

VERSION 1.0
SEPTEMBER 26, 2021



ALGORITMA PEMOGRAMAN

ALGORITMA BRUTE FORCE | MODUL 6

DISUSUN OLEH:

- FARKHAN HAMZAH FIRDAUS
- GILLY HUGA ANARGYA

DIAUDIT OLEH:

- DIDIH RIZKI CHANDRANEGARA, S.KOM., M.KOM
- CHRISTIAN SRI KUSUMA, S.KOM, M.KOM

PRESENTED BY: TIM LAB-IT

UNIVERSITAS MUHAMMADIYAH MALANG

ALGORITMA PEMOGRAMAN

PERSIAPAN MATERI

String Matching

Pada pengembangan aplikasi yang berisi data-data, string adalah sesuatu yang sangat krusial. Terlebih lagi jika aplikasi tersebut berisi database atau aplikasi tersebut merupakan aplikasi pemrosesan teks. Hampir semua bahasa pemrograman memiliki library khusus yang melakukan pemrosesan string. Salah satunya adalah java, java memiliki library yang berupa class bernama String untuk melakukan pemrosesan String. Salah satu method yang disediakan adalah method matches().

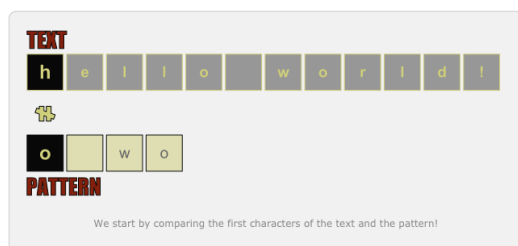
Method tersebut digunakan untuk mencocokkan antara kedua string apakah memiliki kesamaan. Untuk mencari kesamaan antara dua string (biasanya berupa text dan pattern yang ingin dicocokkan), pada dasarnya kita hanya membandingkan masing-masing karakter dari text apakah cocok dengan pattern yang dibandingkan, lalu kemudian me return apakah pattern ini muncul didalam text. Selain menggunakan method matches() dari Class String, kita juga bisa menggunakan pendekatan brute-force.

Algoritma Brute-Force

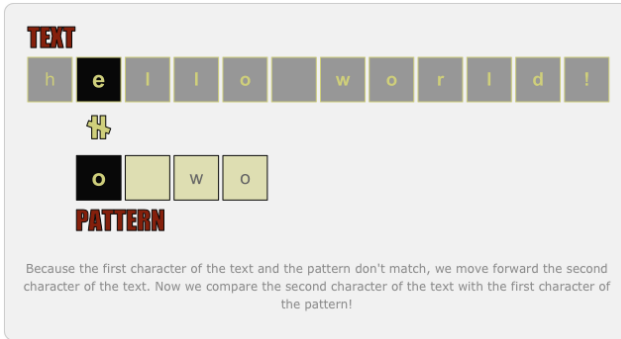
Merupakan suatu algoritma yang memiliki pendekatan yang lempang untuk memecahkan suatu persoalan. Algoritma brute force memecahkan persoalan dengan sangat sederhana, langsung, dan jelas. Algoritma ini cukup banyak digunakan dan sebenarnya berkinerja cukup baik dalam banyak kasus.

Penerapan Algoritma Brute Force pada String Matching

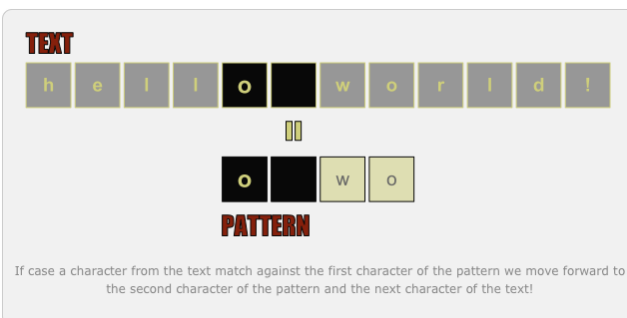
Prinsip dari brute force pada string matching sebenarnya cukup simpel. Kita harus mengecek kecocokan antara karakter pertama di pattern dan karakter di text.



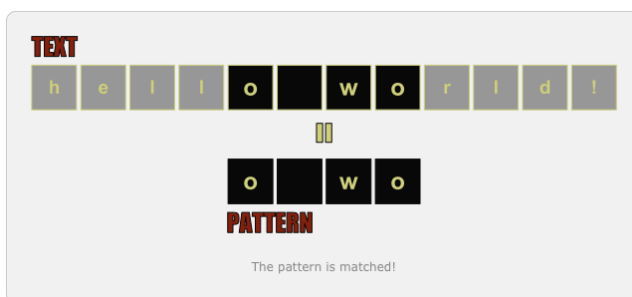
Langkah pertama adalah membandingkan karakter pertama pada pattern dengan karakter pertama pada text. Jika tidak cocok, maka akan berpindah ke karakter kedua pada text.



Kemudian program akan di looping hingga karakter pada text cocok pada karakter pertama pada pattern. Kemudian setelah karakter pada text cocok dengan karakter pertama pada pattern, maka program akan berlanjut membandingkan karakter setelah karakter yang cocok pada text dengan karakter setelah karakter pertama pada pattern.



Pencocokkan akan terus dilakukan hingga karakter terakhir pada pattern cocok dengan karakter yang ada di text.



Pencocokkan akan mereturn posisi awal dari karakter dan posisi akhir dari karakter pada text yang cocok dengan pattern.

Dibawah ini merupakan referensi yang bisa kalian gunakan untuk Referensi :

<https://dzone.com/articles/algorithm-week-brute-force>

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-\(2016\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf)

Implementasi :

```

for (int i = 0; i < n-m; i++) {
    int j = 0;
    while (j < m && t[i+j] == p[j]){
        j++;
    }
    if (j == m) return i;
}
System.out.println("No match found"); return -1;

```

TUJUAN

1. Mahasiswa mampu menjelaskan dan menerapkan algoritma brute force untuk menyelesaikan masalah yang dihadapi.
2. Mahasiswa mampu menganalisa kompleksitas algoritma brute force dalam menyelesaikan masalah.

TARGET MODUL

1. Mahasiswa mampu membaca, mencari dan menguraikan algoritma brute force yang sesuai untuk diterapkan pada program yang dibuat.
2. Mahasiswa mampu berpikir kreatif dalam menciptakan suatu.

PERSIAPAN SOFTWARE/APLIKASI

Eclipse/ Netbeans/IntelliJ IDEA/ dsb (IDE Bahasa Java)

TUGAS

1. Carilah implementasi lain dari algoritma brute force selain string matching dan buat program dengan menggunakan bahasa java, kemudian gambarkan flowchartnya.
2. Buatlah pseudocode dari program yang kalian buat.
3. Analisis kompleksitas waktu dari program diatas.
4. Jelaskan kepada asisten alur kerja dari program yang kalian buat.
5. **Jika ada program yang sama/identik dengan praktikan lain, nilai maksimal adalah C+.**

DETAIL PENILAIAN TUGAS

1. Menyelesaikan tugas praktikum sesuai dengan ketentuan. (40 poin)
2. Membuat pseudocode sesuai dengan program yang dibuat. (15 poin)
3. Dapat menjelaskan alur dan kompleksitas waktu dari program yang dibuat (45 poin)