



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.

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN PROGRAM

3.1 Metodologi Penelitian

1. Telaah Literatur

Pada tahap ini, dilakukan pengumpulan dan pembelajaran berbagai bentuk informasi yang dibutuhkan dalam penelitian ini. Informasi tersebut antara lain mengenai *quantum bit*, *quantum gate*, algoritma kuantum Grover, Quantum Computing Playground, dan Rigetti Forest SDK.

2. Perancangan dan Implementasi

Perancangan program implementasi algoritma kuantum Grover akan dijelaskan menggunakan *flowchart* dan rancangan tampilan antarmuka. Program dibuat menggunakan bahasa pemrograman Python dengan library pyQuil dan Rigetti Forest SDK. Tampilan antarmuka program dibuat menggunakan Flask.

3. Pengujian dan Evaluasi

Pada tahap ini, program yang telah dibuat akan diuji dengan metode *white box*. Evaluasi terhadap performa waktu eksekusi program *user time* dibandingkan dengan waktu eksekusi *user time* di Quantum Computing Playground.

4. Dokumentasi

Pada tahap ini, dibuat sebuah laporan sebagai dokumentasi dari penelitian dan pembuatan program secara bertahap, mulai dari pendahuluan hingga simpulan dan saran.

5. Konsultasi dan Penulisan

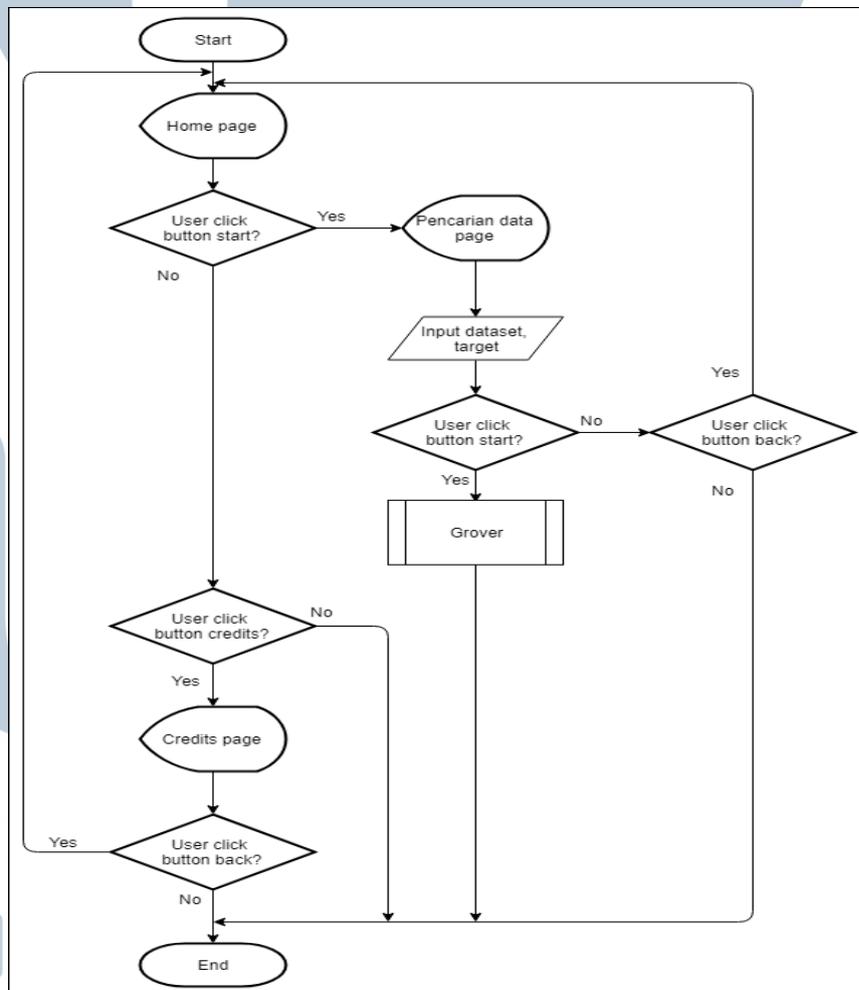
Konsultasi dan penulisan dilakukan bersama dosen pembimbing agar karya ilmiah ini dapat terselesaikan dengan baik.

3.2 Perancangan Program

Perancangan program ini dilakukan dengan menggambarkan diagram flowchart dan membuat rancangan tampilan antarmuka.

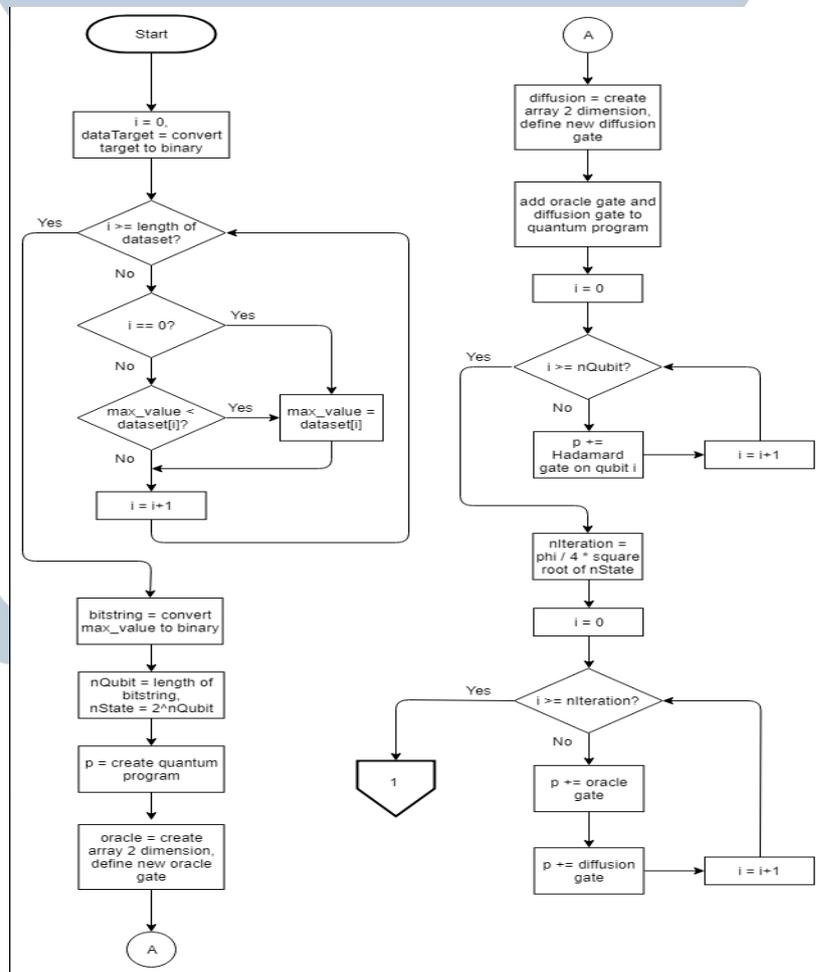
3.2.1 Flowchart

Berikut merupakan *flowchart* dari program ini.



Gambar 3.1 Flowchart utama program

Gambar 3.1 merupakan *flowchart* utama dari program implementasi algoritma kuantum Grover. *User* akan diarahkan ke halaman *home* pertama kali. *User* dapat masuk ke halaman pencarian data dengan menekan tombol *start* atau masuk ke halaman *credits* dengan menekan tombol *credits*. Di halaman pencarian data, *user* memasukkan *input dataset* dan *target* yang akan dicari. Tipe data yang bisa dimasukkan pada *dataset* adalah *integer* dan *char*. Untuk menjalankan pencarian, *user* dapat mengklik tombol *start* dan didapat hasil pencarian. *User* dapat kembali ke halaman *home* dengan menekan tombol *back*. Di halaman *credits*, terdapat tampilan informasi *credits* dan *user* dapat kembali ke halaman *home* dengan menekan tombol *back*.



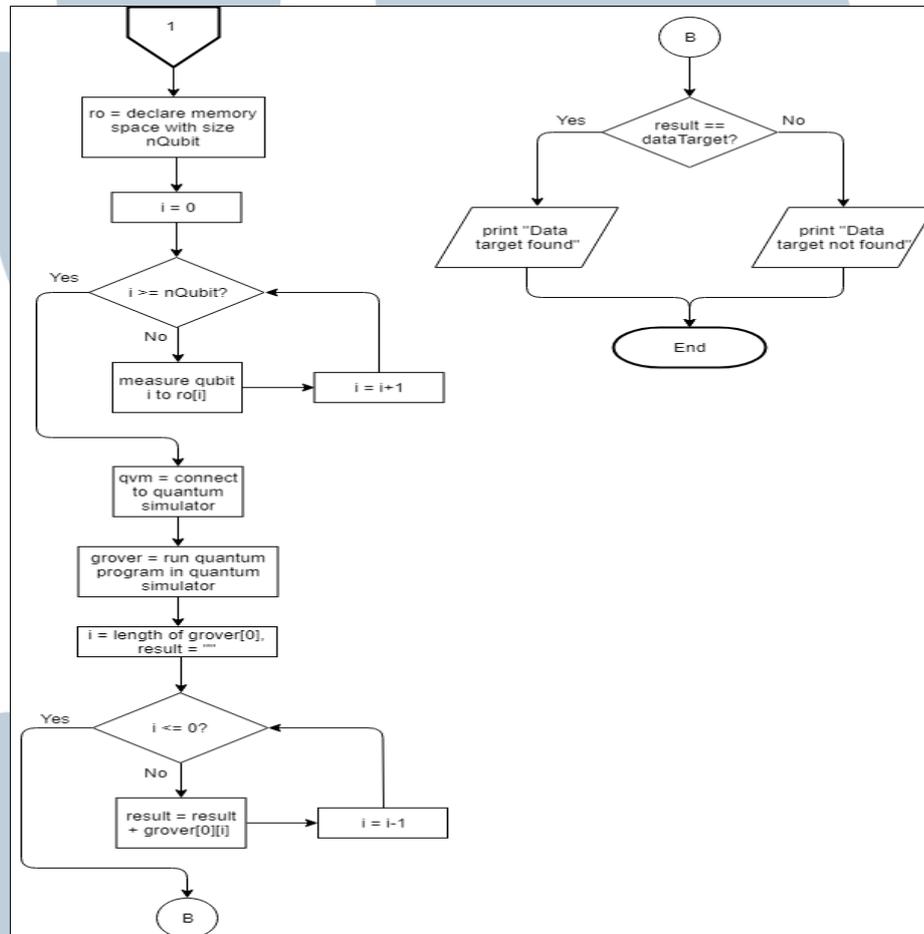
Gambar 3.2 Flowchart implementasi algoritma kuantum Grover (1)

Gambar 3.2 merupakan *flowchart* dari program implementasi algoritma kuantum Grover. Nilai dari *input target* akan diubah ke bentuk *binary* dan disimpan pada variabel *dataTarget*. Kemudian program akan mencari nilai terbesar dari *dataset* dengan melakukan perulangan pada setiap elemen data dan disimpan pada variabel *max_value*. Setelah perulangan selesai, nilai terbesar yang didapat diubah ke bentuk *binary* pada variabel *bitstring*. Jumlah *qubits* yang digunakan pada program ini bergantung pada panjang variabel *bitstring*. Sementara jumlah *state* yang mungkin adalah dua pangkat jumlah *qubits*.

Pembuatan objek Program diperlukan untuk membuat program kuantum di Forest. Kemudian, dibuat gerbang Oracle dalam bentuk *array* dua dimensi. *Array* ini berisi nilai 0 untuk indeks baris tidak sama dengan indeks kolom dan nilai 1 untuk indeks baris sama dengan indeks kolom. Tetapi, untuk indeks baris sama dengan indeks kolom dan nilai biner dari baris sama dengan variabel *dataTarget* dengan urutan terbalik, bernilai -1. Selain gerbang Oracle, dibuat juga gerbang Diffusion dalam bentuk *array* dua dimensi. *Array* ini berisi nilai $\frac{2}{nState}$ untuk indeks baris tidak sama dengan indeks kolom dan nilai $-1 + \frac{2}{nState}$ untuk indeks baris sama dengan indeks kolom. Selanjutnya gerbang Oracle dan Diffusion ditambahkan ke dalam objek Program agar bisa digunakan.

Langkah pertama dari algoritma kuantum Grover adalah melakukan gerbang Hadamard pada setiap *qubit*. Langkah ini membuat setiap *qubit* berada pada superposisi dan memiliki *amplitude* yang sama untuk setiap *state*. Langkah berikutnya adalah mencari jumlah iterasi yang diperlukan dalam proses *amplitude amplification*. Kemudian dilakukan perulangan sebanyak nilai dari variabel *nIteration*. Pada setiap perulangan, gerbang Oracle dilakukan untuk semua *qubit*,

kemudian gerbang Diffusion dilakukan juga untuk semua *qubit*. Gerbang Oracle berfungsi untuk membalikkan (*reverse*) nilai *amplitude* dari *state* yang sesuai, sedangkan *amplitude* dari *state* yang tidak sesuai tidak berubah. Gerbang Diffusion berfungsi untuk membalikkan setiap *amplitude* di sekitar rata-rata *amplitude*.



Gambar 3.3 Flowchart implementasi algoritma kuantum Grover (2)

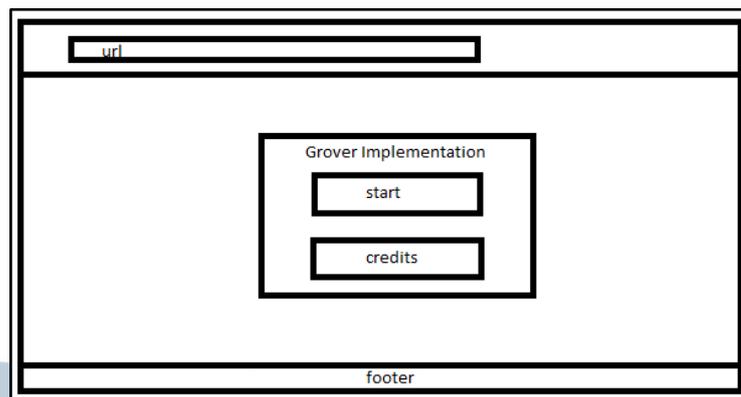
Gambar 3.3 merupakan *flowchart* lanjutan dari program implementasi algoritma kuantum Grover. Langkah terakhir dari algoritma kuantum Grover adalah dilakukan pengukuran (*measure*) pada setiap *qubit*. Untuk menampung hasil dari pengukuran *qubit*, dibuat variabel *ro* untuk mendeklarasikan ruang

memori sebesar variabel $nQubit$. Selanjutnya dilakukan perulangan sebanyak variabel $nQubit$ untuk mengukur setiap $qubit$.

Objek program yang telah dibuat menyimpan semua perintah-perintah yang telah dimasukkan. Untuk menjalankan objek program tersebut, dibuat terlebih dahulu koneksi ke *quantum simulator* (Quantum Virtual Machine). Setelah dibuat koneksinya, objek program dapat dieksekusi dan hasilnya disimpan dalam variabel *grover*. Hasil yang didapat berupa *array* bit 0 atau 1 dengan indeks 0 dimulai dari kanan. Variabel *result* menyimpan *bitstring* hasil dari perulangan variabel *grover* dalam urutan terbalik. Nilai dari variabel *result* dibandingkan dengan nilai dari variabel *dataTarget* apakah sesuai atau tidak.

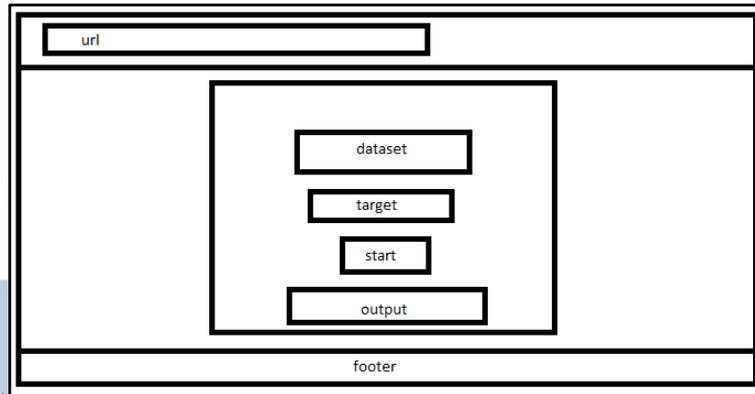
3.2.2 Rancangan Tampilan Antarmuka

Tampilan antarmuka program yang dibuat memiliki 3 halaman, yaitu halaman *home*, halaman pencarian data, dan halaman *credits*



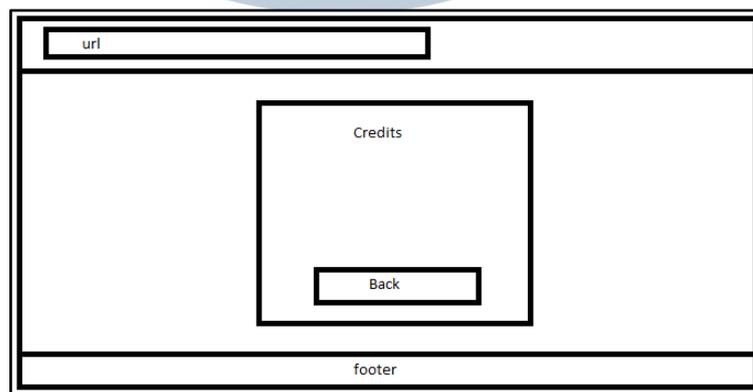
Gambar 3.4 Rancangan halaman *home*

Gambar 3.4 merupakan rancangan halaman *home*. Pada halaman ini terdapat 2 tombol, yaitu tombol *start* dan tombol *credits*. Tombol *start* akan mengarahkan ke halaman pencarian data ketika diklik. Tombol *credits* akan mengarahkan ke halaman *credits*.



Gambar 3.5 Rancangan halaman pencarian data

Gambar 3.5 merupakan rancangan halaman pencarian data. Pada halaman ini terdapat sebuah tombol *start* dan dua *input field*. *Input field* pertama untuk *user* memasukkan *dataset*. *Input field* kedua untuk *user* memasukkan data *target* yang ingin dicari. Ketika tombol *start* diklik, program akan menjalankan algoritma kuantum Grover berdasarkan *input user* dan menampilkan hasilnya dibawah tombol.



Gambar 3.6 Rancangan tampilan *credits*

Gambar 3.6 merupakan rancangan halaman *credits*. Pada halaman ini terdapat sebuah tombol untuk kembali ke halaman *home* jika diklik dan terdapat beberapa teks.

UNIVERSITAS
MULTIMEDIA
NUSANTARA