

IBM-Q: PENGKOMPUTERAN QUANTUM BERASASKAN AWAN

Tan Ying Fen

Dr. Bahari Idrus

*Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia***ABSTRAK**

Komputer quantum dapat memproses jauh lebih cepat berbanding dengan komputer sedia ada. Namun begitu, komputer quantum masih belum ada dalam bentuk komersial. Pada masa ini terdapat beberapa agensi utama dunia seperti IBM, Google, Microsoft, Intel, Rigetti dan beberapa agensi yang lain telah membenarkan masyarakat untuk mencapai komputer quantum mereka berasaskan awan (cloud-based). Kajian ini meneroka IBM Quantum Experience (IBM-Q) yang dihasilkan oleh IBM yang menyediakan capaian ke komputer quantum yang sebenar. Dengan kemudahan capaian ini, penerokaan dilakukan terhadap IBM-Q untuk menyediakan manual pemasangan (installation) Qiskit iaitu bahasa pengaturcaraan yang digunakan oleh IBM-Q, manual operasi asas pengkomputeran quantum seperti get quantum dan litar quantum. Akhir sekali, pengujian terhadap IBM-Q dengan menggunakan Qiskit untuk beberapa perkakasan komputer quantum seperti 5 qubit dan 16 qubit diuji dengan mengimplimentasikan algoritma Grover. Dengan penerokaan IBM-Q ini dapat memahami bidang komputer quantum dengan lebih baik agar beberapa masalah sebenar seperti pengoptimuman, pembelajaran mesin (machine learning), kriptografi dan lain-lain dapat diselesaikan dengan menggunakan komputer quantum.

1 PENGENALAN

Komputer quantum merupakan suatu yang realiti pada masa kini. Terdapat beberapa agensi utama dunia seperti IBM, Rigetti, Google, Microsoft, Intel dan beberapa lagi yang sedang cuba merealisasikan komputer quantum (LaRose, R. 2019). Komputer quantum

dapat memproses jauh lebih pantas berbanding dengan komputer yang sedia ada walaupun dengan superkomputer yang paling pantas buat masa ini. Ini telah dibukti oleh Shor (1994) dengan menyelesaikan masalah pemfaktoran menggunakan algoritma Shor yang dapat diselesaikan sepantas $O((\log N)^3)$. Mana kala Grover (1996) dengan menggunakan algoritma Grover dapat menyelesaikan pencarian tak berstruktur dengan kepantasan $O(\sqrt{N})$. Terkini, komputer quantum Google telah menyelesaikan masalah yang kompleks iaitu membuktikan kerawakan nombor yang dihasilkan oleh penjana nombor rawak dalam masa 3 minit 20 saat di mana masalah yang sama jika dilakukan pada superkomputer yang terpanjang di dunia iaitu IBM summit memerlukan 10,000 tahun (Oon Yeoh 2019).

Walau bagaimanapun, cabaran utama untuk menghasilkan komputer quantum pada masa kini amat tinggi terutama dari segi perkakasan yang memerlukan penyimpanan di bawah paras suhu beku. Pada masa kini terdapat beberapa agensi utama dunia berlumba-lumba untuk menghasilkan komputer quantum yang dapat dikomersialkan. Pada September, 2019, IBM telah mengistiharkan telah menghasilkan komputer quantum komersial yang boleh dipasarkan untuk 53 qubit yang boleh diguna oleh masyarakat (Hruska, J. 2019). Begitu juga dengan Microsoft, telah mengistiharkan penggunaan komputer quantum awan mereka yang dikenali sebagai Azure (Simonite T. 2019). Manakala Google juga tidak ketinggalan telah melancarkan komputer quantum supremacy mereka pada Oktober 2019 (Gibney E. 2019).

Sekarang ini, dengan teknologi berasaskan awan, masyarakat dibenarkan menggunakan komputer quantum yang sebenar sama ada secara percuma atau berbayar. Beberapa agensi yang menawarkan perkhidmatan ini adalah: Forest oleh Rigetti Computing, LIQUi|> oleh Microsoft, IBM-Q Experience menggunakan Qiskit oleh IBM, ProjectQ oleh ETH Zurich dan juga Google. Namun terdapat 3 platform yang dikatakan mempunyai capaian dengan komputer quantum yang sebenar iaitu, Forest, Qiskit dari IBM dan ProjectQ (LaRose 2019). Komputeran quantum berasaskan awan, membenarkan komuniti menghantar aturcara quantum mereka ke komputer pelayan

quantum secara komputeran awan, hasilnya akan dimaklumkan kepada komuniti atau pengguna.

Projek ini akan meneroka IBM-Q yang menggunakan Qiskit berasaskan bahasa pengaturcaraan Python. IBM-Q Experience menawarkan sepuluh komputer quantum yang terdiri dari lima 20 qubit, satu 14 qubit dan empat 5 qubit (Hruska, J. 2019). Fokus projek ini adalah penerokaan IBM-Q dari segi pemasangan, penggunaan bahasa pengaturcaraan terutama untuk operasi-operasi mudah dalam quantum dan juga beberapa aplikasi mudah yang piawai digunakan dalam masalah quantum. Projek ini juga melihat sokongan yang diberi seperti tutorial, dokumentasi dan bantuan. Akhirnya, untuk menguji platform IBM-Q experience ini dalam menyelesaikan masalah quantum dalam bentuk pengkomputeran awan, algoritma Grover dicuba sebagai sampel. Algoritma Grover adalah algoritma carian data dalam pengkomputeran quantum yang dicipta oleh Lov Grover pada tahun 1996. Algoritma Grover adalah mencari pangkalan data yang tidak berstruktur dengan menggunakan entri N dalam masa $O(\sqrt{N})$ dan menggunakan ruang storan informasi $O(\log N)$.

2 PENYATA MASLAH

Lumrahnya, komputeran yang ingin dihasilkan mestilah dapat memproses sesuatu masalah atau data dengan cepat. Kepantasan pemprosesan komputer merupakan elemen penting dalam sesuatu komputer. Oleh itu, setiap tahun, komputer yang akan dihasilkan dipertingkatkan dari segi pemprosesan untuk menghasilkan pemprosesan yang makin cepat. Menurut Hukum Moore, setiap 2 tahun, bilangan transistor dalam komputer akan bertambah dua kali ganda. Dengan ini, bilangan transistor akan sentiasa ditambah di dalam mikrochip yang menyebabkan saiz transistor semakin kecil setiap tahun. Ini menyebabkan sampai satu tahap, saiz transistor mencapai satu tahap yang sangat kecil atau dipanggil saiz atom. Ini menyebabkan hukum fizik moden tidak boleh diguna pakai lagi menyebabkan hukum quantum diperkenalkan. Komputer quantum menggunakan fenomena mekanik quantum dalam operasinya dan dipercayai dapat memproses jauh lebih pantas dari komputer yang ada sekarang ini. Elemen utama komputer quantum yang

menyebabkan berlaku kepantasan yang luar biasa ialah sifat superposisi dan keterbelitan (entanglement) yang merupakan ciri dalam mekanik quantum.

Pada masa ini terdapat banyak komputer quantum simulator yang dibangunkan seperti yang disenaraikan dalam <https://www.quantiki.org/wiki/list-qc-simulators>, namun, kebanyakannya dilaksanakan menggunakan teknologi komputer yang ada sekarang, bukan dengan komputer quantum. Ini boleh menghasilkan keputusan yang kurang tepat disebabkan tidak menggunakan komputer quantum yang sebenar. IBM-Q, salah satu platform komputeran awan quantum yang menggunakan komputer quantum yang sebenar. Jadi, IBM-Q digunakan dalam projek ini untuk diterokai untuk memahami operasi-operasi dalam komputer quantum.

3 OBJEKTIF KAJIAN

Objektif utama kajian ini adalah untuk meneroka IBM-Q yang merupakan komputer quantum berasaskan awan yang menggunakan teknologi komputer quantum yang sebenar. Fokus utama kajian ini adalah dari segi pengimplimentasi bahasa pengaturcaraan yang di gunakan oleh IBM-Q untuk melaksanakan algoritma quantum yang sedia ada seperti algoritma Grover pada beberapa perkakasan komputer quantum yang disediakan oleh IBM-Q seperti sistem 5 qubit dan sistem 16 qubit. Hasil kajian ini dapat:

1. Menerbitkan beberapa manual untuk IBM-Q seperti manual pemasangan (install) IBM-Q, manual pengimplimentasian beberapa operasi piawai dalam komputeran quantum seperti operasi get (gate) dan litar (circuits) untuk algoritma-algoritma quantum yang mudah.
2. Menguji dan menganalisis algoritma Grover menggunakan perkakasan quantum IBM-Q yang berbeza seperti `ibmq_qasm_simulator` dan `ibmq` Burlington.

4 METOD KAJIAN

Error! Bookmark not defined.

4.1 Error! Bookmark not defined.

Fasa pertama kajian ini adalah membaca artikel mengenai komputer quantum, komputer quantum berasaskan awan, IBM-Q dan algoritma Grover. Tujuan utama kajian ini adalah untuk meneroka IBM-Q, salah satu komputer quantum berasaskan awan yang menggunakan komputer quantum sebenar. Algoritma Grover dijadikan sampel untuk pengujian IBM-Q. Pemahaman teori asas komputeran quantum diperlukan terutamanya bit quantum atau qubit (quantum bit), get logik dan operasinya serta litar quantum. Pemahaman ini diperlukan untuk memahami algoritma quantum. Akhirnya, perlu memahami algoritma Grover iaitu algoritma yang akan dijadikan sampel ujian dalam kajian ini.

Error! Bookmark not defined.

4.2 Baca dan memahami manual IBM-Q

Fasa kedua kajian ini adalah membaca manual dan tutorial mengenai IBM-Q. Pemahaman tentang keperluan IBM-Q dan cara pemasangan IBM-Q amat diperlukan dalam kajian ini. Selain itu, tutorial IBM-Q akan mengajar pengetahuan asas mengenai komputer quantum yang akan meningkatkan pemahaman mengenai komputer quantum.

4.3 Error! Bookmark not defined.

Fasa ketiga kajian ini adalah untuk menulis atur cara dalam IBM-Q yang menggunakan Qiskit sebagai platform aturcara. Pengimplimentasian aturcara dalam Qiskit terbahagi kepada dua iaitu, aturcara di larikan menggunakan simulator yang disediakan secara sendiri (standalone). Apabila di dapati tiada kesilapan aturcara maka aturcara akan dilarikan menggunakan komputer quantum sebenar yang disediakan oleh IBM secara awan.

Error! Bookmark not defined.

4.4 Error! Bookmark not defined.

Fasa keempat kajian ini adalah mengkaji algoritma Grover menggunakan IBM-Q. Dalam fasa ini, pengujian dilakukan dengan menggunakan komputer quantum sebenar IBM-Q berasaskan pengkomputeran awan. Tujuan ujian ini adalah untuk membandingkan kelajuan carian terhadap pangkalan data pencarian antara komputer quantum dan komputer klasik. Program ini akan dihantar ke IBM-Q melalui awan dan dijalankan oleh penyelidik IBM-Q menggunakan komputer quantum sebenar.

4.5 Error! Bookmark not defined.

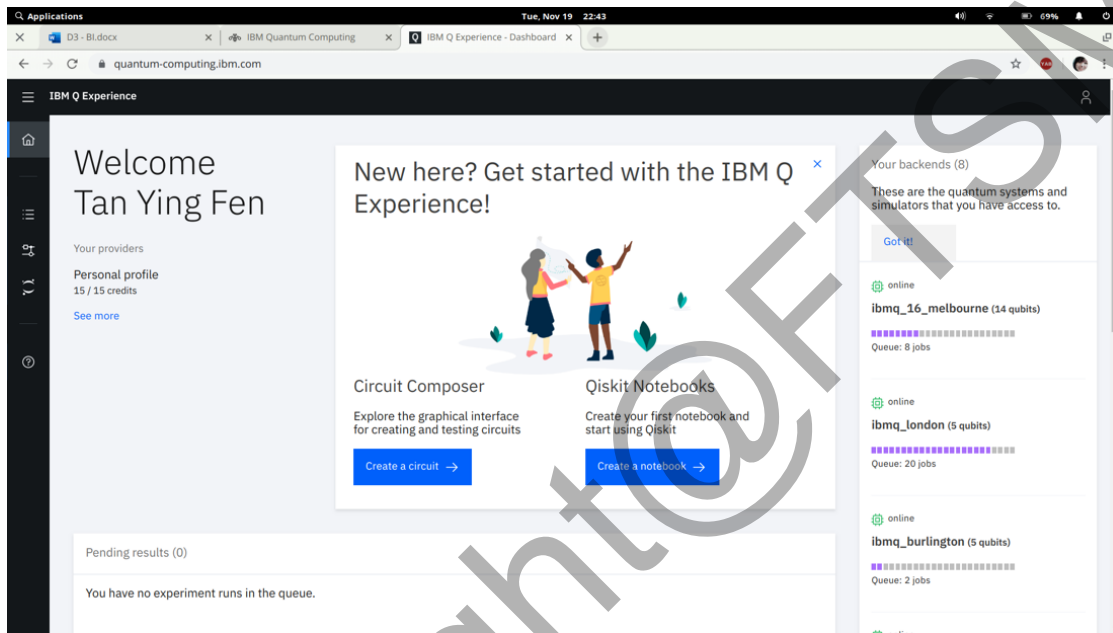
Fasa terakhir kajian ini adalah menganalisis data. Selepas algoritma dilaksanakan oleh IBM-Q, data perhitungan akan dikembalikan. Analisis data akan dijalankan dan hasilnya akan ditulis di dalam laporan.

5 HASIL KAJIAN

Komputer quantum merupakan satu mesin yang menggunakan hukum mekanik quantum yang mana berbeza dengan komputer masa kini yang menggunakan hukum fizik moden. Terdapat dua ciri utama dalam komputer quantum yang menyebabkan ia dapat memproses jauh lebih pantas dari komputer sedia ada iaitu superposisi dan keterbelitan (entanglement). Superposisi bermakna kita boleh wakikan data iaitu bit quantum atau qubit dalam bentuk $|0\rangle$ dan $|1\rangle$ secara serentak. Ini berbeza dengan komputer sedia ada di mana kita boleh wakikan data sama ada dalam bit 0 atau 1. Ciri keterbelitan akan mewujudkan satu korelasi antara dua zarah tanpa mengambil kira jaraknya.

Terdapat banyak syarikat yang mempunyai peranti quantum yang sebenar telah menyediakan pengkomputeran quantum berasaskan awan kepada orang ramai. IBM-Q Experience merupakan salah satu platform atas talian yang memberi pengguna untuk mengakses ke set pemproses quantum IBM berasaskan awan. Ia mempunyai forum untuk membincangkan topik yang berkaitan dengan pengkomputeran quantum dan tutorial mengenai cara memprogramkan peranti IBM-Q. Di laman web IBM-Q Experience, beberapa pengetahuan asas mengenai pengkomputeran quantum boleh dipelajari. Selepas

log masuk ke akaun IBM-Q, pengguna boleh membina litar kompleks dengan get logik. IBM-Q juga boleh membangunkan perisian quantum dengan kod mesin OpenQASM. Terdapat alat visualisasi memberi cara yang berbeza untuk memperhatikan, melarikan dan memeriksa ralat litar. Pengguna boleh mengedit kod dan melihat perubahan litar secara grafik pada masa ke masa.



Rajah 1 Dashboard IBM-Q Experience

Qiskit adalah kerangka untuk pembangunan perisian pengkomputeran quantum. Ia diasaskan oleh IBM Research untuk membolehkan pembangunan perisian menggunakan pengkomputeran quantum awan IBM-Q Experience. Ia menyediakan alat untuk menghasilkan dan memanipulasi program quantum dan menjalankannya menggunakan simulator atau komputer quantum sebenar. Qiskit tersedia untuk Windows, macOS dan Linux. Untuk memasang Qiskit, Python 3.5 atau versi yang lebih tinggi diperlukan. Untuk memeriksa versi Python yang terdapat di komputer,

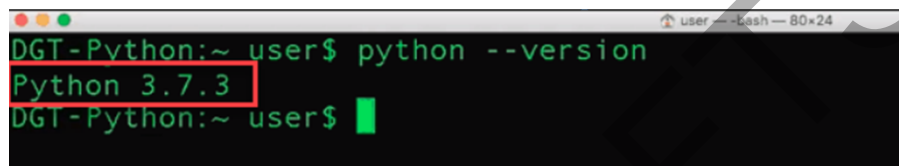
Untuk pengguna Windows : buka “windows search” dan menaip “python”. Menekan “python command line” untuk membuka terminal Python. Nombor versi python akan dipaparkan pada barisan pertama.



```
Python 3.6 (32-bit)
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018)
Type "help", "copyright", "credits" or "license()" for more
>>> _
```

Rajah 2 Versi Python untuk Windows

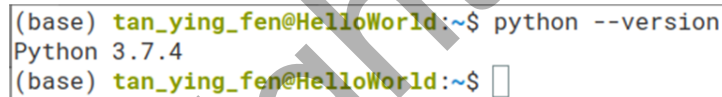
Untuk pengguna macOS : buka terminal dan menaip “ python --version”. Nombor versi python akan dipaparkan selepas kata “Python”.



```
DGT-Python:~ user$ python --version
Python 3.7.3
DGT-Python:~ user$ █
```

Rajah 3 Versi Python untuk MacOS

Untuk pengguna linux : buka terminal dan menaip “ python --version”. Nombor versi python akan dipaparkan selepas kata “Python”.



```
(base) tan_ying_fen@HelloWorld:~$ python --version
Python 3.7.4
(base) tan_ying_fen@HelloWorld:~$ █
```

Rajah 4 Versi Python untuk Linux

Untuk memasang QISKit, platform yang lebih dicadangkan ialah menggunakan aplikasi Anaconda, satu platform silang pengedaran Python untuk pengkomputeran saintifik. Anaconda boleh dimuat turun di laman web rasmi atau melalui pautan berikut <https://www.anaconda.com/distribution/#download-section>.

Untuk pengguna MacOS dan Linux, selepas memuat turun Anaconda, pembangkitkan persekitaran conda perlu dilakukan dengan menaip perintah berikut di terminal. Untuk pengguna Windows, selepas pemasangan anaconda, pengguna hanya perlu mencari Anaconda Prompt dan melaksanakan perintah berikut di Anaconda Prompt.

```
conda create -n name_of_my_env python=3
```



```
source activate name_of_my_env python
```

Selepas itu, untuk pemasangan Qiskit package yang mengandungi Terra, Aer, Ignis, dan Aqua, command di bawah perlu dilaksanakan. “pip” adalah manager Python package.

```
pip install qiskit
```

Dokumentasi Qiskit boleh dicari di internet atau melalui pautan berikut <https://qiskit.org/documentation/install.html>. Ia mengandungi instruksi pemasangan Qiskit, tutorial, contoh program, rumusan Qiskit dan dokumentasi.

Get quantum merupakan elemen yang amat penting dalam pengkomputeran quantum. Ia adalah litar quantum asas yang beroperasi pada sebilangan kecil qubit. Ia adalah analog untuk komputer quantum atas get logik klasik untuk komputer klasik. Get quantum boleh diterbalikkan(reversible) tidak seperti dengan get logik klasik. Beberapa get quantum akan diperkenalkan.

1. Get identiti adalah operasi satu qubit yang tidak mengubah status $|0\rangle$ dan $|1\rangle$. Ia adalah untuk mengenal pasti tidak ada operasi dilaksanakan ke atas qubit tersebut. Matrix get identiti ialah :

2. Get Hadamard memutar status $|0\rangle$ and $|1\rangle$ to $|+\rangle$ and $|-\rangle$ masing-masing. Ia sentiasa digunakan untuk superposisi. Berikut adalah superposisi selepas melaksanakan get Hadamard to $|0\rangle$ and $|1\rangle$. Matrix get Hadamard ialah :

3. Get Pauli X menukar amplitud $|0\rangle$ and $|1\rangle$. Ia memutar vektor di sepanjang paksi-x. Ia sering digambarkan sebagai klasikal get NOT kerana ia membalikkan $|0\rangle$ to $|1\rangle$ (atau sebaliknya). Matrix get Pauli-X ialah :

4. Get Pauli Y menukar amplitud $|0\rangle$ and $|1\rangle$ dan mendarabkannya dengan nombor khayalan (fasa). Dari perspektif kebarangkalian, get ini mempunyai keputusan yang sama seperti get X. Walau bagaimanapun, get ini sangat berguna untuk mewujudkan gangguan konstruktif dan dekonstruktif. Matrix get Pauli-Y ialah :

5. Get Pauli-Z meninggalkan amplitud $|0\rangle$ qubit tidak berubah, manakala mendarab amplitud $|1\rangle$ qubit dengan -1. get ini dari fakta bahawa ia hanya memberi kesan kepada komponen $|1\rangle$, yang akan sering digunakan untuk membuat pemilihan dan tidak mengubah yang lain. Matrix get Pauli-Z ialah :

6. Get anjakan fasa seiras dengan get Pauli-Z. Ia meninggalkan amplitud $|0\rangle$ qubit tidak berubah, manakala mendarabkan amplitud $|1\rangle$ qubit dengan fasa $e^{i\theta}$. Kegunaan get ini kebanyakan sama seperti get Z, memilih status tertentu sementara tidak mengubah yang lain. Walau bagaimanapun, pendaraban fasa $e^{i\theta}$ menjadikan get ini amat berguna untuk mewujudkan kesan gangguan tertentu. Matrix get anjakan fasa ialah :

7. Get fasa-S ialah pra-takrif get untuk R_ϕ , $\phi = \pi/2$. Ia tidak mengubah amplitud $|0\rangle$ qubit sementara mendarabkan amplitud $|1\rangle$ qubit dengan i (fasa). Matrix get phase-S ialah :

8. Get fasa-T ialah pra-takrif get untuk R_ϕ , $\phi = \pi/2$. Ia tidak mengubah amplitud $|0\rangle$ qubit sementara mendarabkan amplitud $|1\rangle$ qubit dengan $e^{i\pi/4}$ (fasa). Matrix get fasa-T ialah :

9. Get putaran-X adalah putaran satu qubit putaran dengan sudut θ (radian) di paksi-x. Sudut putaran mesti ditulis dalam radian dan boleh dalam status positif atau negatif. Matrix get putaran-X ialah :

10. Get putaran-Y adalah putaran satu qubit putaran dengan sudut θ (radian) di paksi-Y. Sudut putaran mesti ditulis dalam radian dan boleh dalam status positif atau negatif. Matrix get putaran-Y ialah :

11. Get putaran-Z adalah putaran satu qubit rotation dengan sudut θ (radian) di paksi-z. Sudut putaran mesti ditulis dalam radian dan boleh dalam status positif atau negatif. Matrix get putaran-Z ialah :

12. . Dalam Get Not Terkawal (CNOT), qubit pertama adalah kawalan dan qubit kedua adalah sasaran. Sekiranya qubit kawalan adalah 0, tidak ada apa-apa untuk sasaran qubit. Jika qubit kawalan adalah 1, sasaran qubit akan berubah. get CNOT adalah salah satu alat penting dalam pengkomputeran quantum. Matrix CNOT ialah :















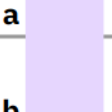

13. Get terkawal-Z get hanya memberi kesan ke status $|11\rangle$ dan status lain yang mempunyai 0 tidak berubah. Matrix get Terkawal-Z ialah :

14. Dalam get tukaran ini, qubit pertama adalah kawalan dan qubit kedua dan ketiga adalah sasaran. Get Tukaran adalah sasaran qubit yang mempunyai status 0 dan 1 akan bertukar tempat. Jadinya, ia tidak memberi kesan kepada status $|00\rangle$ and $|11\rangle$. Matrix get Tukaran ialah :

15. Get Tukaran terkawal juga dikenali sebagai get Fredkin. Ia menggunakan qubit kawalan (qubit pertama) untuk menentukan untuk mengaplikasikan get Tukaran untuk sasaran dua qubit. Sekiranya qubit kawalan berada dalam status $|1\rangle$, maka get Tukaran dapat dijalankan. Matrix CSWAP ialah :

16. Get CCNOT juga dikenali get Toffoli. Ia menggunakan dua qubits kawalan untuk memastikan adakah X-get diaplikasikan ke satu qubit sasaran. Matrix CCNOT ialah :

Litar quantum adalah rutin komputasi yang terdiri daripada data quantum yang menjalankan operasi quantum koheren, seperti qubit, dan komputasi yang serentak. Ia adalah susunan get quantum , pengukuran dan reset. Program quantum boleh ditunjukkan dengan susunan litar quantum dan komputasi klasik yang tidak serentak. Jadual berikut ialah contoh komponen litar.

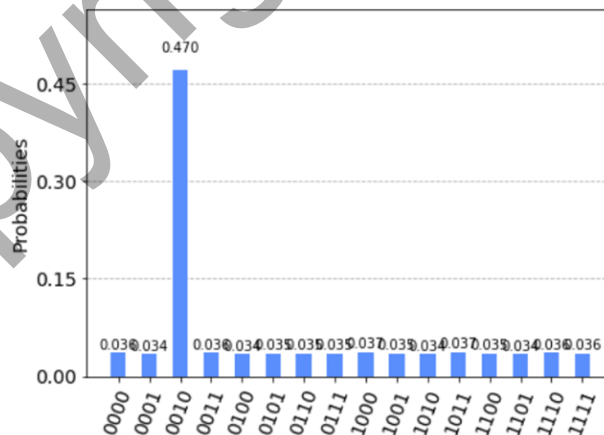
 Identiti	 Hadamard	 Pauli-X	 Pauli-Y
 Pauli-Z	 Anjakan Fasa	 Fasa-S	 Fasa-T
 Putaran-X	 Putaran-Y	 Putaran-Z	 CNOT
 Terkawal-Z	 Tukaran	 CSWAP	 CCNOT

Jadual 1 Jadual contoh komponen litar quantum

Algoritma Grover merupakan contoh yang diguna untuk membuat kajian terhadap perkakasan IBM-Q. Algoritma Grover adalah algoritma carian quantum yang dicipta oleh Lov Grover pada tahun 1996. Algoritma Grover adalah mencari pangkalan data yang tidak diselaraskan dengan entri N dengan $O(N^{1/2})$ kali dan menggunakan ruang penyimpanan $O(\log N)$. Dalam pembinaan litar quantum algoritma grover, terdapat empat langkah yang penting iaitu inisialisasi, oracle, amplifikasi dan pengukuran. Peringkat inisialisasi adalah untuk mendaftarkan quantum dan klasik register. Semua qubit yang didaftar perlu ditetapkan sebagai superposisi supaya semua status mempunyai amplitud . Selepas itu, Satu oracle ditambah ke litar. Oracle ini berfungsi dengan menjalankan fasa pemutaran untuk mensongsangkan amplitud sasaran. Kemudian itu, amplifikasi dijalankan. Ini adalah untuk meningkatkan amplitud sasaran. Akhir sekali, pengukuran dijalankan.

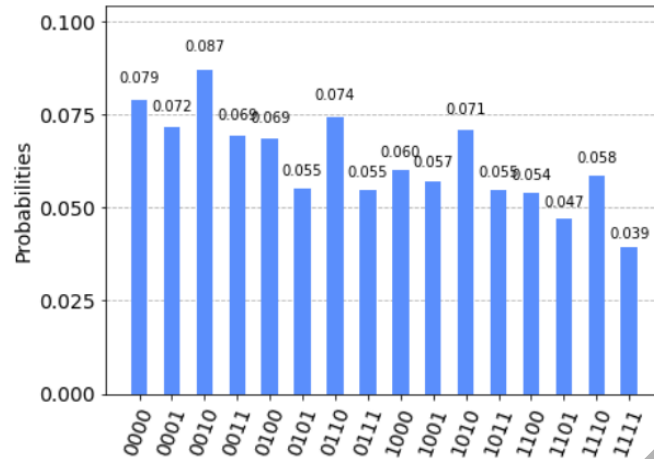
Litar algoritma grover telah dikaji dengan dua backend iaitu `ibmq_qasm_simulator` dan `ibmq_burlington`. Kedua-duanya dikaji dengan 8192 shot.

```
RESULT: {'1100': 290, '1110': 297, '0000': 296, '1010': 277, '1000': 304, '0010': 3853, '0101': 284, '0011': 297, '1101': 280, '0100': 281, '0001': 280, '1001': 284, '1011': 301, '0110': 290, '1111': 291, '0111': 287}
```



Rajah 5 Keputusan dari `ibmq_qasm_simulator`

```
RESULT: {'0101': 451, '0000': 646, '0111': 447, '1011': 448, '1000': 491, '0110': 609, '1101': 385, '1010': 579, '1111': 321, '0001': 586, '1001': 468, '1100': 441, '0011': 568, '0100': 562, '1110': 479, '0010': 711}
```



Rajah 6 Keputusan dari ibmq_burlington

Keadaan yang ditetapkan ialah $|0010\rangle$. Kedua-dua keputusan yang didapati menunjukkan keadaan $|0010\rangle$ mendapat kiraan rawak yang paling tinggi berbanding dengan keadaan lain. Ibmq_qasm_simulator mendapat 3853 kiraan dari 8192 shot manakala ibmq_burlington mendapat 711 kiraan dari 8192 shot. Keberangskalian masing-masing ialah 47.03% dan 8.67%.

Keputusan yang didapati dari ibmq_qasm_simulator adalah lebih sesuai dengan ramalan algoritma grover. Keadaan yang ditetapkan akan mempunyai amplitud yang tertinggi dan akan mendapat kiraan rawak yang paling tinggi. Keadaan yang lain mempunyai amplitud negatif yang sama akan memberi kiraan yang kira-kira sama dan mempunyai kiraan yang rendah. Kiraan rawak yang paling tinggi dalam keadaan lain ialah $|1000\rangle$ yang mendapat 304 kiraan. Namun begitu, berbanding dengan kiraan 3853 yang didapati oleh $|0010\rangle$ perbezaan adalah sebanyak 3549 atau 43.32%.

Keputusan yang didapati dari ibmq_burlington adalah tidak sesuai dengan ramalan algoritma grover. Walaupun keadaan yang ditetapkan mendapat kiraan rawak yang paling tinggi tetapi kiraan keadaan lain adalah amat dekat. Kiraan yang paling tinggi dalam keadaan lain ialah $|0000\rangle$ yang mendapat 646 kiraan. Perbandingan dengan kiraan 711 yang didapati oleh $|0010\rangle$ perbezaan hanya 65 atau 0.79%.

Mengikuti keputusan yang didapati, komputer quantum sekarang masih terdapat perbezaan berbanding dengan bayangan kita untuk menghasilkan komputer quantum yang sempurna. Terdapat banyak masalah perlu diselesaikan lagi supaya komputer quantum dikomersialkan. Pengkomputer quantum masih di peringkat awal dan hasil yang didapati akan menjadi contoh kepada orang lain untuk membuat perbaikan.

6 KESIMPULAN

Secara keseluruhannya, kajian ini dijalankan bertujuan untuk meneroka IBM-Q yang merupakan komputer quantum berasaskan awan yang menggunakan teknologi komputer quantum yang sebenar.

Komputeran quantum yang semakin berkembang akan menjadi tren untuk pengkomputeran. Pengkomputeran quantum berasaskan awan merupakan langkah yang amat memperkembangkan pengkomputeran quantum. Pengkomputeran quantum berasaskan awan memberikan perkhidmatan yang membolehkan pengguna mengakses platform awan komputer quantum dan simulatonya. Kajian ini adalah untuk melakukan perbandingan komputer quantum sebenar dengan komputer quantum simulator. Pemahaman asas tentang komputer quantum seperti qubit, gate quantum dan litar quantum adalah amat penting.

7 RUJUKAN

University of Maryland, 2018. Semiconductor quantum transistor opens the door for photon-based computing. . [29th November 2019]

Lov K. Grover, 1996. A fast quantum mechanical algorithm for database search, *Proceedings, 28th Annual ACM Symposium on the Theory of Computing*, 212-221. [29th November 2019]

Lov K. Grover, 2001. From Schrodinger's equation to quantum search algorithm, *American Journal of Physics*, 69(7): 769-777, 2001. [29th November 2019]

Craig Gidney, 2013. Grover's Quantum Search Algorithm. http://twistedoakstudios.com/blog/Post2644_grovers-quantum-search-algorithm . [29th November 2019]

IBM Q Experience. <https://quantum-computing.ibm.com> [29th November 2019]

R. Feynman, "Quantum mechanical computers," *Science*, vol. 273, no.5278, pp. 1073-1078, 1996. [29th November 2019]

I. Buluta and F. Nori, "Quantum simulators," *Science*, vol. 326, no. 5949, pp. 108-111, 2009. [29th November 2019]

Jonathan Hui, 2018. QC — Programming with Quantum Gates (Single Qubit) https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-8996b667d256[20th November 2019]

Jonathan Hui, 2018. QC — Programming with Quantum Gates (2-Qubit operator) https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-2-qubit-operator-871528d136db [20th November 2019]

Qiskit Development Team, 2019. Qiskit documentation. <https://qiskit.org/documentation/install.html> [20th November 2019]

Fredkin gate. https://en.wikipedia.org/wiki/Fredkin_gate [20th November 2019]

Toffoli gate. https://en.wikipedia.org/wiki/Toffoli_gate [20th November 2019]

List of QC simulators. <https://www.quantiki.org/wiki/list-qc-simulators>[1st June 2020]

Cloud-based quantum computing. https://en.wikipedia.org/wiki/Cloud-based_quantum_computing [2nd June 2020]

IBM Q Experience. https://en.wikipedia.org/wiki/IBM_Q_Experience [2nd June 2020]

Jonathan Hui, 2018. QC — Grover's algorithm. https://medium.com/@jonathan_hui/qc-grovers-algorithm-cd81e61cf248 [3rd June 2020]

Kevin Ali, 2019. Analysis of Grover's Algorithm and Quantum Computation [15th June 2020]

Tom Tseng, 2015. Quantum Computation Lecture 4: Grover's Algorithm, *CMU 15-859BB, Fall 2015*. [3rd June 2020]

Vera Blomkvist Karlsson Philip Strömberg, 2018. 4-qubit Grover's algorithm implemented for the ibmqx5 architecture [3rd June 2020]

Grover algorithm with code, 2019. <https://quantumcomputinguk.org/tutorials/grovers-algorithm-with-code> [3rd June 2020]