Analisis Mengenai Implementasi Vigenere Cipher pada Bahasa Pemrograman C++

Rifky Hamdani / 13508024
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if18024@students.if.itb.ac.id

***Abstract*—Vigenere Chiper merupakan metode enkripsi dengan cara mengganti karater-karakter yang ada dengan karakter yang lain sesuai dengan kunci yang diberikan. Pada makalah ini akan dicoba beberapa cara implementasi Vigenere Chiper pada Bahasa Pemrograman C++. Dalam implementasi Vigenere Chiper terdapat beberapa cara yaitu menggunakan rumus, *array*, dan *vector*. Dari beberapa cara implementasi tersebut akan dibandingkan cara mana yang paling baik untuk mengimplementasikan Vigenere Chiper.**

***Kata Kunci*—Vigenere, C++, rumus, *array*, *vector*.**

I. PENDAHULUAN

Vigenere Cipher termasuk dalam kriptografi klasik. Seperti Kriptografi klasik yang lain Vigenere Chiper merupakan kriptografi yang berbasis karakter. Yang dimaksud dengan berbasis karakter adalah setiap operasi yang ada dilakukan pada karakter.

Bahasa Pemrogaraman C++ merupakan bahasa pemrograman yang populer dan banyak digunakan. Terdapat beberapa faktor bahasa pemrograman C++ banyak digunakan.

Pertama, bahasa pemrograman C++ merupakan bahasa pemrograman yang di *compile*(*native*) sehingga memiliki performa yang lebih baik dibanding bahasa pemrograman yang diinterpretasi seperti pada Javascript atau bahasa pemrograman *compiled on the fly* (*Just in Time*) seperti pada Java atau C#.

Kedua, bahasa pemrograman C++ memiliki *library* (*Standard Template Library*) yang lengkap sehingga tidak perlu mengimplementasikan struktur data dari awal.

Ketiga, bahasa pemrograman C++ dapat digunakan secara prosedural maupun berorientasi objek ataupun keduanya sesuai dengan kebutuhan.

Implementasi Vigenere Chiper pada bahasa pemrograman C++ dapat dibuat dengan tiga cara yaitu dengan menggunakan rumus, kedua menggunakan *array* dua dimensi, dan yang terakhir menggunakan *vector*.

II. IMPLEMENTASI DENGAN RUMUS

Vigenere Chiper dapat diimplementasikan dengan menggunakan rumus. Rumus tersebut berupa penambahan karakter plainteks dengan karakter kunci kemudian di*mod* dengan 26.

Implementasi dengan rumus merupakan cara yang paling mudah untuk membuat Vigenere Chiper. Hal tersebut disebabkan tidak perlu dibuatnya tabel substitusi.

Kode program yang menggunakan rumus dalam implementasi Vigenere Chiper ditunjukan pada kode dibawah ini:

```
string Equation(string plain, string key) {
    string temp = "";
    char temp_char;
    int key_position = 0;
    for (unsigned int i = 0; i < plain.length();
++i) {
      temp_char = int2char((char2int(plain.at(i)) +
char2int(key.at(key_position % key.length()))) %
26);
      key_position++;
      temp += temp_char;
    }
    return temp;
}
```

III. IMPLEMENTASI DENGAN ARRAY DUA DIMENSI

Selain dengan rumus, Vigenere Cipher juga dapat diimplementasikan dengan menggunakan *array* dua dimensi. Pada cara ini harus dibuat tabel substitusi terlebih dahulu.

Implementasi Vigenere Chiper dengan cara ini lebih susah dari pada menggunakan rumus. Akan tetapi, cara ini termasuk sederhana. Awalnya dibuat sebuah *array* dua dimensi dengan panjang 26 dan lebar 26. Kemudian *array* tersebut diisi dengan angka yang sesuai dengan tabel substitusi yang ada pada tabel substitusi Vigenere Chiper.

Setelah tabel substitusi selesai dibuat kita dapat mendapatkan chiperteks dengan mengambil nilai dari

array dengan lebar dari plainteks dan panjang dari kunci atau sebaliknya.

Kode program yang menggunakan *array* dua dimensi dalam implementasi Vigenere Chiper ditunjukan pada kode dibawah ini:

```
string Array2D(string plain, string key, int Tabel[26][26]) {
  string temp = "";
  char temp_char;
  int key_position = 0;
  for (unsigned int i = 0; i < plain.length(); ++i)
{
    temp_char                                          =
int2char(Tabel[char2int(plain.at(i))][char2int(key.a
t(key_position % key.length()))]);
    key_position++;
    temp += temp_char;
  }
  return temp;
}
```

IV. IMPLEMENTASI DENGAN VECTOR

Bahasa Pemrograman C++ sangat populer karena memiliki library yang cukup lengkap. Library C++ atau yang disebut juga dengan Standard Template Library memiliki salah satu elemen yang bernama *vector*.

Kita dapat mengimplementasikan Vigenere Chiper dengan *vector*. Dalam hal ini *vector* akan menggantikan fungsi *array* dua dimensi sebagai tabel substitusi.

Implementasi Vigenere Chiper menggunakan *vector* mirip dengan implementasi menggunakan *array* dua dimensi.

Kode program yang menggunakan *array* dua dimensi dalam implementasi Vigenere Chiper ditunjukan pada kode dibawah ini:

```
string Vector(string plain, string key,
vector<vector<int>> Tabel) {
  string temp = "";
  char temp_char;
  int key_position = 0;
  for (unsigned int i = 0; i < plain.length(); ++i)
{
    temp_char                                          =
int2char(Tabel.at(char2int(plain.at(i))).at(char2int
(key.at(key_position % key.length()))));
    key_position++;
    temp += temp_char;
  }
  return temp;
}
```

V. PENGUJIAN

Pada pengujian ini digunakan komputer dengan spesifikasi:

CPU : Intel® Core™ i5-2410M CPU @ 2.30GHz
Memory(RAM): 4.00 GB
Operating System: Windows 7 Professional 64-bit
IDE: Visual Studio 2010 Ultimate
Bahasa Pemrograman: C++.

Kode program pengujian dalam implementasi Vigenere Chiper ditunjukan pada kode dibawah ini:

```
int char2int(char a) return (int)(a - 'A');
char int2char(int i) return (char)(i + 65);
string Equation(string plain, string key);
string Array2D(string plain, string key, int
Tabel[26][26]);
string Vector(string plain, string key,
vector<vector<int>> Tabel);

int main() {
  clock_t start_equation, end_equation, start_array,
end_array, start_vector, end_vector;
  string plaintext = "", chiper1, chiper2, chiper3,
line = "";
  ifstream fileteks("fileteks.txt");
  if(fileteks.is_open()) {
    while(fileteks.good()){
      getline(fileteks,line);
      plaintext += line;
    }
    fileteks.close();
  } else {
    cout << "can't open file" << endl;
  }

  //EQUATION
  start_equation = clock();
  chiper1 = Equation(plaintext,"VIGENERE");
  end_equation = clock();

  //TABLE
  start_array = clock();
  int table[26][26];
  for(unsigned int i = 0; i < 26; ++i) {
    for(unsigned int j = 0; j < 26; ++j) {
      table[i][j] = (i + j) % 26;
    }
  }
  chiper2 = Array2D(plaintext,"VIGENERE",table);
  end_array = clock();

  //VECTOR
  start_vector = clock();
  vector<int> table1;
  vector<vector<int>> table2;

  for(unsigned int i = 0; i < 26; ++i) {
    for(unsigned int j = 0; j < 26; ++j) {
      table1.push_back((i + j) % 26);
    }
    table2.push_back(table1);
    table1.clear();
  }
  chiper3 = Vector(plaintext,"VIGENERE",table2);
  end_vector = clock();
```

```
  //SHOW EXECUTION TIME
  cout << endl << "Equation Time: " << ((double)(end_equation - start_equation))/CLOCKS_PER_SEC << endl;
  cout << endl << "Table Time: " << ((double)(end_array - start_array))/CLOCKS_PER_SEC << endl;
  cout << endl << "Vector Time: " << ((double)(end_vector - start_vector))/CLOCKS_PER_SEC << endl;

  system("pause");
  return 0;
}
```

Program di atas menggunakan fungsi int2char untuk mengubar interger menjadi karakter dan fungsi char2int yang mengubah karakter menjadi integer. Selain itu juga digunakan fungsi-fungsi implementasi Vigenere Chiper yang sudah ditunjukan pada bab-bab sebelumnya.

Agar data yang di dapat lebih valid pengujian dilakukan dengan beberapa jumlah karakter dan beberapa kali dengan jumlah karakter yang sama.

Pada pengujian yang dilakukan memakai teks dengan karakter berjumlah 3302, 6604, 13208, 26416, 52832, 105664, 211328, 422656, 845312, dan 845312. Selain itu, pada tiap teks dilakukan pengujian sebanyak lima kali lalu diambil rataannya.

Untuk menyederhanakan operasi yang ada, plaintext yang dipakai hanya berisi huruf alfabet kapital tanpa spasi. Penyederhaan ini dilakukan agar mengurangi operasi “if” dalam program yang dapat menyebabkan tidak akuratnya data yang dihasilkan.

Uji 1 dengan 3302 Karakter

I	
Formula	0.002
Array	0.002
Vector	0.004
II	
Formula	0.002
Array	0.002
Vector	0.005
III	
Formula	0.002
Array	0.002
Vector	0.004
IV	
Formula	0.002
Array	0.002
Vector	0.004
V	
Formula	0.003
Array	0.003
Vector	0.005
Rataan	
Formula	0.0022
Array	0.0022
Vector	0.0044

Uji 2 dengan 6604 Karakter

I	
Formula	0.012
Array	0.007
Vector	0.01
II	
Formula	0.01
Array	0.009
Vector	0.014
III	
Formula	0.004
Array	0.004
Vector	0.007
IV	
Formula	0.004
Array	0.004
Vector	0.007
V	
Formula	0.013
Array	0.007
Vector	0.012
Rataan	
Formula	0.0086
Array	0.0062
Vector	0.01

Uji 3 dengan 13208 karakter

I	
Formula	0.011
Array	0.009
Vector	0.014
II	
Formula	0.008
Array	0.008
Vector	0.011
III	
Formula	0.007
Array	0.008
Vector	0.011
IV	
Formula	0.009
Array	0.008
Vector	0.012
V	
Formula	0.009
Array	0.007
Vector	0.012
Rataan	
Formula	0.0088
Array	0.008
Vector	0.012

Uji 4 dengan 26416 karakter

I	
Formula	0.016
Array	0.015
Vector	0.021
II	
Formula	0.016
Array	0.016
Vector	0.024
III	
Formula	0.025
Array	0.016
Vector	0.02
IV	
Formula	0.024
Array	0.016
Vector	0.02
V	
Formula	0.02
Array	0.017
Vector	0.021
Rataan	
Formula	0.0202
Array	0.016
Vector	0.0212

Uji 5 dengan 52832 karakter

I	
Formula	0.016
Array	0.015
Vector	0.021
II	
Formula	0.016
Array	0.016
Vector	0.024
III	
Formula	0.025
Array	0.016
Vector	0.02
IV	
Formula	0.024
Array	0.016
Vector	0.02
V	
Formula	0.02
Array	0.017
Vector	0.021
Rataan	
Formula	0.0202
Array	0.016
Vector	0.0212

Uji 6 dengan 105664 karakter

I	
Formula	0.063
Array	0.063
Vector	0.079
II	
Formula	0.088
Array	0.079
Vector	0.081
III	
Formula	0.062
Array	0.062
Vector	0.076
IV	
Formula	0.064
Array	0.062
Vector	0.076
V	
Formula	0.065
Array	0.063
Vector	0.078
Rataan	
Formula	0.0684
Array	0.0658
Vector	0.078

Uji 7 dengan 211328 karakter

I	
Formula	0.127
Array	0.124
Vector	0.152
II	
Formula	0.124
Array	0.124
Vector	0.15
III	
Formula	0.126
Array	0.125
Vector	0.154
IV	
Formula	0.127
Array	0.125
Vector	0.154
V	
Formula	0.127
Array	0.126
Vector	0.154
Rataan	
Formula	0.1262
Array	0.1248
Vector	0.1528

Uji 8 dengan 422656 karakter

I	
Formula	0.251
Array	0.245
Vector	0.299
II	
Formula	0.254
Array	0.246
Vector	0.303
III	
Formula	0.251
Array	0.247
Vector	0.309
IV	
Formula	0.252
Array	0.248
Vector	0.302
V	
Formula	0.253
Array	0.247
Vector	0.307
Rataan	
Formula	0.2522
Array	0.2466
Vector	0.304

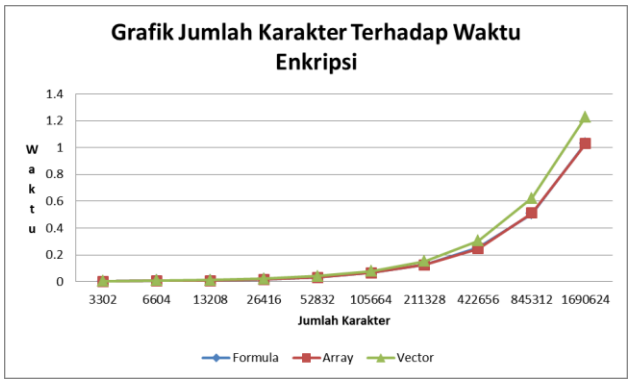
Uji 9 dengan 845312 karakter

I	
Formula	0.557
Array	0.575
Vector	0.703
II	
Formula	0.494
Array	0.497
Vector	0.602
III	
Formula	0.5
Array	0.493
Vector	0.61
IV	
Formula	0.494
Array	0.498
Vector	0.596
V	
Formula	0.494
Array	0.491
Vector	0.599
Rataan	
Formula	0.5078
Array	0.5108
Vector	0.622

Uji 10 dengan 845312 karakter

I	
Formula	1.142
Array	0.993
Vector	1.392
II	
Formula	0.984
Array	0.981
Vector	1.194
III	
Formula	0.984
Array	0.982
Vector	1.182
IV	
Formula	1.062
Array	1.196
Vector	1.182
V	
Formula	0.988
Array	0.992
Vector	1.194
Rataan	
Formula	1.032
Array	1.0288
Vector	1.2288

Dari data di atas dapat dibuat grafik waktu enkripsi terhadap jumlah karakter yang digunakan



Pada hasil pengujian dapat dilihat bahwa enkripsi yang paling cepat diperoleh pada implementasi Vigenere Chiper dengan menggunakan *array* dua dimensi kemudian dengan menggunakan rumus, dan yang terakhir adalah menggunakan *vector* dalam implementasi.

Pengujian dilakukan dengan menggunakan struktur data clock_t. Struktur data clock_t merupakan satuan tik waktu tiap detik. Untuk mendapatkan waktu yang diperlukan suatu program untuk melakukan eksekusi diambil dua buah tik waktu yaitu *start* dan *end*. Dari ke dua data tersebut kemudian dicari selisihnya dan kemudian dibagi dengan `CLOCKS_PER_SEC` untuk menghasilkan waktu dalam satuan detik.

VI. KESIMPULAN

Dari hasil pengujian yang telah dilakukan, dapat ditarik kesimpulan bahwa:

- Untuk meningkatkan performa dari implementasi Vigenere Chiper pada bahasa pemrograman C++ kiata harus membuat table substitusi dengan menggunakan *array* dua dimensi.
- Penggunaan *Standard Template Library* pada bahasa pemrograman C++ tidak menjamin adanya peningkatan performa pada program. Bahkan penggunaan STL dapat menurunkan performa pada program.

REFERENSI

[1] Munir, Rinaldi. 2005. Diktat Kuliah IF5054 Kriptografi .Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[2] http://en.wikipedia.org/wiki/Vigenere_cipher (diakses pada tanggal 10 Maret 2012).

[3] http://www.cplusplus.com/reference/clibrary/ctime/ (diakses pada tanggal 10 Maret 2012)

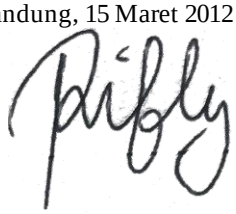
[4] http://www.cplusplus.com/reference/clibrary/ctime/clock_t/ (diakses pada tanggal 10 Maret 2012)

[5] http://www.cplusplus.com/reference/clibrary/ctime/clock/ (diakses pada tanggal 10 Maret 2012)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Maret 2012



Rifky Hamdani / 13508024