

Aplikasi Sederhana Pattern Matching dengan Algoritma Brute Force pada Validasi Suatu Teks

Rheno Manggala Budiasa - 13506119

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
e-mail: if16119@students.if.itb.ac.id

ABSTRAK

Banyak sekali program aplikasi yang dibuat oleh pengembang perangkat lunak untuk berbagai keperluan. Saat ini, mulai dari Industri besar hingga perbankan semuanya memerlukan pengolahan data yang cukup handal sebagai alat penunjang bisnis mereka. Salah satu aplikasi yang paling terkenal yaitu DBMS (Database Management Systems). Ada beberapa DBMS yang sudah cukup kita kenal seperti MYSQL, ORACLE, Microsoft SQL Server, DB2 dan lain-lain. Semua program tersebut dibuat untuk mengolah sekumpulan data. Dalam mengakses dan mengolahnya program-program tersebut menggunakan beberapa operasi. Operasi-operasi yang sudah cukup kita kenal antara lain SELECT, INSERT, UPDATE dan lain-lain. Sekumpulan operasi tersebut kita kenal sebagai *Query*. Penyusunan suatu *Query* akan melibatkan beberapa algoritma. Beberapa Algoritma yang sudah kita kenal antar lain *Brute Force*, Boyer-Moore, KMP dan lainnya. Selain beberapa operasi yang sudah kita kenal, beberapa DBMS juga mempunyai keunggulan tertentu seperti pengelolaan memori yang baik sampai akses data tercepat. Seiring berkembangnya teknologi para vendor perangkat lunak juga mulai mengembangkan aplikasi yang dapat mengolah data sedemikian banyaknya menjadi suatu Informasi tertentu tanpa harus mengetahui *Query*.

Kata kunci: Program, Aplikasi, Pengembang, Perangkat Lunak, DBMS, data, operasi, *Query*, algoritma, memori, akses, teknologi, Informasi.

I. PENDAHULUAN

I.1 String

Pattern Matching atau Pencocokan Pola adalah suatu metode yang digunakan untuk mencocokkan suatu pola tertentu (kumpulan huruf) dengan suatu kumpulan kata

(teks) atau *string*. Pada bidang sains komputer metode *pattern matching* sangat banyak digunakan antar lain Editor Teks, Mesin Pencari Web, Analisis Gambar dan lain-lain. *String* dapat kita asumsikan sebagai kumpulan dari beberapa karakter yang membentuk suatu kesatuan. *String* dapat juga kita sebut sebagai senarai dari karakter. Karena sifatnya senarai maka *string* ini mempunyai indeks dan tentunya panjang. Secara notasi string dapat kita tuliskan sebagai :

$S[i..j]$ = string S dengan karakter-karakter bagiannya adalah S[i] sampai S[j]

m = Panjang String (j - i + 1)

Panjang String (m) adalah jumlah semua bagian karakter yang terdapat pada string S.

I.2 Algoritma *Brute Force*

Algoritma *Brute Force* adalah algoritma yang lempang atau langsung (to the point/straight forward) ke pokok masalah. Algoritma ini biasanya tidak memerlukan teori khusus untuk mengimplementasikannya. Algoritma ini sering juga disebut Algoritma sapu jagad karena hampir semua persoalan pemrograman bisa diselesaikan dengan algoritma ini.

II. METODE

Seperti yang telah dijelaskan pada bab pendahuluan, String dapat kita asumsikan juga sebagai kumpulan dari senarai, karena itu setiap elemen indeks dari string tersebut merupakan himpunan bagian dari string.

T:

a	n	d	r	e	w
---	---	---	---	---	---

T[0] = a T[5] = w
T[1] = n
T[2] = d
T[3] = r
T[4] = e

Secara sederhana pencocokkan string secara brute force dapat kita gambarkan sebagai berikut, misalnya terdapat sebuah teks T dengan *pattern* P :

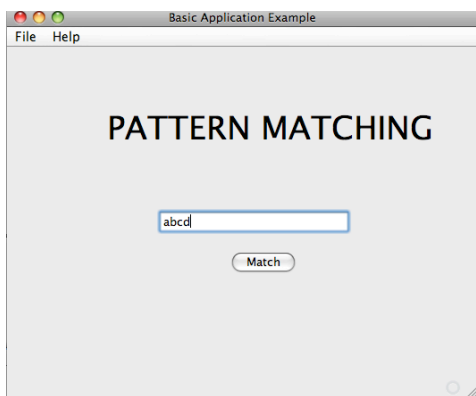


Dapat kita lihat pada contoh di atas *pattern* P akan maju satu karakter terhadap teks T sampai P sesuai dengan salah satu segmen T (substring T).

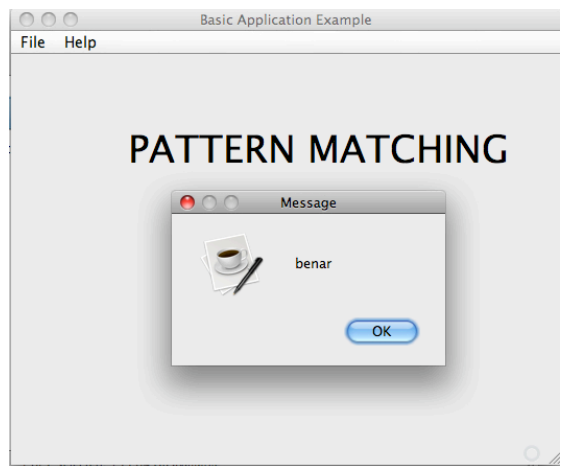
Berikut ini akan kita lihat bagaimana program *Pattern Matching* yang dibuat menggunakan bahasa java. Pertama-tama kita asumsikan bahwa teks berada pada komponen **jTextField1**, sedangkan *pattern* kita anggap berada pada program (lihat bagian **if**). Berikut potongan program *pattern matching* secara *brute force* :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent ev) {
try{
if((jTextField1.getText().charAt(0)=='a')&&(jTextField1.getText().charAt(1)=='b')
&&(jTextField1.getText().charAt(2)=='c')&&(jTextField1.getText().charAt(3)=='d'))
{
JOptionPane.showMessageDialog(JOptionPane.getRootFrame(),
"benar");
}
else
JOptionPane.showMessageDialog(JOptionPane.getRootFrame(),
"salah");
}catch(Exception e){}
}
```

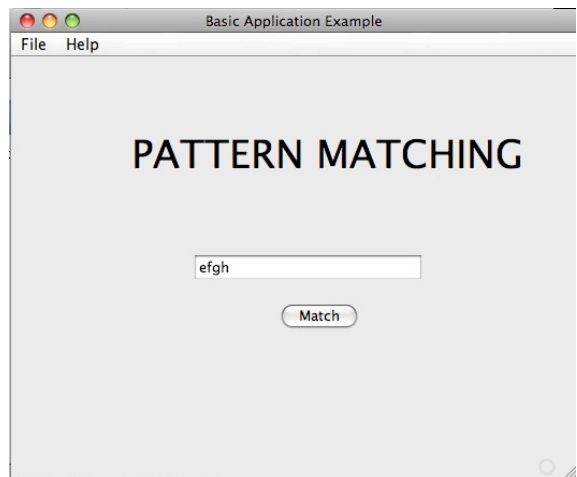
Aplikasi sederhana dari metode ini adalah sebagai berikut :



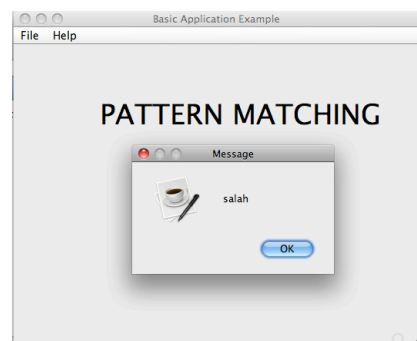
Gambar 1. Pengguna (User) Memasukkan input dengan benar



Gambar 2. Pesan ketika Pengguna (User) Memasukkan input dengan benar



Gambar 3. Pengguna (User) Memasukkan input dengan salah



Gambar 4. Pesan ketika Pengguna (User) Memasukkan input dengan salah

Cara kerja dari program ini adalah, program akan mengambil indeks teks dari 0 sampai 3 yaitu abcd. Selanjutnya indeks itu akan dibandingkan satu-persatu dengan input dari *user*. Jika user memasukkan dengan benar maka akan timbul pesan benar. Jika tidak akan muncul pesan salah.

III. KESIMPULAN

- Algoritma *Brute Force* pada *pattern matching* bukan merupakan algoritma yang mangkus (optimal). Sebab jika teks yang digunakan sangat panjang maka diperlukan waktu yang sangat lama untuk mencarinya.
- Aplikasi *Pattern Matching* pada bidang komputer sangat banyak salah satunya *Search Engine* dari Google.

IV. REFERENSI

- [1] Rinaldi Munir, Strategi Algoritma, 2006, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- [2] MIT Lecture Notes 2009
- [3] <http://www.informatika.org/~rinaldi>