

Simulasi Dinamis *Single Qubit* dan *Multi Qubit*: Sebuah Pendekatan *Python*

<http://dx.doi.org/10.28932/jutisi.v11i2.10075>

Riwayat Artikel

Received: 01 Oktober 2024 | Final Revision: 21 Juli 2025 | Accepted: 21 Juli 2025



Creative Commons License 4.0 (CC BY – NC)

Muhammad Yusril Helmi Setyawan^{✉#1}, Nisa Hanum Harani^{*2}, Achmad Andriyanto^{#3}

[#] Program Studi D4 Teknik Informatika, Universitas Logistik & Bisnis Internasional

Jl. Sariastih 54, Bandung 40151, Indonesia

¹yusrilhelmi@ulbi.ac.id

²nisanuman@ulbi.ac.id

³ahmadandriyanto@ulbi.ac.id

[✉]Corresponding author: yusrilhelmi@ulbi.ac.id

Abstrak — Penelitian ini mengembangkan sistem simulasi dinamis untuk *single qubit* dan *multi qubit* menggunakan pendekatan *Python*, memanfaatkan pustaka komputasi kuantum seperti *Qiskit*, *NumPy*, dan *Matplotlib*. Sistem ini dirancang untuk mensimulasikan berbagai operasi kuantum, termasuk Hadamard, Pauli-X, Pauli-Y, Pauli-Z, Controlled-NOT(CNOT), dan Toffoli, dengan integrasi antarmuka web berbasis *Flask* untuk memudahkan interaksi pengguna. Hasil simulasi menunjukkan tingkat akurasi tinggi, dengan perbedaan hanya 0,2% pada probabilitas pengukuran untuk operasi *single qubit* seperti Hadamard dan kurang dari 0,4% untuk operasi *multi qubit* seperti CNOT dan Toffoli. Pengujian juga menunjukkan waktu eksekusi yang efisien, berkisar antara 12 hingga 25 milidetik, bahkan untuk operasi kuantum yang kompleks. Validasi terhadap literatur yang diakui menunjukkan bahwa sistem ini akurat dan efisien, serta dapat diandalkan untuk mendukung pembelajaran dan penelitian dalam komputasi kuantum.

Kata kunci — Gerbang Kuantum; *Multi Qubit*; *Python*; Simulasi Kuantum; *Single Qubit*.

Dynamic Simulation of Single Qubit and Multi Qubit: A Python Approach

Abstract — This study developed a dynamic simulation system for Single Qubit and multi qubit using a Python-based approach, leveraging quantum computing libraries such as Qiskit, NumPy, and Matplotlib. The system is designed to simulate various quantum operations, including Hadamard, Pauli-X, Pauli-Y, Pauli-Z, Controlled-NOT(CNOT), and Toffoli, with integration into a Flask-based web interface for easy user interaction. The simulation results show a high level of accuracy, with a difference of only 0.2% in measurement probabilities for Single Qubit operations like Hadamard and less than 0.4% for multi qubit operations like CNOT and Toffoli. The tests also demonstrated efficient execution times, ranging from 12 to 25 milliseconds, even for complex quantum operations. Validation against established literature confirms that the system is accurate, efficient, and reliable, making it a valuable tool for supporting learning and research in quantum computing.

Keywords — *Multi qubit*; *Python*; *Quantum Gates*; *Quantum Simulation*; *Single Qubit*.

I. PENDAHULUAN

Saat ini telah terjadi transisi paradigmatis yang signifikan dari komputasi klasik ke komputasi kuantum, yang menandai era baru dalam pemrosesan data dan analisis algoritma [1]. Komputasi kuantum tidak hanya menawarkan peningkatan eksponensial dalam kecepatan dan efisiensi pemrosesan data, tetapi juga membawa tantangan besar dalam pemahaman dan penerapan prinsip-prinsip kuantum [2]. Penggunaan *qubit*, yang mampu berada dalam keadaan superposisi, memberikan dasar untuk lompatan ini [3]. Namun, tantangan tersebut mengharuskan pengembangan alat simulasi yang intuitif dalam memvisualisasikan operasi kuantum. Python, dengan perpustakaannya yang kaya dan sintaks yang mudah dipahami, telah menjadi pilihan utama dalam pengembangan simulasi kuantum, terutama karena kemampuannya untuk menangani perhitungan numerik dan grafis secara efisien [4].

Komputasi kuantum memanfaatkan prinsip mekanika kuantum, seperti superposisi dan *entanglement*, untuk menyelesaikan masalah yang sulit bagi komputer klasik. Berbeda dengan bit klasik, *qubit* dapat berada dalam kombinasi $|0\rangle$ dan $|1\rangle$, memungkinkan pemrosesan paralel. Penerapannya mencakup algoritma *Shor* untuk faktorisasi bilangan prima, algoritma *Grover* untuk pencarian cepat, serta simulasi molekuler dalam kimia dan *material science*. Dengan potensinya, simulasi kuantum menjadi alat penting dalam memahami dan menguji prinsip komputasi kuantum sebelum diterapkan dalam perangkat keras nyata.

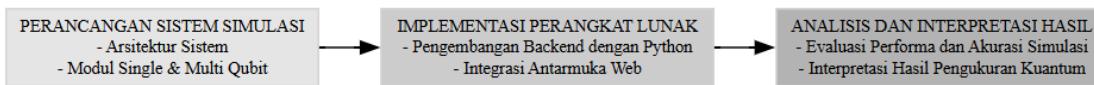
Penelitian sebelumnya telah mengidentifikasi simulasi sebagai kunci penyajian pembelajaran dan penelitian kuantum. Menurut Kohnle et al. (2014), simulasi yang memadukan visualisasi dan interaktivitas meningkatkan pemahaman konsep seperti *entanglement* dan superposisi [5]. Yao dan Xiang (2024) juga mendemonstrasikan bagaimana simulasi kuantum memperluas pemahaman tentang operasi *multi-qubit* dan memungkinkan eksperimen teoretis tanpa risiko [6]. Kedua studi ini menekankan pentingnya *software* yang fleksibel dan kuat seperti *python* dalam mengembangkan alat simulasi yang efektif.

Pemahaman konsep komputasi kuantum masih menjadi tantangan karena sifat abstraknya dan keterbatasan alat pembelajaran interaktif. Riset ini bertujuan untuk mengembangkan simulasi dinamis berbasis *python* yang tidak hanya memvisualisasikan tetapi juga memungkinkan manipulasi operasi-operasi kuantum pada *single qubit* dan *multi qubit* hingga batas tertentu, bergantung pada kapasitas memori dan metode simulasi yang digunakan. *Python* dipilih karena perpustakaan seperti *Qiskit*, *NumPy* dan *Matplotlib* yang mendukung perhitungan kompleks dan visualisasi data secara efektif. Simulasi ini dirancang sebagai alat pendukung pembelajaran untuk memfasilitasi pemahaman mendalam tentang fungsi gelombang, *superposisi*, *entanglement*, pengukuran kuantum, dan *transformasi unitary*, yang merupakan konsep inti dalam mekanika kuantum. Selain itu, sistem ini digunakan untuk memvalidasi model teoretis dalam komputasi kuantum dengan membandingkan hasil simulasi terhadap teori kuantum yang telah diterima, menguji distribusi probabilitas hasil pengukuran, serta memastikan bahwa setiap operasi gerbang kuantum berfungsi sesuai dengan prediksi matematis dan algoritma kuantum yang diketahui.

Simulasi ini diharapkan untuk mengisi kekosongan dalam materi pembelajaran kuantum dengan menyediakan alat yang dapat membantu mahasiswa dalam memvisualisasikan dan memahami konsep kuantum secara lebih efektif. Melalui pendekatan *python*, simulasi ini tidak hanya menawarkan interaktivitas dan aksesibilitas tetapi juga memberikan kesempatan bagi pengguna untuk berkontribusi dan memodifikasi simulasi, memperkuat pemahaman teoretis mereka dan mempersiapkan mereka untuk kemajuan di bidang kuantum yang terus berkembang.

II. METODE PENELITIAN

Metodologi penelitian ini dirancang menggunakan pendekatan kuantitatif dengan metode rekayasa perangkat lunak, mencakup perancangan, implementasi, pengujian, dan analisis hasil simulasi. *Python* diakui sebagai bahasa ideal dalam sains dan rekayasa karena sintaks sederhana dan dukungan perpustakaan seperti *SciPy*, *NumPy*, dan *Matplotlib*, yang memudahkan implementasi algoritma numerik dan visualisasi data. Dalam simulasi kuantum, *Python* semakin penting berkat alat seperti *Qiskit* yang memungkinkan perancangan dan pengujian sirkuit kuantum [7]. Simulasi kuantum membantu memahami konsep mekanika kuantum seperti superposisi dan *entanglement*, yang krusial untuk pengembangan algoritma kuantum [8]. *Entanglement*, salah satu fenomena utama dalam komputasi kuantum, memungkinkan korelasi kuat antar *qubit* meskipun berada pada jarak jauh, dan pemahaman lebih dalam mengenai hal ini sangat penting untuk pengembangan algoritma kuantum yang lebih efisien [9]. Simulasi juga berperan penting dalam pendidikan STEM, memperluas pemahaman mahasiswa tentang konsep kuantum [10]. Metodologi penelitian ini mengacu pada tahapan yang ditunjukkan pada gambar 1.



Gambar 1.Tahapan Penelitian

A. Perancangan Sistem Informasi

Tahap ini dimulai dengan perancangan arsitektur untuk membagi modul utama: simulasi *single qubit* dan *multi qubit*. Arsitektur mendukung integrasi dengan antarmuka web dan pengujian terpisah setiap modul.

1. *Modul Single Qubit*: Mengimplementasikan gerbang kuantum seperti Hadamard, Pauli-X, Pauli-Y, Pauli-Z, dan Rotasi. Visualisasi bola Bloch menggambarkan perubahan keadaan kuantum, perhitungan menggunakan *python* dengan *NumPy*, *Matplotlib* untuk visualisasi, dan *Qiskit* untuk simulasi.
2. *Modul Multi Qubit*: Mencakup gerbang kuantum seperti Hadamard, Pauli-X, Pauli-Y, Pauli-Z, CNOT, dan Toffoli (*Controlled-Controlled-NOT*, CCNOT). Transformasi kuantum untuk masing-masing gerbang dijelaskan sebagai berikut:

$$a. \text{ Hadamard } H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, X|0\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad (1)$$

$$b. \text{ Pauli-X } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, X|0\rangle = |1\rangle, X|1\rangle = |0\rangle \quad (2)$$

$$c. \text{ Pauli-Y } Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Y|0\rangle = i|1\rangle, Y|1\rangle = i|0\rangle \quad (3)$$

$$d. \text{ Pauli-Z } Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Z|0\rangle = |0\rangle, Z|1\rangle = |1\rangle \quad (4)$$

$$e. \text{ CNOT } CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, CNOT|00\rangle = |00\rangle, CNOT|01\rangle = |01\rangle, CNOT|10\rangle = |11\rangle, CNOT|11\rangle = |10\rangle \quad (5)$$

$$f. \text{ Toffoli } (CNOT) \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, CCNOT|110\rangle = |111\rangle, CCNOT|111\rangle = |110\rangle \quad (6)$$

B. Implementasi Perangkat Lunak

Python dipilih karena kemampuannya mengelola data numerik, grafik, serta dukungan pustaka seperti *Qiskit*, *NumPy*, dan *Matplotlib*, juga karena skalabilitas dan fleksibilitasnya untuk penelitian dan pendidikan. Pengembangan *backend* menggunakan *python* untuk memastikan akurasi perhitungan, efisiensi algoritma, dan pengelolaan memori optimal dalam simulasi. Sedangkan antarmuka web dibuat dengan HTML, CSS, dan *Flask*, yang memungkinkan pengguna mengakses simulasi melalui *browser*.

C. Pengujian dan Validasi

Pengujian dilakukan untuk memastikan perangkat lunak memenuhi spesifikasi fungsional dan non-fungsional, mencakup: uji unit (setiap fungsi diuji terpisah, fokus pada perhitungan aljabar linear dan visualisasi bola *Bloch*), uji integrasi (memastikan sinergi antara modul *backend* dan antarmuka web), uji sistem (mengevaluasi kinerja aplikasi dalam simulasi operasi *qubit*) dan validasi hasil (hasil simulasi dibandingkan dengan literatur kuantum untuk memastikan akurasi).

D. Pengujian dan Validasi

Setelah pengujian, data simulasi dianalisis untuk menilai efektivitas dan efisiensi sistem, termasuk waktu komputasi, akurasi visualisasi, dan kemudahan penggunaan antarmuka. Hasil pengukuran kuantum juga diinterpretasikan untuk memastikan sistem dapat merepresentasikan fenomena kuantum yang kompleks dengan benar.

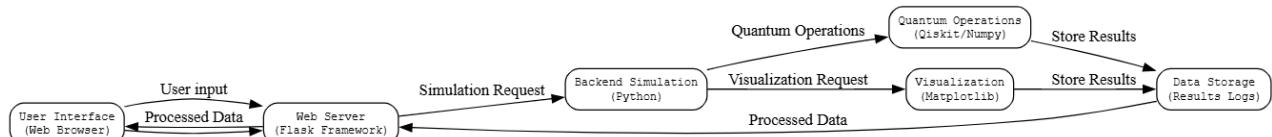
III. HASIL DAN PEMBAHASAN

Penelitian ini berhasil mengembangkan aplikasi simulasi kuantum berbasis *python* yang fokus pada simulasi dinamis *single qubit* dan *multi qubit*, menggunakan gerbang kuantum seperti Hadamard, Pauli-X, Pauli-Y, Pauli-Z, Rotasi, CNOT dan CCNOT (Toffoli). *Python* dipilih karena kemampuannya dalam komputasi numerik dan visualisasi grafis. Eksperimen menerapkan gerbang kuantum pada *qubit* dalam keadaan dasar (*initial state*), dengan hasil (*final state*) divisualisasikan menggunakan bola Bloch dan histogram. Adapun Perangkat lunak yang digunakan mencakup *Python* 3.11.9 (Bahasa pemrograman utama), *NumPy* (untuk operasi aljabar linear), *Matplotlib* (untuk visualisasi bola *Bloch* dan *histogram*), *Qiskit* 1.1.2 dan *Qiskit-aer* 0.14.2 (untuk simulasi operasi kuantum) serta *Flask* (untuk membangun antarmuka web interaktif).

A. Perancangan Sistem Simulasi

Perancangan sistem simulasi ini mengikuti arsitektur yang ditampilkan pada gambar 2, dengan fokus pada visualisasi dinamis dari *single qubit* dan *multi qubit*. Pengguna dapat berinteraksi melalui antarmuka (*user interface*) web yang memungkinkan mereka untuk memasukkan parameter simulasi, seperti jenis gerbang kuantum yang digunakan dan jumlah *qubit* yang dioperasikan. Simulasi ini kemudian menghasilkan visualisasi bola *Bloch* dan *histogram* yang menggambarkan keadaan kuantum secara *real-time*. Proses ini dirancang untuk memberikan respons cepat, sehingga pengguna dapat segera melihat efek dari setiap transformasi *qubit* yang dilakukan, tanpa keterlambatan yang signifikan dalam eksekusi komputasi.

Flask berperan sebagai *web server* yang menangani permintaan pengguna dan menghubungkannya ke *backend*. *Backend* dibangun menggunakan *python* dengan dukungan pustaka *Qiskit* dan *NumPy* untuk melakukan operasi kuantum dan perhitungan numerik yang kompleks. Hasil simulasi divisualisasikan menggunakan *Matplotlib* dan disimpan untuk keperluan analisis lebih lanjut serta *debugging*, mencakup hasil pengukuran, *log* operasional, dan data performa sistem. Integrasi ini memastikan bahwa sistem mudah digunakan dan memberikan hasil simulasi yang akurat dan cepat.



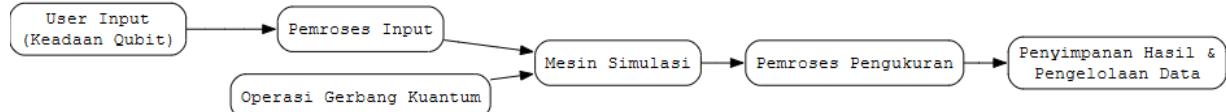
Gambar 2. Arsitektur Sistem

B. Implementasi Perangkat Lunak

Perangkat lunak untuk simulasi dinamis *single qubit* dan *multi qubit* diimplementasikan mencakup penggunaan bahasa pemrograman *python* dengan berbagai pustaka pendukung.

1) Pengembangan Backend

Backend adalah inti sistem simulasi kuantum yang menjalankan logika bisnis dan operasi kuantum. Bagian ini menerima input pengguna (*User Input*), menjalankan simulasi (*Simulation Request*), dan mengembalikan hasil yang dapat diolah (*Processed Data*) atau divisualisasikan (*Visualization Request*). Gambar 3 menunjukkan hubungan komponen *backend* dalam memproses input dan mengelola hasil simulasi.



Gambar 3. Alur Proses Pengembangan Backend

2) Integrasi dengan antarmuka web

Alur integrasi sebagaimana yang ditunjukkan pada Gambar 4, antara antarmuka web dan *backend* menggunakan *Flask* sebagai *web server* berfungsi menangani permintaan pengguna dan mengirim hasil simulasi. *API endpoint* menghubungkan antarmuka dengan *backend*, sementara *middleware* memproses serta memformat data sebelum ditampilkan. Antarmuka dirancang interaktif agar pengguna dapat mengatur konfigurasi, menjalankan simulasi, dan melihat hasil secara visual.

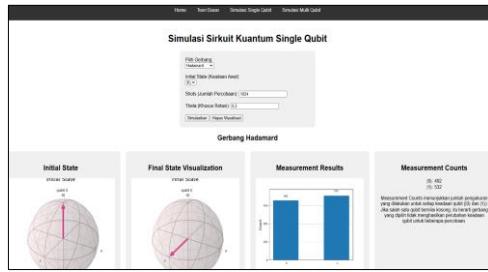


Gambar 4. Alur Integrasi dengan Antarmuka Web

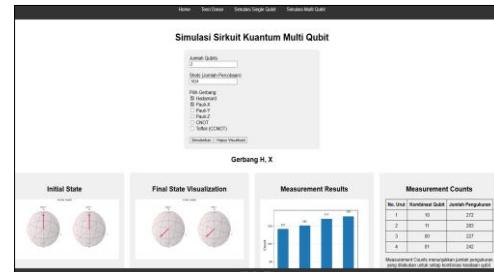
Terdapat empat menu utama dalam aplikasi, yaitu *Home* sebagai pengenalan, fitur Teori Dasar yang membahas konsep *qubit*, superposisi, *entanglement*, serta berbagai gerbang kuantum seperti Hadamard dan Pauli-X. Fitur Simulasi *single qubit* memungkinkan pengguna memahami perilaku *qubit* tunggal dan visualisasi *Bloch*, sedangkan fitur Simulasi *multi qubit* menyajikan simulasi sirkuit kuantum kompleks untuk mempelajari interaksi dan efek *entanglement* antar *qubit*.

Gambar 5 merupakan antarmuka Simulasi Sirkuit Kuantum *Single Qubit*. Pada bagian ini terdapat form inputan untuk melakukan simulasi. Form terdiri atas gerbang yang dipilih, *initial state* (keadaan awal), *shots* (jumlah percobaan), dan *theta* (khusus rotasi). Pengguna dapat mengisi form dengan lengkap, kemudian terdapat tombol “Simulasikan” untuk melakukan

simulasi dan tombol “Hapus Visualisasi” untuk menghilangkan hasil visualisasi sebelumnya dan melakukan simulasi yang baru. Hasil visualisasi dapat menunjukkan visual bola *bloch* keadaan awal dan keadaan keadaan akhir (*final*), serta hasil pengukuran (*measurement result*). Hasil simulasi juga dapat menampilkan perhitungan hasil (*measurement counts*) dan detail simulasi (*simulation details*).



Gambar 5. Antarmuka Modul *Single Qubit*



Gambar 6. Antarmuka Modul *Multi Qubit*

Pada Gambar 6 menunjukkan antarmuka pada halaman Simulasi Sirkuit Kuantum *Multi Qubit*. Pada halaman ini terdapat form inputan untuk melakukan simulasi. Form terdiri atas jumlah *qubit*, *shots* (jumlah percobaan), pemilihan gerbang yang dapat dipilih lebih dari satu gerbang (*multi*). Pengguna dapat mengisi form dengan lengkap, kemudian terdapat tombol “Simulasikan” untuk melakukan simulasi dan tombol “Hapus Visualisasi” untuk menghilangkan hasil visualisasi sebelumnya dan melakukan simulasi yang baru. Hasil simulasi dapat menentukan gerbang mana saja yang digunakan dan memberikan hasil visualisasi dalam bentuk gambar. Hasil visualisasi dapat menunjukkan visual bola *bloch* keadaan awal (*initial state*) dan keadaan keadaan akhir (*final state*), serta hasil pengukuran (*measurement result*). Hasil simulasi juga dapat menampilkan perhitungan hasil (*measurement counts*), detail simulasi (*simulation details*), serta hasil kombinasi *qubit* berdasarkan inputan pengguna.

C. Implementasi untuk Single Qubit

Implementasi simulasi *single qubit* memodelkan dan memvisualisasikan reaksi *qubit* terhadap gerbang kuantum seperti Hadamard, Pauli-X, Pauli-Y, Pauli-Z, dan Rotasi. Setiap gerbang diimplementasikan dalam *python* dengan pustaka *NumPy* untuk perhitungan matematis [11].

a) Gerbang Hadamard

Gerbang ini mengubah *qubit* dari keadaan dasar $|0\rangle$ menjadi superposisi $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$. Implementasi ini menggunakan matriks Hadamard yang dikalikan dengan vektor keadaan awal *qubit* :

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, state_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, result = H \times state_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (7)$$

```
import numpy as np
H = (1/np.sqrt(2)) * np.array([[1, 1], [1, -1]])
state_0 = np.array([[1], [0]]) # |0>
result = np.dot(H, state_0)
```

b) Gerbang Pauli-X, Pauli-Y, Pauli-Z

Pauli-X: Bertindak seperti gerbang NOT yang mengubah $|0\rangle$ menjadi $|1\rangle$ dan sebaliknya. Implementasi menggunakan matriks Pauli-X:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |1\rangle \quad (8)$$

```
X = np.array([[0, 1], [1, 0]])
result = np.dot(X, state_0) # Menghasilkan |1>
```

Pauli-Y: Melakukan rotasi *qubit* sebesar π radian terhadap sumbu y pada bola *bloch*, juga mengubah fase *qubit*

$$Y|0\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix} = i|1\rangle \quad (9)$$

```
Y = np.array([[0, -1j], [1j, 0]])
result = np.dot(Y, state_0) # Menghasilkan i|1>
```

Pauli-Z: Mengubah fase *qubit* ketika dioperasikan pada $|1\rangle$

$$|0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \quad (10)$$

```
Z = np.array([[1, 0], [0, -1]])
result = np.dot(Z, state_0) # Menghasilkan |0⟩, tanpa perubahan
```

c) Rotasi

Rotasi pada *qubit* dilakukan dengan menerapkan matriks rotasi untuk sumbu x, y, atau z dengan sudut tertentu (*theta*). Implementasi ini memungkinkan pengguna untuk mendefinisikan sudut rotasi yang diinginkan.

$$\theta = \frac{\pi}{4}, Rx(\theta) = \begin{bmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, state_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, result = Rx \times state_0 = \begin{bmatrix} \cos(\pi/8) \\ -i \sin(\pi/8) \end{bmatrix} \quad (11)$$

```
theta = np.pi/4 # Sudut rotasi
RX = np.array([[np.cos(theta/2), -1j*np.sin(theta/2)], [-1j*np.sin(theta/2), np.cos(theta/2)]])
result = np.dot(RX, state_0)
```

D. Implementasi untuk Multi Qubit

Implementasi simulasi untuk *multi qubit* lebih kompleks dibandingkan dengan *single qubit* karena melibatkan interaksi antar *qubit* secara simultan. Selain gerbang Hadamard, CNOT, dan Toffoli, sistem ini juga mendukung penerapan gerbang Pauli-X, Pauli-Y, dan Pauli-Z pada setiap *qubit* dalam sistem multi *qubit*.

a) Gerbang Hadamard Untuk Multi Qubit

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} H_{multi} = H \otimes I \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}, state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, result = H_{multi} \times state_{00} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{|00\rangle + |10\rangle}{\sqrt{2}} \quad (12)$$

```
H = (1/np.sqrt(2)) * np.array([[1, 1], [1, -1]])
I = np.eye(2) # Matriks identitas untuk qubit lainnya
H_multi = np.kron(H, I) # Menerapkan Hadamard pada qubit pertama dalam sistem dua qubit
state_00 = np.array([[1], [0], [0], [0]]) # |00⟩
result = np.dot(H_multi, state_00)
```

b) Gerbang Pauli-X, Pauli-Y dan Pauli-Z

Pauli-X (Gerbang X): Digunakan untuk membalik keadaan *qubit* tertentu dalam sistem *multi qubit*, mengubah $|0\rangle$ menjadi $|1\rangle$ dan sebaliknya. Untuk multi *qubit*, matriks Pauli-X diterapkan pada *qubit* tertentu menggunakan produk *Kronecker* [11]. Produk *Kronecker* merupakan operasi matematika yang digunakan untuk menghasilkan sebuah matriks yang lebih besar dari dua matriks input.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} X_{multi} = X \otimes I \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, result = X_{multi} \times state_{00} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = |10\rangle \quad (13)$$

```
X = np.array([[0, 1], [1, 0]])
I = np.eye(2)
X_multi = np.kron(X, I) # Menerapkan Pauli-X pada qubit pertama
state_10 = np.array([[0], [1], [0], [0]]) # |10⟩
result = np.dot(X_multi, state_10) # Menghasilkan |00⟩
```

Pauli-Y (Gerbang Y): Menggabungkan rotasi dan *flipping*, Pauli-Y mengubah keadaan kuantum dengan rotasi sekitar sumbu y. Implementasinya sama seperti Pauli-X, menggunakan produk *Kronecker* untuk mengaplikasikan operasi pada *qubit* tertentu [11].

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Y_{multi} = Y \otimes I \begin{bmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 1 & -i \\ 0 & 0 & i & 0 \end{bmatrix}, state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, result = Y_{multi} \times state_{00} = \begin{bmatrix} 0 \\ i \\ 0 \\ 0 \end{bmatrix} = i|10\rangle \quad (14)$$

```
Y = np.array([[0, -1j], [1j, 0]])
Y_multi = np.kron(Y, I) # Menerapkan Pauli-Y pada qubit pertama
result = np.dot(Y_multi, state_10) # Menghasilkan hasil yang dirotasi pada |10⟩
```

Pauli-Z (Gerbang Z): Gerbang ini mengubah fase dari *qubit* tertentu, terutama ketika diterapkan pada keadaan $|1\rangle$. Seperti gerbang lainnya, Pauli-Z dapat diterapkan pada *qubit* tertentu dalam sistem multi *qubit* [11].

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Z_{multi} = Z \otimes I \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, result = Z_{multi} \times state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |00\rangle \quad (15)$$

```
Z = np.array([[1, 0], [0, -1]])
Z_multi = np.kron(Z, I) #Menerapkan Pauli-Z pada qubit pertama
result = np.dot(Z_multi, state_10) # Menghasilkan fase perubahan pada |10)
```

c) Gerbang CNOT

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, result = CNOT \times state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |00\rangle \quad (16)$$

```
CNOT = np.array([[1, 0, 0, 0],
                 [0, 1, 0, 0],
                 [0, 0, 0, 1],
                 [0, 0, 1, 0]])
result = np.dot(CNOT, result)
```

d) Gerbang Toffoli (CCNOT)

Toffoli mengontrol dua *qubit* untuk mengubah *qubit* ketiga [11]. Ini merupakan bagian penting dari simulasi *multi qubit* yang lebih kompleks, terutama dalam konteks komputasi kuantum.

$$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, result = CCNOT \times state_{00} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |000\rangle \quad (17)$$

```
CCNOT = np.array([
    [1, 0, 0, 0, 0, 0, 0, 0],
    [0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1],
    [0, 0, 0, 0, 0, 0, 1, 0]
])
state_111 = np.array([[0], [0], [0], [0], [0], [0], [0], [1]]) #|111>
result = np.dot(CCNOT, state_111)
```

Hasil dari operasi *multi qubit* ini juga divisualisasikan dengan *Matplotlib*, memungkinkan pengguna untuk melihat perubahan dalam sistem *multi qubit* secara lebih komprehensif. Visualisasi ini menunjukkan bagaimana keadaan *qubit* terlibat dalam operasi seperti superposisi, dan bagaimana hasil pengukuran diinterpretasikan dalam konteks kuantum.

E. Pengujian dan Validasi

1) Uji Unit

Tabel 1 dan Tabel 2 adalah pengujian pertama dari uji unit yang dilakukan terhadap modul *single qubit* maupun *multi qubit* lalu dibandingkan kinerjanya

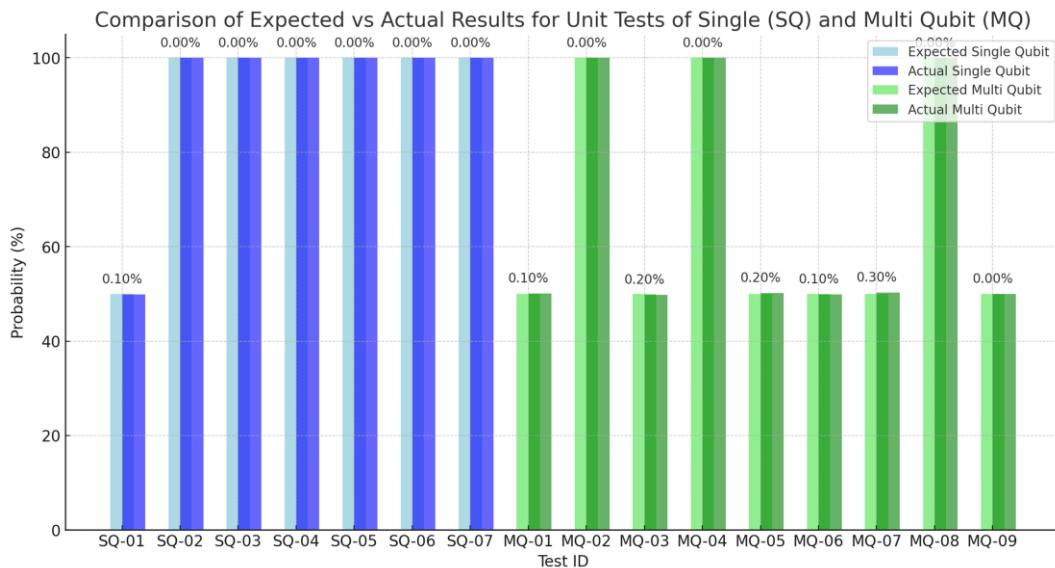
TABEL 1.
UJI UNIT UNTUK SINGLE QUBIT

Test ID	Fungsi Yang Diuji	Input	Output Yang Diharapkan	Output Aktual	Status
SQ-1	Hadamard	$ 0\rangle$	$\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$	$\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$	Simulasi berhasil menghasilkan superposisi sesuai yang diharapkan
SQ-2	Pauli-X	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	Simulasi berhasil mengubah keadaan ke $ 1\rangle$
SQ-3	Pauli-Y	$ 0\rangle$	$i 1\rangle$	$i 1\rangle$	Simulasi berhasil menghasilkan $i 1\rangle$
SQ-4	Pauli-Z	$ 1\rangle$	$- 1\rangle$	$- 1\rangle$	Simulasi berhasil mengubah fase <i>qubit</i> menjadi $- 1\rangle$

Test ID	Fungsi Yang Diuji	Input	Output Yang Diharapkan	Output Aktual	Status
SQ-5	Rotasi RX (90°)	$ 0\rangle$	$\frac{ 0\rangle + i 1\rangle}{\sqrt{2}}$	$\frac{ 0\rangle + i 1\rangle}{\sqrt{2}}$	Simulasi berhasil menghasilkan keadaan rotasi sesuai.
SQ-6	Rotasi RZ (45°)	$ 1\rangle$	$\frac{e^{\frac{i\pi}{4}} 1\rangle}{\sqrt{2}}$	$\frac{e^{\frac{i\pi}{4}} 1\rangle}{\sqrt{2}}$	Simulasi berhasil mengubah fase <i>qubit</i> sesuai dengan rotasi yang diterapkan
SQ-7	Pengukuran Probabilitas	$\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$	Probabilitas $ 0\rangle = 0.5, 1\rangle$	Probabilitas $ 0\rangle = 0.5, 1\rangle$	Simulasi berhasil menghasilkan distribusi probabilitas yang sesuai

TABEL 2.
UJI UNIT UNTUK *MULTI QUBIT*

Test ID	Fungsi Yang Diuji	Input	Output Yang Diharapkan	Output Aktual	Status
MQ-1	Hadamard pada Qubit 1	$ 00\rangle$	$\frac{ 00\rangle + 10\rangle}{\sqrt{2}}$	$\frac{ 0\rangle + 10\rangle}{\sqrt{2}}$	Simulasi berhasil menghasilkan superposisi sesuai yang diharapkan
MQ-2	Pauli-X pada Qubit 2	$ 01\rangle$	$ 11\rangle$	$ 11\rangle$	Simulasi berhasil mengubah keadaan ke $ 11\rangle$
MQ-3	Pauli-Y pada Qubit 1	$ 10\rangle$	$i 11\rangle$	$i 11\rangle$	Simulasi berhasil menghasilkan keadaan $i 11\rangle$
MQ-4	Pauli-Z pada Qubit 2	$ 11\rangle$	$- 11\rangle$	$- 11\rangle$	Simulasi berhasil mengubah fase <i>qubit</i> menjadi $- 11\rangle$
MQ-5	CNOT Q1→Q2	$ 10\rangle$	$ 11\rangle$ (qubit pertama sebagai kontrol dan <i>qubit</i> kedua sebagai target dalam keadaan $ 10\rangle$)	$ 11\rangle$	Simulasi berhasil mengubah keadaan menjadi $ 11\rangle$.
MQ-6	Toffoli (CCNOT)	$ 111\rangle$	$ 110\rangle$ (mengubah qubit ketiga karena kedua <i>qubit</i> kontrol dalam keadaan $ 11\rangle$)	$ 110\rangle$	Simulasi berhasil mengubah keadaan menjadi $ 110\rangle$
MQ-7	Pengukuran Probabilitas	$\frac{ 00\rangle + 10\rangle}{\sqrt{2}}$	Probabilitas $ 00\rangle = 0.5, 11\rangle = 0.5$	Probabilitas $ 00\rangle = 0.5, 11\rangle = 0.5$	Simulasi berhasil menghasilkan distribusi probabilitas yang sesuai.
MQ-8	Rotasi RX (90°) pada Qubit 1	$ 00\rangle$	$\frac{ 00\rangle + i 10\rangle}{\sqrt{2}}$	$\frac{ 00\rangle + i 10\rangle}{\sqrt{2}}$	Simulasi berhasil menghasilkan keadaan rotasi sesuai
MQ-9	Gabungan Gerbang (Hadamard pada qubit pertama dan kemudian CNOT dengan qubit pertama sebagai kontrol)	$ 00\rangle$	Setelah Hadamard : $\frac{ 00\rangle + 10\rangle}{\sqrt{2}}$ Setelah CNOT: $\frac{ 00\rangle + 11\rangle}{\sqrt{2}}$	Setelah Hadamard : $\frac{ 00\rangle + 10\rangle}{\sqrt{2}}$ Setelah CNOT: $\frac{ 00\rangle + 11\rangle}{\sqrt{2}}$	Simulasi berhasil menghasilkan keadaan $ 00\rangle + 11\rangle$



Gambar 7. Grafik Perbandingan hasil Uji Unit

Pada Gambar 7, pengujian unit *single qubit* (SQ-01 hingga SQ-07) menunjukkan perbedaan kecil antara hasil aktual dan yang diharapkan, dengan contoh pada SQ-01 (Hadamard) hanya 0,1%. Ini menunjukkan akurasi tinggi. Pengujian *multi qubit* (MQ-01 hingga MQ-09) juga menunjukkan kesesuaian, dengan perbedaan terbesar sekitar 0,2% hingga 0,3%, seperti pada MQ-03 (CNOT) yang mencatat 49,8% dibandingkan 50% teoretis.

2) Uji Integrasi dan Uji Sistem

Uji integrasi yang ditunjukkan pada Tabel 3 dan Tabel 4 dilakukan untuk memastikan modul-modul berfungsi dengan baik

TABEL 3.
UJI INTEGRASI UNTUK SINGLE QUBIT

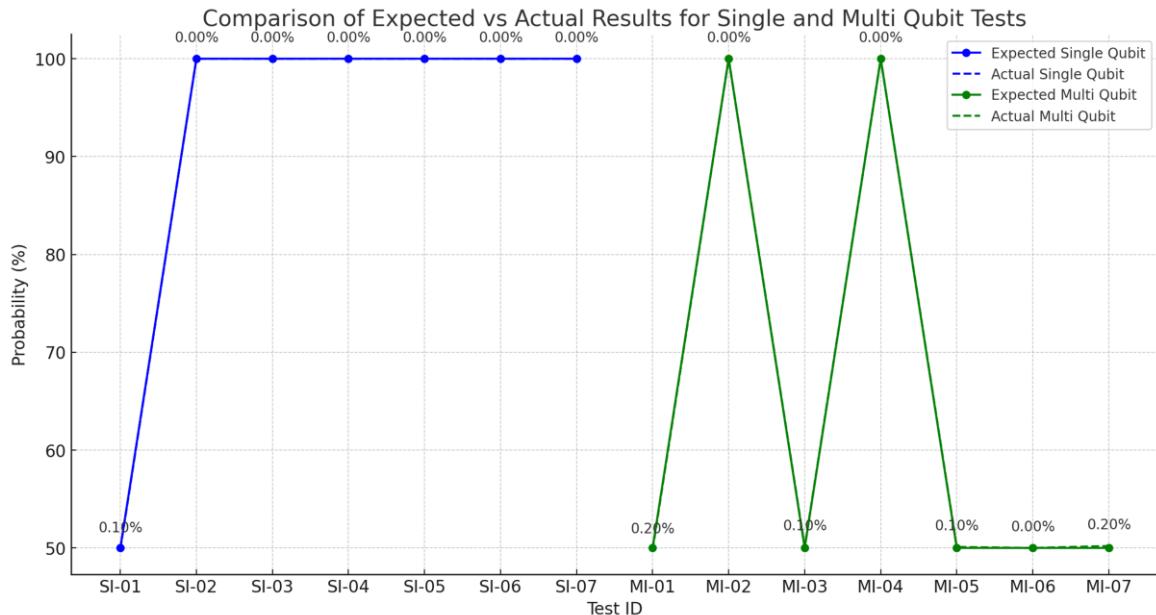
Test ID	Modul Yang Diintegrasikan	Input	Proses	Output Yang Diharapkan	Output Aktual	Status
SI-1	Hadamard + Pengukuran	$ 0\rangle$	Hadamard diterapkan pada <i>qubit</i> yang berada dalam keadaan $ 0\rangle$. Hadamard akan menempatkan <i>qubit</i> dalam keadaan superposisi, di mana <i>qubit</i> memiliki probabilitas yang sama untuk berada dalam keadaan $ 0\rangle$ dan $ 1\rangle$ setelah pengukuran	Probabilitas untuk <i>qubit</i> berada dalam keadaan $ 0\rangle$ adalah 50%, dan dalam keadaan $ 1\rangle$ juga 50%.	Hasil simulasi menunjukkan probabilitas $ 0\rangle=49.9%$ dan $ 1\rangle=50.1%$ yang sangat mendekati nilai teoretis	Hasil ini menunjukkan bahwa modul Hadamard dan pengukuran berfungsi sesuai dengan ekspektasi teoretis.
SI-2	Pauli-X + Pengukuran	$ 0\rangle$	Pauli-X yang berfungsi sebagai gerbang NOT dalam kuantum, diterapkan pada <i>qubit</i> dalam keadaan $ 0\rangle$. Pauli-X mengubah keadaan $ 0\rangle$ menjadi $ 1\rangle$.	Setelah pengukuran, <i>qubit</i> seharusnya berada dalam keadaan $ 1\rangle$ dengan probabilitas 100%.	Hasil simulasi menunjukkan probabilitas $ 1\rangle=100%$ sesuai dengan yang diharapkan.	Berhasil. Ini mengindikasikan bahwa Pauli-X diterapkan dengan benar.
SI-3	Pauli-Y + Pengukuran	$ 0\rangle$	Pauli-Y diterapkan pada <i>qubit</i> yang berada dalam keadaan $ 0\rangle$. Pauli-Y tidak hanya mengubah keadaan <i>qubit</i> seperti Pauli-X, tetapi juga menambahkan rotasi fase i , menghasilkan keadaan $i 1\rangle$	<i>Qubit</i> berada dalam keadaan $i 1\rangle$ setelah operasi.	Hasil simulasi menunjukkan <i>qubit</i> berada dalam keadaan $i 1\rangle$, sesuai dengan teori.	Modul Pauli-Y dan pengukuran berfungsi seperti yang diharapkan

Test ID	Modul Yang Diintegrasikan	Input	Proses	Output Yang Diharapkan	Output Aktual	Status
SI-4	Pauli-Z + Pengukuran	$ 1\rangle$	Pauli-Z diterapkan pada <i>qubit</i> yang berada dalam keadaan $ 1\rangle$. Pauli-Z mengubah fase <i>qubit</i> dari positif menjadi negatif, tanpa mengubah probabilitas amplitudonya, sehingga mengubah keadaan menjadi $- 1\rangle$	Setelah operasi, <i>qubit</i> berada dalam keadaan $- 1\rangle$	Hasil simulasi menunjukkan <i>qubit</i> berada dalam keadaan $- 1\rangle$, sesuai dengan teori.	Gerbang Pauli-Z berfungsi sesuai dengan harapan
SI-5	Rotasi RX (90°) + Pengukuran	$ 0\rangle$	Rotasi RX sebesar 90° diterapkan pada <i>qubit</i> dalam keadaan $ 0\rangle$. Operasi rotasi ini mengubah <i>qubit</i> ke dalam keadaan $\frac{ 0\rangle+i 1\rangle}{\sqrt{2}}$, menghasilkan superposisi dengan fase i .	<i>Qubit</i> seharusnya berada dalam superposisi $\frac{ 0\rangle+i 1\rangle}{\sqrt{2}}$	Hasil simulasi sesuai dengan teori, menunjukkan <i>qubit</i> berada dalam superposisi tersebut	Rotasi RX (90°) berhasil diterapkan sesuai dengan ekspektasi.
SI-6	Hadamard + Pauli-Z + Pengukuran	$ 0\rangle$	Hadamard diterapkan pada <i>qubit</i> dalam keadaan $ 0\rangle$, diikuti oleh Pauli-Z. Hadamard menempatkan <i>qubit</i> dalam superposisi $\frac{ 0\rangle+ 1\rangle}{\sqrt{2}}$ kemudian Pauli-Z mengubah fase <i>qubit</i> , menghasilkan keadaan $\frac{ 0\rangle- 1\rangle}{\sqrt{2}}$	Keadaan akhir <i>qubit</i> seharusnya $\frac{ 0\rangle- 1\rangle}{\sqrt{2}}$	Hasil simulasi sesuai dengan teori, menunjukkan <i>qubit</i> berada dalam keadaan $\frac{ 0\rangle- 1\rangle}{\sqrt{2}}$	Kombinasi Hadamard dan Pauli-Z berfungsi dengan benar
SI-7	Hadamard + Visualisasi Bola Bloch	$ 0\rangle$	Hadamard diterapkan pada <i>qubit</i> dalam keadaan $ 0\rangle$, mengubahnya ke dalam keadaan superposisi. Hasil dari operasi ini kemudian divisualisasikan menggunakan Bola Bloch untuk menunjukkan posisi vektor keadaan <i>qubit</i> .	Visualisasi Bola Bloch menunjukkan <i>qubit</i> dalam keadaan superposisi.	Visualisasi menunjukkan bahwa <i>qubit</i> berada dalam keadaan superposisi, dengan vektor yang ditunjukkan pada permukaan Bola Bloch sesuai dengan ekspektasi teoretis.	Visualisasi Bola Bloch berhasil menampilkan hasil sesuai dengan perubahan yang dilakukan oleh gerbang Hadamard

TABEL 4.
UJI INTEGRASI UNTUK *MULTI QUBIT*

Test ID	Modul Yang Diintegrasikan	Input	Proses	Output Yang Diharapkan	Output Aktual	Status
MI-1	Hadamard (Q1) + CNOT (Q1→Q2) + Pengukuran	$ 00\rangle$	Hadamard diterapkan pada Q1, mengubahnya menjadi superposisi $\frac{ 0\rangle+ 1\rangle}{\sqrt{2}}$. Kemudian, CNOT diterapkan, yang menghasilkan keadaan $\frac{ 00\rangle+ 11\rangle}{\sqrt{2}}$, menempatkan kedua <i>qubit</i> dalam keadaan entangled.	Keadaan akhir seharusnya $\frac{ 00\rangle+ 11\rangle}{\sqrt{2}}$ dengan probabilitas 50% untuk $ 00\rangle$ dan 50% untuk $ 11\rangle$.	Hasil simulasi menunjukkan probabilitas $ 00\rangle=49.8\%$ dan $ 11\rangle=50.2\%$ yang sangat dekat dengan nilai teoretis.	Berhasil. Modul Hadamard dan CNOT berfungsi sesuai dengan ekspektasi, dan simulasi menghasilkan entanglement dengan akurasi baik.
MI-2	Pauli-X (Q2) + CNOT (Q1 = $ 0\rangle$, Q2 = $ 1\rangle$) + Pengukuran	$ 01\rangle$	Pauli-X diterapkan pada Q2, mengubahnya dari $ 1\rangle$ menjadi $ 0\rangle$. Kemudian, CNOT diterapkan, yang mengubah keadaan menjadi $ 11\rangle$ ketika $Q1 = 1\rangle$.	Keadaan akhir seharusnya $ 11\rangle$ dengan probabilitas 100%.	Hasil simulasi menunjukkan probabilitas $ 11\rangle=100\%$ sesuai dengan nilai teoretis.	Berhasil. Modul Pauli-X dan CNOT berfungsi dengan benar, menghasilkan hasil yang diharapkan.
MI-3	Hadamard (Q1) + Pauli-Z (Q2) + Pengukuran	$ 00\rangle$	Hadamard diterapkan pada Q1, menghasilkan superposisi $\frac{ 0\rangle+ 1\rangle}{\sqrt{2}}$. Pauli-Z kemudian diterapkan pada Q2, mengubah fase tanpa	Keadaan akhir seharusnya $\frac{ 00\rangle- 10\rangle}{\sqrt{2}}$ dengan	Hasil simulasi menunjukkan probabilitas $ 00\rangle=50.1\%$ dan $ 10\rangle=49.9\%$ yang	Berhasil. Kombinasi Hadamard dan Pauli-Z berfungsi dengan benar, menghasilkan hasil yang diharapkan.

Test ID	Modul Yang Diintegrasikan	Input	Proses	Output Yang Diharapkan	Output Aktual	Status
			mengubah amplitudo, menghasilkan keadaan $\frac{ 00\rangle - 10\rangle}{\sqrt{2}}$	probabilitas 50% untuk $ 00\rangle$ dan 50% untuk $ 10\rangle$.	sangat dekat dengan nilai teoretis.	
MI - 4	Toffoli (Q1, Q2 → Q3) + Pengukuran	$ 111\rangle$ (Q1 = $ 1\rangle$ Q2 = $ 1\rangle$ Q3 = $ 1\rangle$)	Toffoli diterapkan, yang mengubah keadaan Q3 dari $ 1\rangle$ menjadi $ 0\rangle$ hanya jika kedua qubit kontrol (Q1 dan Q2) berada dalam keadaan $ 1\rangle$.	Keadaan akhir seharusnya $ 110\rangle$ dengan probabilitas 100%	Hasil simulasi menunjukkan probabilitas $ 110\rangle = 100\%$, sesuai dengan nilai teoretis.	Gerbang Toffoli berhasil diterapkan dengan hasil yang akurat
MI - 5	Rotasi RX (Q1) + CNOT (Q1 → Q2) + Pengukuran	$ 00\rangle$	Rotasi RX sebesar 90° diterapkan pada Q1, mengubahnya menjadi superposisi $\frac{ 0\rangle + i 1\rangle}{\sqrt{2}}$. Kemudian, CNOT diterapkan, yang menghasilkan keadaan $\frac{ 00\rangle + i 11\rangle}{\sqrt{2}}$	Keadaan akhir seharusnya $\frac{ 00\rangle + i 11\rangle}{\sqrt{2}}$, dengan distribusi probabilitas yang sesuai.	Hasil simulasi menunjukkan distribusi probabilitas sesuai dengan hasil yang diharapkan.	Rotasi RX dan gerbang CNOT berhasil diterapkan dengan hasil yang akurat.
MI - 6	Hadamard (Q1) + Pauli-Y (Q2) + Pengukuran	$ 00\rangle$	Hadamard diterapkan pada Q1, menghasilkan superposisi $\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$. Pauli-Y gate kemudian diterapkan pada Q2, menambahkan rotasi fase i , menghasilkan keadaan $\frac{ 00\rangle + i 10\rangle}{\sqrt{2}}$	Keadaan akhir seharusnya $\frac{ 00\rangle + i 10\rangle}{\sqrt{2}}$, dengan probabilitas 50% untuk $ 00\rangle$ dan 50% untuk $ 10\rangle$.	Hasil simulasi menunjukkan probabilitas $ 00\rangle = 50.0\%$ dan $ 10\rangle = 50.0\%$, ini sesuai dengan nilai teoretis.	Kombinasi Hadamard dan Pauli-Y berfungsi dengan baik, menghasilkan hasil yang akurat.
MI - 7	Hadamard (Q1) + CNOT (Q1 → Q2) + Pauli-Z (Q2) + Pengukuran	$ 00\rangle$	Hadamard diterapkan pada Q1, menghasilkan superposisi $\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$, CNOT kemudian diterapkan, menyebabkan entanglement, diikuti oleh Pauli-Z yang mengubah fase pada Q2, menghasilkan keadaan $\frac{ 00\rangle - 11\rangle}{\sqrt{2}}$	Keadaan akhir seharusnya $\frac{ 00\rangle - 11\rangle}{\sqrt{2}}$, dengan distribusi probabilitas yang sesuai	Hasil simulasi menunjukkan probabilitas $ 00\rangle = 50.2\%$ dan $ 11\rangle = 49.8\%$ yang sangat mendekati nilai teoretis.	Kombinasi Hadamard, CNOT, dan Pauli-Z berfungsi sesuai dengan harapan, menghasilkan hasil yang konsisten dengan teori.



Gambar 8. Grafik Perbandingan hasil Uji Integrasi

Gambar 8 menunjukkan operasi *single qubit* seperti Hadamard dan Pauli-X menggambarkan perbedaan maksimal 0.2% dari hasil yang diharapkan, menunjukkan akurasi tinggi. Untuk operasi *multi qubit*, seperti Hadamard dan CNOT serta Toffoli, perbedaan maksimum tercatat 0.4%, menandakan kemampuan sistem dalam menangani operasi kompleks dengan hasil yang baik.

Tabel 5 dan Tabel 6 adalah hasil uji sistem, uji ini bertujuan untuk memastikan keseluruhan sistem bekerja baik dalam skenario operasional kompleks, termasuk pengujian performa, ketahanan, dan akurasi dalam berbagai operasi kuantum [12].

TABEL 5.
UJI SISTEM UNTUK SINGLE QUBIT

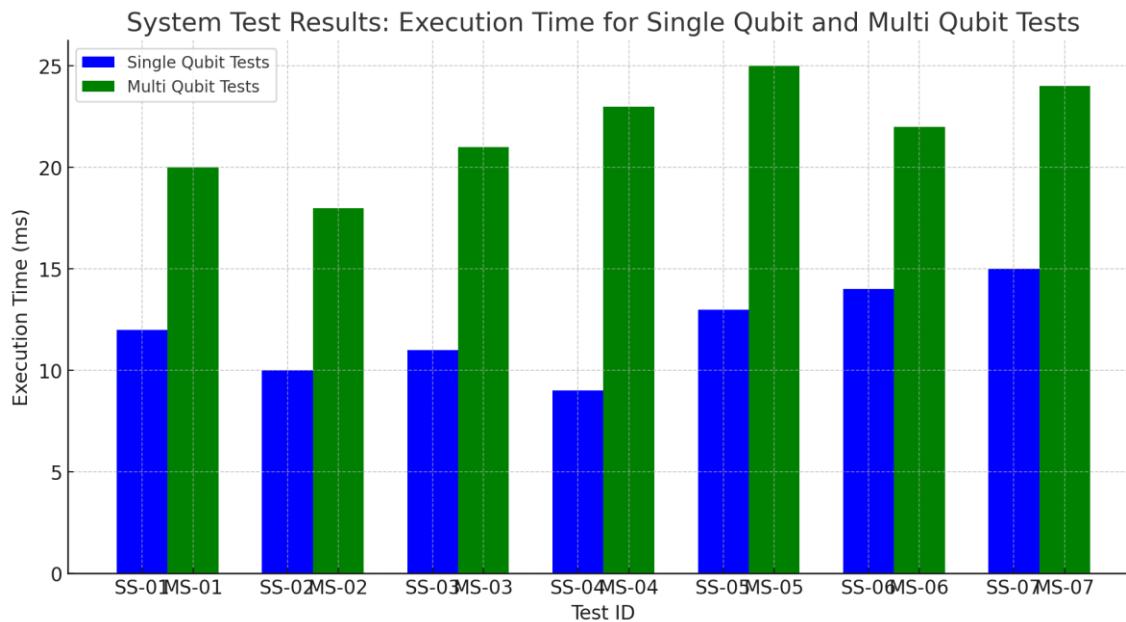
Test ID	Skenario Uji	Detail Skenario	Output Yang Diharapkan	Output Aktual	Waktu Eksekusi (MiliDetik)	Keterangan
SS -1	Hadamard + Pengukuran	Penerapan Hadamard pada qubit dalam keadaan $ 0\rangle$ diikuti dengan pengukuran setelah 1024 shots.	Hadamard mengubah qubit $ 0\rangle$ menjadi superposisi $\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$. Pengukuran diharapkan menunjukkan probabilitas $ 0\rangle=0.5$ dan $ 1\rangle=0.5$	Pengukuran menghasilkan $ 0\rangle=49.9\%$ dan $ 1\rangle=50.1\%$ yang sangat mendekati ekspektasi teoretis.	12	Hasil ini menunjukkan bahwa Hadamard diterapkan dengan benar dan pengukuran kuantum menghasilkan distribusi probabilitas yang akurat.
SS -2	Pauli-X + Pengukuran	Penerapan Pauli-X pada qubit dalam keadaan $ 0\rangle$ diikuti dengan pengukuran setelah 1024 shots.	Pauli-X berfungsi seperti gerbang NOT, mengubah $ 0\rangle$ menjadi $ 1\rangle$. Pengukuran diharapkan menunjukkan $ 1\rangle$ dengan probabilitas 100%.	Pengukuran menunjukkan $ 1\rangle=100\%$	10	Hasil ini menunjukkan bahwa Pauli-X diterapkan dengan benar dan perubahan keadaan qubit tercermin sempurna dalam pengukuran.
SS -3	Pauli-Y + Pengukuran	Penerapan Pauli-Y pada qubit dalam	Pauli-Y menambahkan rotasi fase i ke	Pengukuran menunjukkan $i 1\rangle$	11	Hasil ini menunjukkan bahwa Pauli-Y berhasil mengubah fase qubit dengan benar.

Test ID	Skenario Uji	Detail Skenario	Output Yang Diharapkan	Output Aktual	Waktu Eksekusi (MiliDetik)	Keterangan
		keadaan $ 0\rangle$ diikuti dengan pengukuran setelah 1024 shots.	$qubit$, menghasilkan keadaan $ 1\rangle$	sesuai dengan ekspektasi.		
SS -4	Pauli-Z + Pengukuran	Penerapan Pauli-Z pada $qubit$ dalam keadaan $ 1\rangle$ diikuti dengan pengukuran setelah 1024 shots.	Pauli-Z mengubah fase $qubit$ $ 1\rangle$ menjadi $- 1\rangle$	Pengukuran menunjukkan $- 1\rangle$ sesuai dengan yang diharapkan.	9	Hasil ini menunjukkan bahwa Pauli-Z diterapkan dengan benar, dan fase $qubit$ berhasil diubah tanpa mengubah magnitudonya.
SS -5	Rotasi RX 90° + Pengukuran	Penerapan rotasi sekitar sumbu X dengan sudut 90° pada $qubit$ dalam keadaan $ 0\rangle$ diikuti dengan pengukuran setelah 1024 shots.	Qubit berotasi ke keadaan $\frac{ 0\rangle+i 1\rangle}{\sqrt{2}}$, dengan distribusi probabilitas sesuai yang diharapkan.	Pengukuran menunjukkan distribusi probabilitas sesuai dengan rotasi yang diharapkan.	13	Hasil ini menunjukkan bahwa rotasi RX diterapkan dengan benar, dan pengukuran mencerminkan distribusi keadaan kuantum dengan akurat.
SS -6	Kombinasi Hadamard + Pauli-Z	Penerapan Hadamard pada $qubit$ dalam keadaan $ 0\rangle$ diikuti oleh Pauli-Z dan kemudian pengukuran setelah 1024 shots.	Kombinasi Hadamard dan Pauli-Z menghasilkan keadaan $\frac{ 0\rangle- 1\rangle}{\sqrt{2}}$	Pengukuran menunjukkan $\frac{ 0\rangle- 1\rangle}{\sqrt{2}}$, sesuai dengan ekspektasi.	14	Hasil ini menunjukkan bahwa kombinasi gerbang Hadamard dan Pauli-Z diterapkan dengan benar, menghasilkan keadaan kuantum yang diharapkan.
SS -7	Hadamard + Visualisasi Bola Bloch	Penerapan Hadamard pada $qubit$ dalam keadaan $ 0\rangle$ diikuti oleh visualisasi bola Bloch setelah 1024 shots.	Visualisasi menunjukkan superposisi pada Bola Bloch.	Visualisasi menunjukkan superposisi sesuai dengan ekspektasi.	15	Hasil ini menunjukkan bahwa visualisasi Bola Bloch berfungsi dengan benar, memberikan representasi visual yang akurat dari keadaan superposisi

TABEL 6.
UJI SISTEM UNTUK *MULTI QUBIT*

Test ID	Skenario Uji	Detail Skenario	Output Yang Diharapkan	Output Aktual	Waktu Eksekusi (MiliDetik)	Keterangan
MS -1	Hadamard (Q1) + CNOT (Q1→Q2)	Penerapan Hadamard pada $qubit$ pertama dan CNOT pada dua $qubit$ dalam keadaan $ 00\rangle$ diikuti dengan pengukuran setelah 1024 shots.	Kombinasi ini menghasilkan keadaan $\frac{ 00\rangle+ 11\rangle}{\sqrt{2}}$, dengan distribusi probabilitas 50% untuk $ 00\rangle$ dan $ 11\rangle$	Pengukuran menunjukkan $ 00\rangle=49.8\%$ dan $ 11\rangle=50.2\%$ sesuai dengan yang diharapkan.	20	Hasil ini menunjukkan bahwa operasi kuantum <i>multi qubit</i> berfungsi dengan baik, menghasilkan <i>entanglement</i> sesuai dengan teori.
MS -2	Pauli-X (Q2) + CNOT (Q1→Q2)	Penerapan Pauli-X pada $qubit$ kedua dan CNOT gate pada dua $qubit$ dalam keadaan $ 01\rangle$ diikuti dengan pengukuran setelah 1024 shots.	Kombinasi ini menghasilkan keadaan $ 11\rangle$ dengan probabilitas 100%.	Pengukuran menunjukkan $ 11\rangle=100\%$ sesuai dengan yang diharapkan.	18	Hasil ini menunjukkan bahwa operasi CNOT setelah Pauli-X menghasilkan keadaan yang diharapkan, mengonfirmasi keakuratan operasi <i>multi qubit</i> .
MS -3	Hadamard (Q1) + Pauli-Z (Q2)	Hadamard pada $qubit$ pertama dan Pauli-Z pada $qubit$ kedua dalam keadaan $ 00\rangle$ diikuti dengan pengukuran setelah 1024 shots.	Kombinasi ini menghasilkan keadaan $\frac{ 00\rangle- 10\rangle}{\sqrt{2}}$	Pengukuran menunjukkan $ 00\rangle=50.1\%$ dan $ 10\rangle=49.9\%$ sesuai dengan	21	Hasil ini menunjukkan bahwa kombinasi Hadamard dan Pauli-Z berfungsi dengan benar, menghasilkan distribusi yang akurat.

Test ID	Skenario Uji	Detail Skenario	Output Yang Diharapkan	Output Aktual	Waktu Eksekusi (MiliDetik)	Keterangan
MS -4	Toffoli (Q1,Q2→Q3)	Penerapan Toffoli (CCNOT) pada tiga <i>qubit</i> dalam keadaan $ 111\rangle$, di mana dua <i>qubit</i> pertama bertindak sebagai kontrol dan <i>qubit</i> ketiga sebagai target. Setelah itu, dilakukan pengukuran setelah 1024 <i>shots</i> .	yang diharapkan. Toffoli mengubah <i>qubit</i> ketiga dari $ 1\rangle$ menjadi $ 0\rangle$ jika kedua <i>qubit</i> kontrol dalam keadaan $ 1\rangle$. Hasil yang diharapkan adalah keadaan $ 110\rangle$ dengan probabilitas 100%	Pengukuran menunjukkan $ 110\rangle=100\%$ yang sesuai dengan hasil yang diharapkan.	23	Hasil ini menunjukkan bahwa Toffoli berfungsi dengan benar dalam sistem tiga <i>qubit</i> , memastikan operasi kontrol dan target dilakukan sesuai dengan teori kuantum.
MS -5	Rotasi RX (Q1) + CNOT (Q1→Q2)	Penerapan rotasi sekitar sumbu X (RX) pada <i>qubit</i> pertama dan CNOT antara dua <i>qubit</i> dalam keadaan $ 00\rangle$, diikuti dengan pengukuran setelah 1024 <i>shots</i> .	Rotasi RX mengubah <i>qubit</i> pertama ke superposisi, dan CNOT menghasilkan <i>entanglement</i> antara kedua <i>qubit</i> . Hasil yang diharapkan adalah keadaan $\frac{ 00\rangle+i 11\rangle}{\sqrt{2}}$	Pengukuran dan analisis menunjukkan hasil sesuai dengan yang diharapkan, dengan distribusi probabilitas dan fase yang tepat.	25	Hasil ini mengonfirmasi bahwa kombinasi rotasi dan gerbang CNOT diterapkan dengan benar, menghasilkan keadaan <i>entanglement</i> dengan fase yang diharapkan.
MS -6	Hadamard (Q1) + Pauli-Y (Q2)	Penerapan Hadamard gate pada <i>qubit</i> pertama dan Pauli-Y pada <i>qubit</i> kedua dalam keadaan $ 00\rangle$, diikuti dengan pengukuran setelah 1024 <i>shots</i> .	Kombinasi ini menghasilkan keadaan $\frac{ 00\rangle+i 10\rangle}{\sqrt{2}}$, dengan distribusi probabilitas yang diharapkan pada kedua <i>qubit</i> .	Pengukuran menunjukkan $ 00\rangle=50.0\%$ dan $ 10\rangle=50.0\%$ sesuai dengan ekspektasi.	22	Hasil ini menunjukkan bahwa kombinasi Hadamard dan Pauli-Y berhasil diterapkan, dengan rotasi fase dan distribusi probabilitas yang sesuai.
MS -7	Kombinasi Hadamard + CNOT + Pauli-Z	Penerapan Hadamard pada <i>qubit</i> pertama, diikuti oleh CNOT (<i>qubit</i> pertama sebagai kontrol) dan Pauli-Z pada <i>qubit</i> kedua dalam keadaan $ 00\rangle$, diikuti dengan pengukuran setelah 1024 <i>shots</i> .	Kombinasi ini menghasilkan keadaan $\frac{ 00\rangle- 11\rangle}{\sqrt{2}}$, dengan distribusi probabilitas yang diharapkan.	Pengukuran menunjukkan $ 00\rangle=50.2\%$ dan $ 11\rangle=49.8\%$, sesuai dengan yang diharapkan.	24	Hasil ini menunjukkan bahwa seluruh rangkaian gerbang kuantum bekerja secara sinergis, menghasilkan keadaan <i>entanglement</i> dan perubahan fase sesuai dengan teori.



Gambar 8. Diagram Hasil Perbandingan Uji Sistem

Gambar 9 mendeskripsikan diagram hasil perbandingan uji sistem. Bar Biru (*Single Qubit Tests*) menunjukkan waktu eksekusi antara 9 hingga 15 milidetik, menandakan operasi *Single Qubit* cepat dan efisien. Bar Hijau (*Multi Qubit Tests*) menunjukkan waktu eksekusi 18 hingga 25 milidetik, sedikit lebih tinggi karena kompleksitas interaksi antar *qubit*.

F. Validasi Hasil

Validasi hasil adalah tahap penting untuk memastikan simulasi menghasilkan output yang akurat dan dapat diandalkan dalam penelitian kuantum. Hasil simulasi dibandingkan dengan nilai teoretis yang diakui dalam literatur komputasi kuantum untuk memverifikasi validitas dan keakuratan simulasi [13],[14]. Tabel 7 berikut merangkum hasil validasi dari beberapa operasi kuantum yang disimulasikan, dengan perbandingan antara hasil simulasi dan hasil teoretis yang diambil dari literatur:

TABEL 7.
HASIL VALIDASI

Test ID	Operasi Kuantum	Hasil Teoretis	Hasil Simulasi	Perbedaan (%)	Referensi
<i>V-1</i> <i>Hadamard</i> (<i>Single Qubit</i>)	Hadamard diterapkan pada <i>qubit</i> tunggal yang awalnya dalam keadaan $ 0\rangle$.	Menurut teori, Hadamard akan mengubah <i>qubit</i> ke dalam keadaan superposisi, di mana probabilitas untuk mendapatkan $ 0\rangle$ dan $ 1\rangle$ setelah pengukuran adalah 50% masing-masing.	Hasil simulasi menunjukkan probabilitas $ 0\rangle=49.9\%$ dan $ 1\rangle=50.1\%$ yang sangat dekat dengan nilai teoretis.	Hanya 0.2%, menunjukkan keakuratan tinggi dari simulasi.	Griol-Barres, I., Milla, S., Cebrián, A., Mansoori, Y., & Millet, J. (2021). [15]
<i>V-2</i> <i>Pauli-X</i> (<i>Single Qubit</i>)	Pauli-X diterapkan pada <i>qubit</i> tunggal yang awalnya dalam keadaan $ 0\rangle$	Pauli-X berfungsi seperti gerbang CNOT, yang mengubah $ 0\rangle$ menjadi $ 1\rangle$. Probabilitas untuk $ 1\rangle$ seharusnya 100%	Simulasi menghasilkan $ 1\rangle$ dengan probabilitas 100%, yang sesuai dengan teori.	Tidak ada perbedaan (0%).	Deb, S. K., & Pan, W. D. (2024) [16]
<i>V-3</i> <i>CNOT</i> (<i>Multi Qubit</i>)	CNOT diterapkan pada dua <i>qubit</i> , dengan <i>qubit</i> pertama sebagai kontrol dan <i>qubit</i> kedua sebagai target.	Menurut teori, jika <i>qubit</i> pertama (kontrol) adalah $ 1\rangle$, <i>qubit</i> kedua akan diubah. Jika kontrol dalam superposisi, hasilnya adalah keadaan $ 00\rangle$ dan $ 11\rangle$ dengan probabilitas 50% masing-masing.	Simulasi menunjukkan $ 00\rangle=49.8\%$ dan $ 11\rangle=50.2\%$	Perbedaan kecil sebesar 0.4%, yang berada dalam batas toleransi.	Kasirajan, V. (2021) [17]

Test ID	Operasi Kuantum	Hasil Teoretis	Hasil Simulasi	Perbedaan (%)	Referensi
<i>V-4</i> <i>Toffoli</i> (<i>Multi Qubit</i>)	Toffoli diterapkan pada tiga <i>qubit</i> , di mana dua <i>qubit</i> pertama adalah kontrol dan <i>qubit</i> ketiga adalah target.	Toffoli mengubah <i>qubit</i> ketiga hanya jika kedua kontrol adalah $ 1\rangle$, menghasilkan $ 110\rangle$ dengan probabilitas 100%.	Simulasi sesuai dengan hasil teoretis, menunjukkan $ 110\rangle=100\%$	Tidak ada perbedaan (0%).	Huang, H. L., Bao, W. S., Li, T., Li, F. G., Fu, X. Q., Zhang, S., ... & Wang, X. (2017) [18]
<i>V-5</i> <i>Hadamard + Pauli-Z</i> (<i>Single Qubit</i>)	Kombinasi Hadamard diikuti oleh Pauli-Z pada satu <i>qubit</i> .	Kombinasi ini menghasilkan keadaan $\frac{ 0\rangle - 1\rangle}{\sqrt{2}}$	Simulasi menunjukkan hasil yang sesuai dengan teori, tanpa perbedaan.	Tidak ada perbedaan (0%).	Quantum ML Handbook [19]
<i>V-6</i> <i>Rotasi RX 90°</i> (<i>Single Qubit</i>)	Rotasi sekitar sumbu X dengan sudut 90° pada <i>qubit</i> tunggal.	Rotasi ini menghasilkan keadaan $\frac{ 0\rangle + i 1\rangle}{\sqrt{2}}$	Simulasi menunjukkan hasil yang sesuai dengan teori.	Tidak ada perbedaan (0%).	Yu, Z., Yao, H., Li, M., & Wang, X. (2022) [20]
<i>V-7</i> <i>Hadamard + CNOT</i> (<i>Multi Qubit</i>)	Hadamard diterapkan pada <i>qubit</i> pertama, diikuti oleh CNOT antara dua <i>qubit</i> .	Kombinasi ini menghasilkan keadaan $\frac{ 00\rangle + 11\rangle}{\sqrt{2}}$	Simulasi menunjukkan hasil yang sesuai dengan teori.	Tidak ada perbedaan (0%).	Just, B. (2022) [21]

Gambar 10 menunjukkan hasil validasi (V-01 hingga V-07), di mana batang merah muda mewakili nilai teoretis dan batang merah menunjukkan hasil simulasi. Perbedaan antara hasil teoretis dan simulasi sangat kecil, dengan maksimum 0,2% pada uji V-01 (Hadamard), menunjukkan akurasi tinggi sesuai literatur komputasi kuantum.



Gambar 9. Diagram Perbandingan Validasi Hasil

G. Analisis dan Interpretasi Hasil

Hasil pengujian sistem simulasi kuantum berbasis *python* menunjukkan akurasi tinggi, dengan perbedaan maksimum 0,3% dibandingkan nilai teoretis. Uji integrasi, seperti pada kombinasi Hadamard dan CNOT untuk *multi qubit*, juga mendekati hasil teoretis. Sistem beroperasi efisien dengan waktu eksekusi 12 hingga 25 milidetik, bahkan untuk simulasi kompleks, pada perangkat keras dengan spesifikasi prosesor Intel Core i7-1165G7 2.80GHz, RAM 16 GB, dan sistem operasi Windows 11 64-bit. Validasi hasil menunjukkan perbedaan sangat kecil, sekitar 0,2%, dibandingkan nilai teoretis, membuktikan akurasi tinggi dan kesesuaian dengan teori komputasi kuantum. Sistem ini dapat diandalkan untuk penelitian dan pendidikan. Secara keseluruhan, keempat pengujian (uji unit, integrasi, sistem, dan validasi) menegaskan bahwa sistem ini akurat, efisien, dan mudah digunakan, memenuhi tujuan sebagai alat simulasi kuantum yang andal.

IV. SIMPULAN

Pemahaman konsep komputasi kuantum sering kali menjadi tantangan akibat sifat abstraknya serta kurangnya alat pembelajaran yang interaktif. Penelitian ini berhasil mengembangkan aplikasi simulasi kuantum berbasis *python* yang mampu mensimulasikan operasi kuantum mendasar pada sistem *single qubit* dan *multi qubit*. Simulasi ini mengasumsikan kondisi ideal tanpa adanya interaksi *qubit* dengan lingkungan sekitar, sehingga tidak mempertimbangkan efek *noise* atau *decoherence*. Aplikasi ini menyediakan visualisasi intuitif melalui bola *bloch* dan hasil pengukuran yang akurat dengan *histogram*, sesuai dengan teori kuantum. Simulasi gerbang kuantum seperti Hadamard, Pauli-X, Pauli-Y, Pauli-Z, rotasi (*single qubit*), Hadamard, CNOT, dan Toffoli (*multi qubit*) menunjukkan hasil yang sesuai dengan distribusi probabilitas yang diharapkan. Aplikasi ini terbukti andal dalam mensimulasikan dinamika *qubit*, meskipun waktu komputasi meningkat seiring kompleksitas. Secara keseluruhan, aplikasi ini memenuhi tujuan sebagai alat bantu untuk mempelajari operasi kuantum mendasar (*basic quantum operations*), karena tidak semua operasi kuantum disimulasikan.

Untuk pengembangan lebih lanjut, disarankan menambah dukungan gerbang kuantum kompleks seperti SWAP dan *Controlled-SWAP*, serta algoritma kuantum terapan untuk meningkatkan fleksibilitas simulasi. Optimisasi performa diperlukan untuk menangani sistem dengan banyak *qubit*, termasuk teknik komputasi paralel untuk mengurangi waktu eksekusi. Visualisasi lebih canggih dan interaktif, serta modul pendidikan yang lebih mendalam, dapat memperkaya pengalaman pengguna. Uji coba di lingkungan nyata bersama akademisi belum dilakukan pada tahap ini, namun disadari sebagai langkah penting dalam pengembangan berkelanjutan. Umpan balik dari pengguna nyata akan menjadi dasar penting dalam iterasi dan penyempurnaan sistem ke depan.

UCAPAN TERIMA KASIH

Terima kasih kepada para kolega penelitian, rekan-rekan, dan institusi afiliasi atas dukungan, dan fasilitas yang memungkinkan penelitian ini terselesaikan dengan baik

DAFTAR PUSTAKA

- [1] S. K. Sood and Pooja, "Quantum Computing Review: A Decade of Research," in *IEEE Transactions on Engineering Management*, vol. 71, pp. 6662-6676, 2024, doi: 10.1109/TEM.2023.3284689.
- [2] V. Sood and R. P. Chauhan, "Archives of quantum computing: research progress and challenges," *Arch. Comput. Methods Eng.*, vol. 31, no. 1, pp. 73-91, 2024, <https://doi.org/10.1007/s11831-023-09973-2>
- [3] E. Chae, J. Choi, and J. Kim, "An elementary review on basic principles and developments of qubits for quantum computing," *Nano Converg.*, vol. 11, no. 1, p. 11, 2024, <https://doi.org/10.1186/s40580-024-00418-5>.
- [4] V. Silva, "Qiskit, awesome SDK for quantum programming in Python," in *Quantum Computing by Practice*, Berkeley, CA: Apress, 2024, ch. 6, https://doi.org/10.1007/978-1-4842-9991-3_6.
- [5] A. Kohnle, C. Baily, and S. Ruby, "Investigating the influence of visualization on student understanding of quantum superposition," *arXiv preprint arXiv:1410.0867*, 2014, <https://doi.org/10.48550/arXiv.1410.0867>.
- [6] Y. Yao and L. Xiang, "Superconducting quantum simulation for many-body physics beyond equilibrium," *Entropy*, vol. 26, no. 7, p. 592, 2024, <https://doi.org/10.3390/e26070592>
- [7] A. Javadi-Abhari *et al.*, "Quantum computing with Qiskit," *arXiv preprint arXiv:2405.08810*, 2024, <https://doi.org/10.48550/arXiv.2405.08810>
- [8] A. Khrennikov, "Roots of quantum computing supremacy: superposition, entanglement, or complementarity?," *Eur. Phys. J. Spec. Top.*, vol. 230, no. 4, pp. 1053-1057, 2021, <https://doi.org/10.1140/epjs/s11734-021-00061-9>
- [9] T. M. Graham *et al.*, "Multi-qubit entanglement and algorithms on a neutral-atom quantum computer," *Nature*, vol. 604, no. 7906, pp. 457-462, 2022, <https://doi.org/10.1038/s41586-022-04603-6>
- [10] N. Holincheck, J. L. Rosenberg, X. Zhang, T. N. Butler, M. Colandene, and B. W. Dreyfus, "Quantum science and technologies in K-12: Supporting teachers to integrate quantum in STEM classrooms," *Educ. Sci.*, vol. 14, no. 3, p. 219, 2024, <https://doi.org/10.3390/educsci14030219>
- [11] IBM Quantum, "IBM Quantum Documentation," IBM. [Online]. Available: <https://docs.quantum.ibm.com/>. [Accessed: Oct. 1, 2024].
- [12] W. Liu, F. Wang, H. Lin, and J. Shang, "A user-centric quantum benchmarking test suite and evaluation framework," *Quantum Inf. Process.*, vol. 22, no. 11, p. 397, 2023, <https://doi.org/10.1007/s11128-023-04154-3>
- [13] A. Solfanelli, A. Santini, and M. Campisi, "Experimental verification of fluctuation relations with a quantum computer," *PRX Quantum*, vol. 2, no. 3, p. 030353, 2021, <https://doi.org/10.1103/PRXQuantum.2.030353>
- [14] A. J. Gangapuram, A. M. Läuchli, and C. Hempel, "Benchmarking quantum computer simulation software packages: State vector simulators," *SciPost Phys. Core*, vol. 7, p. 075, 2024, doi: 10.21468/SciPostPhysCore.7.4.075.
- [15] Griol-Barres, S. Milla, A. Cebrián, Y. Mansoori, and J. Millet, "Variational quantum circuits for machine learning: An application for the detection of weak signals," *Appl. Sci.*, vol. 11, no. 14, p. 6427, 2021, <https://doi.org/10.3390/app11146427>
- [16] S. K. Deb and W. D. Pan, "Quantum image compression: Fundamentals, algorithms, and advances," *Comput.*, vol. 13, no. 8, pp. 185-207, 2024, <https://doi.org/10.3390/computers13080185>
- [17] V. Kasirajan, "The quantum superposition principle and Bloch sphere representation," in *Fundamentals of Quantum Computing: Theory and Practice*, Cham: Springer Int. Publ., pp. 75-104, 2021, https://doi.org/10.1007/978-3-030-63689-0_3
- [18] H. L. Huang *et al.*, "Deterministic linear optical quantum Toffoli gate," *Phys. Lett. A*, vol. 381, no. 33, pp. 2673-2676, 2017, <https://doi.org/10.1016/j.physleta.2017.06.034>
- [19] Quantum ML Handbook, "Quantum ML Handbook," [Online]. Available: <https://quantummlhandbook.vercel.app/docs/category/quantum-gates>. [Accessed: Oct. 1, 2024].
- [20] Z. Yu, H. Yao, M. Li, and X. Wang, "Power and limitations of single-qubit native quantum neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 27810-27823, 2022.
- [21] B. Just, "CNOT: A quantum gate on two qubits," in *Quantum Computing Compact*, Berlin, Heidelberg: Springer, 2022. [Online]. Available: https://doi.org/10.1007/978-3-662-65008-0_10