenakan ketika dalam pengerjaan penelitian ini, library qiskit sedang dalam proses update dan migrasi ke versi 1.0 sehingga banyak kendala yang ditemukan dalam pengerjaannya dan juga tidak dapat menggunakan sistem tersebut secara maksimal. Kemudian pengujian dilanjutkan menggunakan Fake Quantum Provider FakeAlgeirs dengan kemampuan 27 Qubit.

Algoritma shor menggunakan prinsip fisika mekanika kuantum untuk mencari periode dari N, dari percobaan yang telah dilakukan dari 4 qubit sampai 10 qubit dengan beberapa variasi N dan a, didapatkan waktu rata – rata algoritma shor yaitu 0.01 detik - 0.09 detik. Sedangkan percobaan pada algoritma faktorisasi klasik membutuhkan waktu sekitar 0.001 detik – 0.008 detik. Waktu yang didapatkan relative tergantung pada background proses yang terjadi pada perangkat yang digunakanTuliskan kesimpulan dari penelitian yang artikelnya Anda tulis ini tanpa mengulang hal-hal yang telah disampaikan di Abstrak. Kesimpulan dapat diisi pula tentang pentingnya hasil yang dicapai dan saran untuk aplikasi dan pengembangannya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Direktorat Pendidikan Tinggi, Kementerian Pendidikan dan Kebudayaan Republik Indonesia atas dukungan finansial yang telah diberikan melalui Beasiswa KIP Kuliah selama periode 2020-2024. Beasiswa ini telah memberikan kesempatan berharga bagi penulis untuk menyelesaikan studi dengan lancar.

DAFTAR PUSTAKA

[1] G. Mounica, G. Manimaran, L. Jerome, and P. Bhattacharjee, “Implementation of 5-Qubit approach-based Shor’s Algorithm in IBM Qiskit,” in *IEEE Pune Section International Conference (PuneCon)*, 2021.

[2] X. Zhou and X. Tang, “Research and implementation of RSA algorithm for encryption and decryption,” in *6th International Forum on Strategic Technology*, 2011.

[3] D. Gautam, C. an Agrawal, P. Sharma, M. Mehta, and P. Saini, “Enhanced cipher technique using vigenere and modified caesar cipher,” in *International Conference on Trends in Electronics and Informatics (ICOEI)*, 2018, pp. 1--9.

[4] A. Aminudin, G. Aditya, and S. Arifianto, “RSA algorithm using key generator ESRKGS to encrypt chat messages with TCP/IP protocol,” *J. Teknol. dan Sist. Komput*, vol. 8, no. 2, pp. 113--130, 2020.

[5] A. Albuainain, J. Alansari, S. Alrashidi, W. Alqahtani, J. Alshaya, and N. Nagy, “Experimental implementation of shor’s quantum algorithm to break RSA,” in *International Conference on Computational Intelligence and Communication Networks (CICN)*, 2022, pp. 748--752.

[6] D. Ogi, F. Hutomo, and R. W. Wardhani, “Implementasi algoritme shor pada sirkuit kuantum untuk cracking algoritme RSA,” *Info Kripto*, vol. 16, no. 3, pp. 111--118, 2022.

[7] N. Zou, “Quantum entanglement and its application in quantum communication,” *Journal of Physics*, vol. 1827, no. 1, 2021.

[8] B. Omer, “Simulation of Quantum Computers,” Department of Theoretical Physics: Technical University of Vienna, 1996.