Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan penelitian yang telah dilaksanakan, maka dapat disimpulkan bahwa telah berhasil proses implementasi algoritma kuantum Shor menggunakan *library* Python ProjectQ sebagai simulator komputer kuantum. Terdapat beberapa simpulan yang dapat diambil dari penelitian ini, sebagai berikut.

1. *Clock-speed* dan jumlah *core* yang digunakan berpengaruh terhadap waktu yang dibutuhkan untuk menyimulasikan algoritma kuantum Shor. Program yang dirancang pada penelitian ini mendapatkan peningkatan performa sebesar 33% dari perangkat 2 pada saat dijalankan dengan perangkat 1.
2. Implementasi algoritma Shor pada penelitian ini memiliki waktu eksekusi terlama dibandingkan dengan implementasi ProjectQ dan Quantum Computing Playground.
3. Implementasi algoritma kuantum Shor ProjectQ memiliki waktu eksekusi yang paling cepat dan memiliki penggunaan jumlah qubit paling rendah dibandingkan implementasi algoritma kuantum Shor pada penelitian ini dan Quantum Computing Playground.
4. Tingkat akurasi pemfaktoran perkalian bilangan pada implementasi algoritma kuantum Shor pada penelitian ini berbanding terbalik dengan jumlah qubit yang digunakan.

UNIVERSITAS MULTIMEDIA NUSANTARA

5.2 Saran

Berdasarkan implementasi algoritma kuantum Shor yang telah dirancang pada penelitian ini, maka berikut ini adalah saran untuk pengembangan program ini ke masa yang akan datang.

1. Jumlah qubit yang digunakan dapat dikurangi untuk mengurangi waktu eksekusi pencarian faktor dari bilangan yang dimasukkan. Ini dapat dilakukan dengan merancang proses *modular exponential* yang membutuhkan jumlah qubit yang lebih sedikit dari yang dijelaskan pada Bab 2.4.
2. Program pemfaktoran yang telah dirancang dapat dikembangkan dari segi akurasi pencarian faktor dari bilangan yang dimasukkan. Ini dapat dilakukan dengan mengurangi jumlah qubit yang digunakan. Karena ini, dengan menerapkan poin pertama, peningkatan pada poin ini juga dapat dihasilkan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Implementasi algoritma kuantum..., Adjie Wahyu Wicaksono, FTI UMN, 2019