Algoritma Pencarian dalam Daftar Tak Terurut pada Komputasi Kuantum (Algoritma Grover)

Paramita[1], Ratna Ekasari Prihandini[2], Mia Kamayani Sulaeman[3]

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if14040@students.if.itb.ac.id[1],
if14043@students.if.itb.ac.id[2], if14052@students.if.itb.ac.id[3]

Abstrak

Komputer yang saat ini kita gunakan tampak bisa mengerjakan banyak tugas pada saat bersamaan, namun kenyataannya adalah prosessor hanya beralih secara cepat dari satu tugas ke tugas selanjutnya. Komputer paralel yang sesungguhnya akan sanggup melakukan banyak operasi pada saat yang bersamaan secara simultan, mencari solusi secara cepat dari sekian banyak kemungkinan. Para ilmuwan menyebutnya komputer kuantum, karena komputer ini bekerja menurut kaidah-kaidah ganjil dalam mekanika kuantum. Kaidah yang memicu timbulnya komputer kuantum adalah bahwa partikel elementer seperti proton, neutron, dan elektron dapat berada dalam dua atau lebih stata pada saat bersamaan. Partikel-partikel ini (secara teoritis) dapat merepresentasikan prosessing unit dalam suatu mesin sehingga lebih efisien dibandingkan komputer 'klasik' konvensional. Lov Grover (1996) telah menemukan algoritma kuantum (dikenal juga dengan algoritma Grover) untuk mencari di dalam suatu daftar tak terurut lebih cepat daripada komputer klasik. Dengan algoritma pencarian pada komputer klasik, suatu daftar dengan N item membutuhkan rata-rata N/2 langkah untuk menemukan solusi, sedangkan dengan komputer kuantum hanya dibutuhkan √N langkah untuk melakukan hal yang sama. Untuk daftar dengan jumlah item yang sangat besar, perbedaan ini sangat signifikan.

***Kata kunci**: komputer kuantum, algoritma grover, komputer klasik, algoritma pencarian*

1. Pendahuluan

Algoritma pencarian dalam suatu daftar merupakan algoritma pencarian paling dasar. Tujuannya adalah mencari sebuah elemen dari sebuah himpunan dengan suatu kunci (kemungkinan memuat informasi yang terkait dengan kunci). Algoritma pencarian yang paling sederhana adalah pencarian linear, yang mencari item secara berurutan. Kompleksitas algoritma pencarian linear ini adalah O(n), dengan n adalah banyaknya item dalam daftar. Algoritma Grover merupakan algoritma pencarian kuantum yang memungkinkan percepatan kuadrat dibandingkan dengan algoritma pencarian linear klasik untuk daftar tak terurut. Dengan demikian, algoritma Grover memungkinkan untuk dapat mencari suatu nama pada buku telepon yang berisi 1000.000 nama dengan 1000 kali percobaan saja, alih-alih 500 ribu percobaan.

2. Komputasi Kuantum

Komputasi kuantum adalah bidang studi yang mengkhususkan pada pengembangan teknologi komputer yang berdasarkan pada prinsip mekanika kuantum, yang menjelaskan sifat dan kelakuan energi dan materi yang berada pada level kuantum (atom dan subatom). Kemampuan komputasi pada komputer kuantum sangat jauh melebihi kemampuan superkomputer saat ini. Komputer kuantum memiliki kemampuan memproses yang sangat besar serta kemampuan untuk berada pada lebih dari satu stata dan mengerjakan banyak operasi pada saat bersamaan secara simultan.

3. Komputer Klasik dan Komputer Kuantum

Perbedaan antara komputer kuantum dengan komputer klasik adalah komputer klasik bergantung pada prinsip aljabar boolean, operasi dengan prinsip gerbang logika. Data diproses dalam biner (0 dan 1) dan hanya bisa dikerjakan satu bit (0 saja atau 1 saja) dalam satu waktu.

Sedangkan komputer kuantum dapat bekerja dengan 2 model gerbang logika: XOR dan Q01 (kemampuan untuk berganti dari 0 ke superposisi dari 0 atau 1, gerbang logika yang tidak bisa muncul di komputasi klasik). Dalam komputer klasik, jumlah data dihitung dengan bit, sedangkan pada komputer kuantum hal ini dilakukan dengan qubit (quantum bit). Prinsip dasar komputasi kuantum adalah bahwa sifat kuantum dari partikel dapat digunakan untuk mewakili data dan struktur data, dan bahwa prinsip mekanika kuantum dapat digunakan untuk melakukan operasi dengan data ini.

4. Algoritma Grover

Pada persoalan pencarian dengan *exhaustive search*, diberikan suatu fungsi *f(x)*,*x*=0,1...*(N*-1*)*, dimana *f(x)* adalah fungsi yang akan selalu menghasilkan 0 untuk semua *x,* kecuali satu nilai *x* yang akan menghasilkan 1. Tujuan dari persoalan ini adalah mencari nilai *x* sehingga *f(x)* = 1.

Ide dasar dari algoritma pencarian kuantum (algoritma Grover) adalah misalkan ada *N* buah status yang berkorespondensi dengan *N* item dalam suatu daftar tak terurut. Peluang untuk setiap status, bahwa status tersebut adalah yang dicari dalam daftar tersebut adalah *1/N*. Dengan prinsip mekanika kuantum, dimungkinkan untuk meningkatkan nilai peluang status yang dicari karena pengaruh status yang lain (status yang bukan status yang dicari), sehingga pada akhirnya status yang dicari akan memiliki nilai peluang tertinggi. Prinsip mekanika kuantum juga memungkinkan untuk berada dalam lebih dari satu status, dan melakukan lebih dari satu komputasi dalam waktu yang bersamaan.

Pada pencarian dengan probabilitas pada komputer klasik, peluang untuk status yang dicari akan meningkat sebesar *1/N* setiap kali iterasi pada kalang *for*, sehingga dengan iterasi sebanyak *N* kali, akan ditemukan solusi dengan nilai peluang tertinggi. Kompleksitas algoritma ini adalah *O(N).*Berikut ini adalah *pseudo code* nya:

```
kamus

function random(input N : integer)→ integer
{random(N) mengembalikan nilai acak antara 0
sampai (N-1) }

function f(input r:integer) → integer
{f(r)    adalah    fungsi    boolean    yang
mengembalikan 1 jika r adalah solusi yang
dicari dan 0 jika bukan solusi}

i,r,answer,N : integer

algoritma

   input ← N
   answer ← -1
   r ← random(N)
   for i = 0 to N do
      if f(r) = 1 then
         answer = r
      r ← random(N);

   output ← answer
```

Perbedaan paling mendasar antara pencarian probabilitas klasik dan algoritma pencarian kuantum adalah bahwa pencarian probabilitas klasik menggunakan nilai peluang yang harus bernilai positif, sedangkan pencarian kuantum menggunakan amplitudo, yang dapat bernilai positif atau negatif.

Berdasarkan prinsip mekanika kuantum, nilai amplitudo adalah sebesar akar dari peluang, sehingga amplitudo untuk setiap status menjadi sebesar akar dari *1/N*, yaitu *1/√N*.

Sesuai dengan penjelasan sebelumnya, amplitudo untuk status yang dicari akan meningkat sebesar *1/√N* setiap kali iterasi pada kalang *for*, sehingga dengan iterasi sebanyak √*N* kali, akan ditemukan solusi dengan nilai peluang tertinggi, dengan peluang sebesar kuadrat dari amplitudo. Oleh karena itu, kompleksitas algoritma pencarian kuantum ini adalah *O(√N).* Berikut ini adalah *pseudo code* nya:

```
kamus

function qrandom(input N:integer, r:integer)
→ integer
{qrandom(N,r)  mengembalikan  nilai  random
antara 0 sampai (N-1)}

function f(input r:integer) → integer
{f(r)  adalah  fungsi  yang  mengembalikan  1
jika r adalah solusi yang dicari dan 0 jika
bukan solusi}
i,r,N : integer

algoritma

   input ← N

   r ← qrandom(N,0)
   {eta adalah √N}
   for i = 0 to eta do
      if f(r) = 1 then
         invert_phase()
      {f(r) dievaluasi selama program
      berjalan tanpa observasi dari
      luar, karena akan mengganggu
      sistem}

      r ← qrandom(N,r);

      if r=0 then
         invert_phase()

      r ← qrandom(N,r)

   output ← r
```

Fungsi `qrandom(N,r)` di atas merupakan fungsi pembangkit bilangan random yang juga merupakan *Walsh-Hadamard Transformation.* Misalkan `qrandom(N,r)` membangkitkan sebuah nilai q, antara 0 sampai (*N*-1) dengan amplitudo bernilai 1/√*N* for each number. Tanda dari amplitudo ditentukan dengan perhitungan berikut. Bilangan q dan r direpresentasikan dengan biner. Jika jumlah dari posisi bit 1 yang berada pada q dan juga r adalah genap, maka tandanya positif. Sebaliknya, jika jumlahnya ganjil, maka tandanya negatif. Sebagai contoh misal q=01110101 dan r=10110111, maka posisi bit 1 yang sama pada kedua bilangan tersebut (dihitung dari *least significant bit*) adalah pada posisi 1, 3, 5, dan 6. Jumlahnya adalah 4 buah maka tanda dari amplitudonya adalah positif.

Berikut ini gambaran yang lebih lengkap mengenai algoritma pencarian kuantum.

Misalkan dalam suatu persoalan terdapat sejumlah N status yaitu $S_1,S_2,...S_N$, dimana $N = 2^n$, status-status ini direpresentasikan dengan string n bit. Di antara status-status tersebut terdapat status unik S_v yang memenuhi $C(S_v)=1$, sedemikian hingga S_v merupakan status yang dicari, sementara status-status lain S, $C(S)=0$. Persoalannya adalah bagaimana mengidentifikasi status S_v, yang dijabarkan sebagai berikut:

(i) inisialisasi sistem menjadi distribusi
$\left(\frac{1}{\sqrt{N}},\frac{1}{\sqrt{N}},\frac{1}{\sqrt{N}},...,\frac{1}{\sqrt{N}}\right)$
Distribusi ini diturunkan dengan kompleksitas $O(\log N)$.
(ii) ulangi operasi berikut sebanyak $\sqrt{N}$ langkah :
(a) Misalkan sistem berada dalam status S:
jika $C(S) = 1$, rotasikan fase sebanyak π rad.
jika $C(S) = 0$, abaikan.

(b) Aplikasikan transformasi difusi D yang didefinisikan oleh matriks D sbb:

$$D_{ij} = \frac{2}{N} \; if \; \mathrm{i} \neq \mathrm{j} \qquad (1)$$

dan

$$D_{ii} = -1 + \frac{2}{N} \qquad (2)$$

D bisa diimplementasikan sebagai:
$D = WRW$, dimana R adalah matriks rotasi dan W adalah matriks transformasi *Walsh-Hadamard* yang didefinisikan sbb:

$$R_{ij} = 0 \; if \; \mathrm{i} \neq \mathrm{j} \qquad (3)$$

$$R_{ii} = 1 \; if \; \mathrm{i} = 0, \; R_{ii} = -1 \; if \; \mathrm{i} \neq 0 \qquad (4)$$

$$W_{ij} = 2^{-n/2}(-1)^{\bar{i}\cdot\bar{j}} \qquad (5)$$

dimana $\bar{i}$ adalah representasi biner dari i dan $\bar{i} \cdot \bar{j}$ merupakan hasil kali titik dari dua string n bit, $\bar{i}$ dan $\bar{j}$.

(iii) Ambil status yang dicari. Jika $C(S_v) = 1$ maka ada status unik S_v sedemikian hingga S_v merupakan status yang dicari dengan peluang $\geq ½$.

5. Contoh penerapan Algoritma Grover

Anggaplah bahwa Anda ingin menemukan suatu nama dalam buku telepon yang hanya memiliki empat aran (entri), maka $N = 4$. Pada umumnya, *eta* (jumlah iterasi pada algoritma pencarian kuantum), berkisar antara $0.5\sqrt{N}$ and $0.8\sqrt{N}$. Untuk $N = 4$, maka *eta* bernilai 1 sehingga iterasi hanya berlangsung satu kali. Fungsi `qrandom(4,r)` akan membangkitkan bilangan q antara 0 sampai 3 dengan amplitudo bernilai $\pm 1/\sqrt{4}$ atau ±0.5. Berikut ini adalah daftar nilai amplitudo untuk bilangan q dan r dengan tanda sesuai dengan perhitungan yang telah dijelaskan sebelumnya:

q \ r	0	1	2	3
0	0.5	0.5	0.5	0.5
1	0.5	-0.5	0.5	-0.5
2	0.5	0.5	-0.5	-0.5
3	0.5	-0.5	-0.5	0.5

Tabel 1 Tabel amplitudo untuk setiap q dan r

Sesuai dengan algoritma pencarian kuantum yang telah dijelaskan sebelumnya, maka berikut ini adalah gambaran langkah-langkah proses pencarian dengan algoritma kuantum, dengan asumsi aran yang dicari berada pada indeks ke-2 (aran ke-3):

Steps of the *quantum_main()* Program	Amplitude Vector for *r*
r = qrandom(4,0);	Step 1 (0.5, 0.5, 0.5, 0.5)
if (!(r) == 1) invert_phase(r);	Step 2 (0.5, 0.5, –0.5, 0.5)
r = qrandom(4,r);	Step 3 (0.5, –0.5, 0.5, 0.5)
if (r==0) invert_phase();	Step 4 (–0.5, –0.5, 0.5, 0.5)
r = qrandom(4,r);	Step 5 (0.0, 0.0, –1.0, 0.0)

(i) Pada awalnya dengan fungsi `qrandom(4,0)` amplitudo yang terbentuk untuk setiap nilai r adalah +0.5, sehingga himpunan amplitudonya v = [0.5, 0.5, 0.5, 0.5].

(ii) Diasumsikan nilai r yang dihasilkan pada langkah (i) adalah 2, maka fungsi `invert_phase()` tereksekusi, dan v[2] menjadi -0.5. Pada langkah ini diperoleh v = [0.5, 0.5, -0.5, 0.5].

(iii) `qrandom(4,r)` tidak hanya membangkitkan bilangan acak antara 0 sampai 3, tetapi juga menghitung amplitudo sesuai dengan amplitudo sebelumnya. Bilangan yang dibangkitkan oleh `qrandom(4,r)`, misalnya q, bernilai antara 0 sampai 3.
Untuk q = 0, maka ada 4 kemungkinan jalur yang dilalui :

a) Sebelum langkah (iii), r = 0, setelah langkah(iii), r tetap 0.
Hasil kali antara amplitudo v[0], dengan amplitudo saat q = 0 dan r = 0 (dilihat dari Tabel 1) adalah: 0.5 x 0.5 = 0.25.

b) Sebelum langkah (iii), r = 1, setelah langkah(iii), r menjadi 0.
Hasil kali antara amplitudo v[1], dengan amplitudo saat q = 0 dan r = 1 (dilihat dari Tabel 1) adalah: 0.5 x 0.5 = 0.25.

c) sebelum langkah (iii), r = 2, setelah langkah(iii), r menjadi 0.
Hasil kali antara amplitudo v[2], dengan amplitudo saat q = 0 dan r = 2 (dilihat dari Tabel 1) adalah: -0.5 x 0.5 = -0.25.

d) sebelum langkah (iii), r = 3, setelah langkah(iii), r menjadi 0.
Hasil kali antara amplitudo v[2], dengan amplitudo saat q = 0 dan r = 3 (dilihat dari Tabel 1) adalah
0.5 x 0.5 = 0.25.

Maka untuk q = 0, amplitudonya adalah
0.25 + 0.25 - 0.25 + 0.25 = 0.5.

Dengan melakukan perhitungan yang sama untuk q = 1, q = 2, dan q = 3, diperoleh amplitudonya adalah -0.5, 0.5 dan 0.5. Pada langkah ini diperoleh v = [0.5, -0.5, 0.5, 0.5].

(iv) Diasumsikan nilai r yang dihasilkan pada langkah (iii) adalah 0, maka fungsi `invert_phase()` tereksekusi, dan v[0] menjadi -0.5. Pada langkah ini diperoleh v = [-0.5, -0.5, 0.5, 0.5].

(v) Pada langkah ini dikerjakan operasi yang sama seperti pada langkah (iii), yaitu menghitung amplitudo untuk semua kemungkinan bilangan yang dibangkitkan oleh fungsi `qrandom(4,r)`. Pada langkah ini diperoleh v = [0.0, 0.0, -1.0, 0.0].

Dari himpunan amplitudo v yang diperoleh pada langkah (v) dapat diketahui peluang untuk:
r = 0 : $(0.0)^2 = 0$
r = 1 : $(0.0)^2 = 0$
r = 2 : $(-1.0)^2 = 1$
r = 3 : $(0.0)^2 = 0$

Dari peluang di atas dapat dilihat bahwa aran yang dicari terletak pada indeks ke-2.

6. Kesimpulan

Karena kecepatan dan kemampuannya, algoritma pencarian Grover sangat mungkin menjadi komponen kunci dalam peranti lunak masa depan. Dengan kompleksitas algoritma sebesar $O(\sqrt{N})$, jelas algoritma pencarian Grover yang merupakan algoritma kuantum jauh lebih baik daripada algoritma pencarian klasik dalam daftar tak terurut, dengan kompleksitas algoritma $O(N)$.

Meski masih dalam bentuk kertas kerja, prospek komputer kuantum sangat menggiurkan, contohnya, suatu algoritma yang dapat memfaktorkan bilangan sepanjang 140 digit dengan kecepatan semilyar (10^9) kali lebih cepat dibanding yang ditawarkan oleh metode nonkuantum terbaik yang kita kenal sekarang, sebuah *search engine* yang sanggup memeriksa setiap 'sudut dan celah' di Internet dalam waktu kurang dari setengah jam**,** atau sebuah pemecah kode '*brute-force*' yang dapat memecahkan transmisi DES dalam tempo 5 menit.

7. Referensi

[1] Grover, L.K., *A fast quantum mechanical algorithm for database search*, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212

[2] Grover, L.K., *A fast quantum mechanical algorithm for database search (alih bahasa Eko Sujatmiko),* Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212.

[3] Munir, Rinaldi, *Strategi Algoritmik.* Departemen Teknik Informatika, Institut Teknologi Bandung, 2005.

[4] http://en.wikipedia.org

[5] http://id.wikipedia.org

[6] http://www.djj.com