Algoritma String Matching pada Mesin Pencarian

Harry Octavianus Purba – 13514050
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Bandung, Indonesia
13514050@stei.itb.ac.id

Proses pencocokan string adalah proses menemukan ataupun mencari suatu pattern di dalam teks. Algoritma pencocokan string yang akan dibahas di makalah ini adalah algortma brute force, algoritma KMP (Knutt-Morris-Pratt) dan algoritma Boyer-Moore. Banyak penerapan yang dapat dilakukan dengan memanfaatkan algoritma pencocokan string. Salah satu contohnya yang akan di bahas di sini adalah pencocokan string untuk mesin pencaria Dalam makalah ini akan dibahas bagaimana proses pencocokan string dalam mesin pencarian yang digunakan untuk mencari informasi.

Keywords—string, pattern, prefix, suffix.

I. PENDAHULUAN

Dalam kehidupan modern saat ini sudah sangat mudah dalam mendapatkan informasi. Hal itu disebabkan karena semakin tingginya kemajuan teknologi seperti internet, smartphone, dan gadget lainnya. Dalam mendapatkan informasi , pengguna biasanya menggunakan mesin pencarian seperti Google, Yahoo, ataupun mesin pencarian yang lainnya. Tetapi bagaimana sebenarnya mesin pencarian memberikan keluaran sesuai dengan keyword yang dimasukkan oleh pengguna ? Caranya adalah dengan menggunakan pencocokan string (String matching).

Pencocokan string adalah teknik mencari suatu pattern dalam suatu teks. Dalam makalah ini akan dijelaskan mengenai bagaimana pencocokan string digunakan dalam mesin pencarian. Baik itu brute force , KMP , maupun Boyer-Moore.

II. DASAR TEORI

1. Mesin Pencarian

Mesin pencari web adalah suatu program komputer yang dirancang untuk melakukan pencarian file-file yang tersimpan dalam layanan seperti www, ftp, publikasi milis, ataupun news group dalam sebuah/sejumlah komputer dalam suatu jaringan.. Hasil pencarian pada umumnya ditampilkan dalam bentuk list yang diurutkan menurut tingkat banyak pengunjung ke file tersebut. Informasi yang didapatkan dapat berupa halaman situs web, gambar, ataupun file lainnya. Mesin pencari juga melakukan pengumpulan informasi atas data yang tersimpan dalam suatu basisdata ataupun direktori web.

Sebagian besar mesin pencari dijalankan oleh perusahaan swasta yang menggunakan algoritma kepemilikan danbasisdata tertutup, di antaranya yang paling populer adalah Google (MSN Search dan Yahoo!).

Cara Kerja Mesin Pencarian

Untuk mengumpulkan informasi dari sebuah situs blog atau website, mesin pencari akan melakukan 3 buah proses yakni :

1. Proses Crawling

Pada proses crawling digunakan istilah spider. Spider bertugas mengumpulkan informasi mengenai situs blog atau website. Spider mengumpulkan informasi dari mulai link, struktur HTML, meta tag, judul, hingga konten teks. Spider dapat merayapi situs blog atau website yang menggunakan/memiliki file robots.txt.Robots.txt berisi script yang kemudian akan diterjemahkan oleh spider sebagai perintah untuk mengumpulkan informasi-informasi yang disebutkan. Dan robots.txt juga akan memudahkan spider untuk mengumpulkan data. Proses crawling merupakan proses yang sangat penting. Jika proses crawling tidak berjalan dengan lancar, maka mesin pencari tidak akan mengenali situs blog atau website tersebut.

2. Proses Indexing

Setelah proses crawling sudah dilakukan , maka informasi yang didapatkan akan disimpan dalam database. Penyimpanan pada database ini menggunakan index yang juga mencantumkan alamat URLnya.Penyimpanan ini dilakukan secara berkala, untuk mempercepat proses pencarian.

3. Proses Searching

Yang terakhir adalah proses searching. Proses searching dilakukan berdasarkan perintah dari pengguna mesin pencarian. Pada saat pengguna melakukan pencarian menggunakan kata kunci (keyword) yang dikehendaki, maka

mesin pencari (search engine) akan menampilkan informasi berdasarkan hasil proses indexing. Mesin pencari (search engine) akan menampilkan judul, cuplikan artikel yang sesuai dengan kata kunci (keyword), beserta cuplikan URL tersebut.

Contoh-contoh Mesin Pencarian

Contoh-contoh mesin pencarian antara lain:

- Google
- Yahoo
- Baidu
- Bing
- Yandex
- Ask
- AOL

2. Pencocokan String

Pencocokan string ataupun pencarian string adalah proses melakukan pencarian semua kemunculan string pendek (panjang lebih besar dari satu dan lebih kecil dari panjang teks) yang disebut dengan pattern terhadap teks (dengan panjang 1 sampai dengan n).

Untuk melakukan proses pencocokan/pencarian string dapat dilakukan dengan algoritma-algoritma seperti :

- Brute Force
- KMP
- Boyer Moore

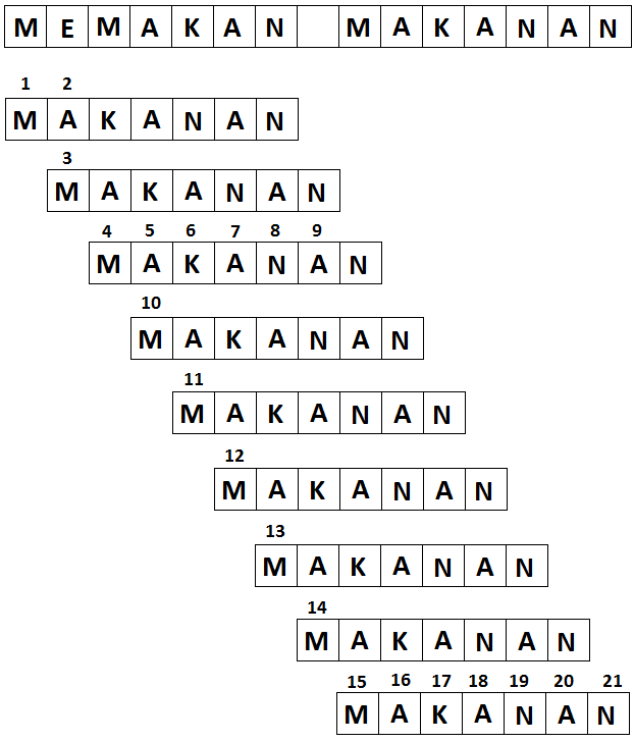
Selain ketiga algoritma di atas , ada juga algoritma-algoritma lainnya namun tidak akan dibahas.

- **Algoritma Brute Force**

Algortima brute force adalah algoritma yang ditulis dengan pemikiran secara langsung tanpa memikirkan peningkatan performa seperti kompleksitas algoritma. Pseudocode dalam pencocokan string dengan menggunakan bruteforce adalah :

```
procedure BruteForceSearch(
      input m, n : integer,
      input P : array[0..n-1] of char,
      input T : array[0..m-1] of char,
      output ketemu : array[0..m-1] of boolean
      )

Deklarasi:
     i, j: integer
Algoritma:
      for (i:=0 to  m-n) do
           j:=0
           while (j < n and T[i+j] = P[j]) do
                 j:=j+1
           endwhile
           if(j >= n) then
                 ketemu[i]:=true;
           endif
      endfor
```

Contoh tahapan penyelesaian dengan menggunakan brute force adalah :



Dalam penyelesaian pencarian string dengan menggunakan brute force memiliki best case, worst case dan juga average case.

Best case

- Kompleksitas kasus terbaik adalah $O(n)$.
- Terjadi bila karakter pertama *pattern P* tidak pernah sama dengan karakter teks *T* yang dicocokkan
- Jumlah perbandingan maksimal *n* kali:
- Contoh:
  T: String ini berakhir dengan zzz
  P: zzz

Worst Case

- Jumlah perbandingan: $m(n - m + 1) = O(mn)$
- Contoh:
  T: "aaaaaaaaaaaaaaaaaaaaaaaaah"
  P: "aaah"

Average Case

- Menggunakan kompleksitas O (m+n)
- Contoh :
  T: "a string searching example is standard"
  P: "store"

- **Algoritma KMP (Knuth-Morris-Pratt)**

Pada algoritma *brute force*, setiap kali ditemukan ketidakcocokan *pattern* dengan teks, maka *pattern* digeser satu karakter ke kanan. Sedangkan pada algoritma KMP, akan dipelihara informasi yang digunakan untuk melakukan jumlah pergeseran. Algoritma menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh, tidak hanya satu karakter seperti pada algoritma *brute force*. Di KMP dikenal istilah fungsi pinggiran (border).

- Fungsi *pinggiran* $b(j)$ didefinisikan sebagai ukuran awalan terpanjang dari P yang merupakan akhiran dari $P[1..j]$.
- Sebagai contoh, tinjau *pattern* P = `abaaba`. Nilai F untuk setiap karakter di dalam P adalah sebagai berikut:

`

j	1	2	3	4	5	6
$P[j]$	a	b	a	a	b	a
$b(j)$	0	0	1	1	2	3

Pseudocode untuk KMP

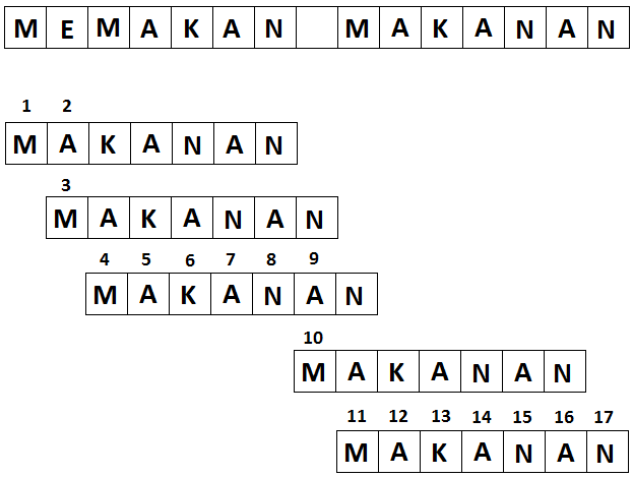
```
procedure preKMP(
    input P : array[0..n-1] of char,
    input n : integer,
    input/output kmpNext : array[0..n] of integer
)
Deklarasi:
i,j: integer

Algoritma
  i := 0;
  j := kmpNext[0] := -1;
  while (i < n) {
     while (j > -1 and not(P[i] = P[j]))
        j := kmpNext[j];
     i:= i+1;
     j:= j+1;
     if (P[i] = P[j])
        kmpNext[i] := kmpNext[j];
     else
        kmpNext[i] := j;
     endif
  endwhile
```

```
procedure KMPSearch(
    input m, n : integer,
    input P : array[0..n-1] of char,
    input T : array[0..m-1] of char,
    output ketemu : array[0..m-1] of boolean
)

Deklarasi:
i, j,next: integer
kmpNext : array[0..n] of interger

Algoritma:
preKMP(n, P, kmpNext)
i:=0
while (i<= m-n) do
    j:=0
    while (j < n and T[i+j] = P[j]) do
       j:=j+1
    endwhile
    if(j >= n) then
       ketemu[i]:=true;
    endif
    next:= j - kmpNext[j]
    i:= i+next
endwhile
```

Kompleksitas untuk KMP :

- Menghitung fungsi pinggiran : $O(m)$,
- Pencarian *string* : $O(n)$
- Kompleksitas waktu algoritma KMP adalah $O(m+n)$.

Tahapan pencocokan string pada KMP



- **Algoritma Boyer-Moore**

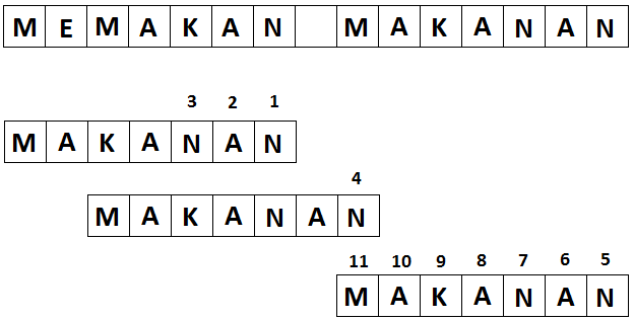
Algoritma Boyer Moore adalah algoritma yang dianggap sebagai algoritma yang paling efisien pada aplikasi umum.Tidak seperti algoritma pencarian string yang ditemukan sebelumnya, algoritma Boyer-Moore mulai mencocokkan karakter dari sebelah kanan pattern. Ide di balik algoritma ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat. Secara sistematis, langkah-langkah yang dilakukan algoritma Boyer-Moore pada saat mencocokkan string adalah:

- Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks.
- Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
    1. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
    2. Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
- Algoritma kemudian menggeser pattern dengan memaksimalkan nilai penggeseran good-suffix dan penggeseran bad-character, lalu mengulangi langkah dua sampai pattern berada di ujung teks.

Pseudocode dalam algoritma Boyer-Moore:

```
int BMSearch(
        input m,n : integer,
        input T : array[0..n-1] of char,
        input T : array[0..n-1] of char
)

Deklarasi:
I,j:integer;

Algoritma:

i <-- m - 1
j <-- m - 1
repeat
        if P[j] = T[i] then
                if j = 0 then
                        return i {a match!}
                else {check next character}
                        i <-- i - 1
                        j <-- j - 1
        else { P[j] <> T[i] move the pattern}
                i <-- i + m - j - 1
                i <-- i + max(j - last(T[i]), match(j))
                j <-- m-1
until i > n - 1
return 0
```

Tahapan pencocokan string untuk Boyer-Moore



III. Pencarian Informasi dengan String Matching

Dalam proses pencarian di mesin pencarian , terlebih dahulu pengguna memasukkan keyword ke dalam kotak mesin pencarian . Setelah input keyword tersebut maka disinilah dipergunakan string matching untuk mencocokkan terhadap data-data yang disimpan didatabase.

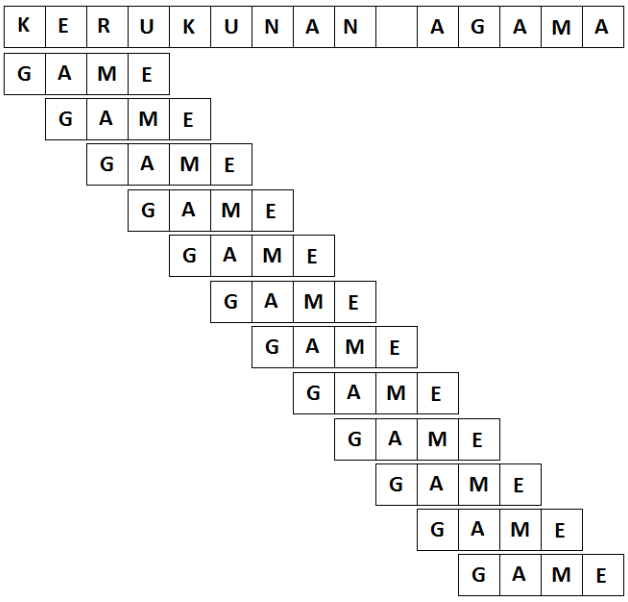
Misalkan pada kotak isian mesin pencarian dimasukkan keyword " GAME" . Sedangkan dalam database berisi informasi :

- … KERUKUNAN AGAMA
- … MENGGAMBAR MANGA
- … DOWNLOAD GAMES GRATIS
- Dst

Setelah diinput maka akan dicocokkan pada indeks-indeks yang ada pada database .

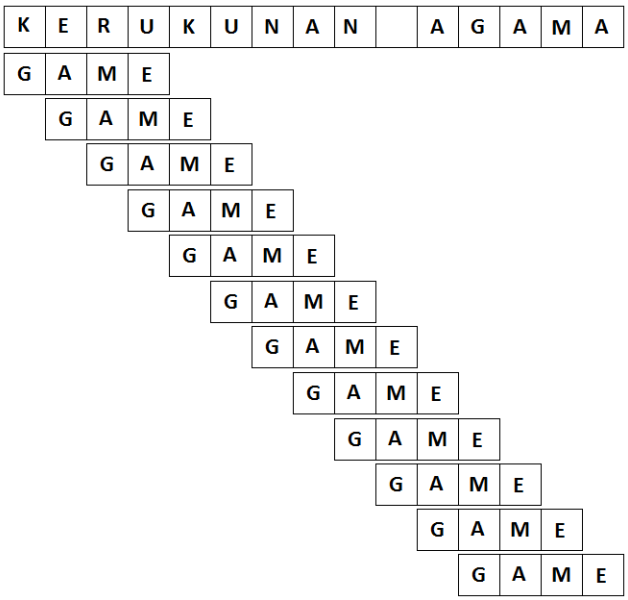
Untuk data pertama

Dengan menggunakan brute force :



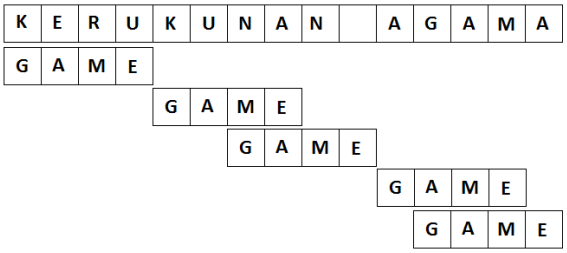
Banyak perbandingan yang dilakukan adalah 12 .

Dengan menggunakan KMP



Perbandingan sama dengan brute force karena tidak ada prefix yang sama dengan prefix pada “GAME”. Sehingga banyak perbandingan adalah 12 .

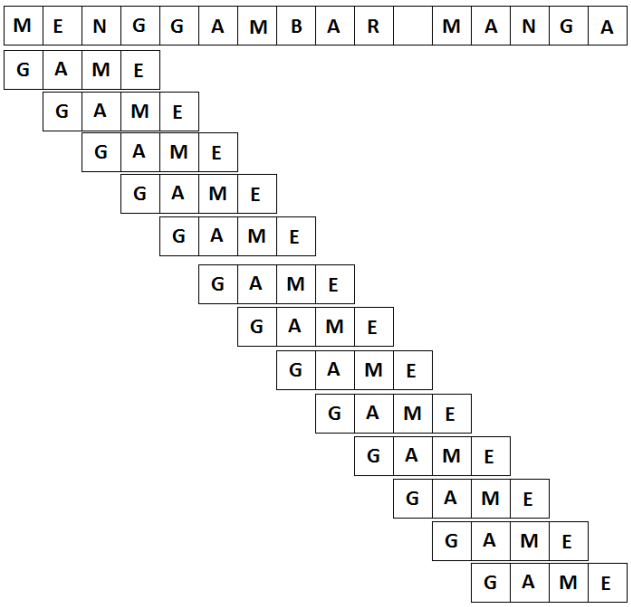
Dengan menggunakan Boyer-Moore



Perbandingan yang dilakukan adalah 1+1+1+1+1 = 5

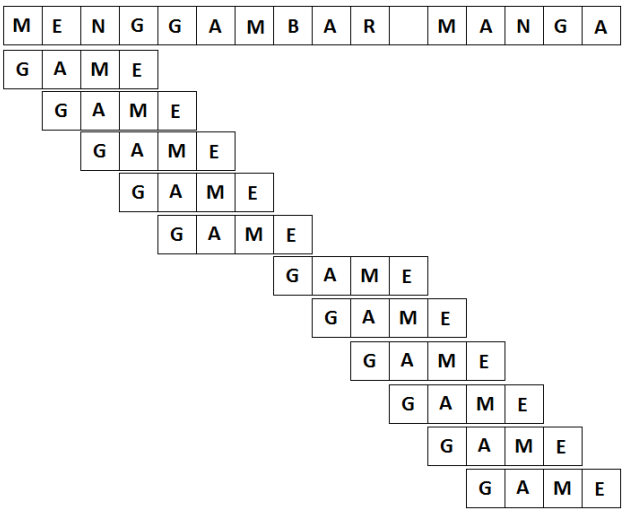
Ternyata setelah melakukan pencocokan string tidak cocok dengan keyword , maka dilakukan pencarian untuk informasi pada indeks berikutnya.

Dengan menggunakan brute force



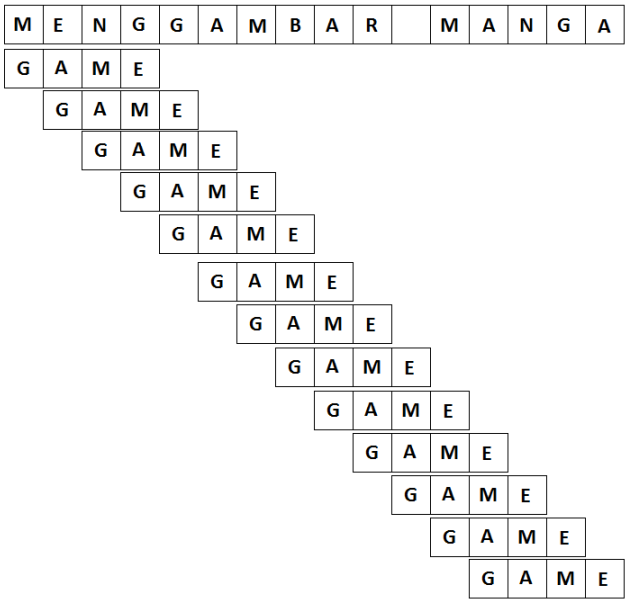
Banyak perbandingan yang dilakukan adalah 16 karena untuk step 4 dan 5 melakukan penghitungan sebanyak dua kali .

Dengan menggunakan KMP



Banyak perbandingan yang dilakukan adalah 14 . Untuk step 4 melakukan perbadingan dua kali , untuk step 5 melakukan perbadingan tiga kali.

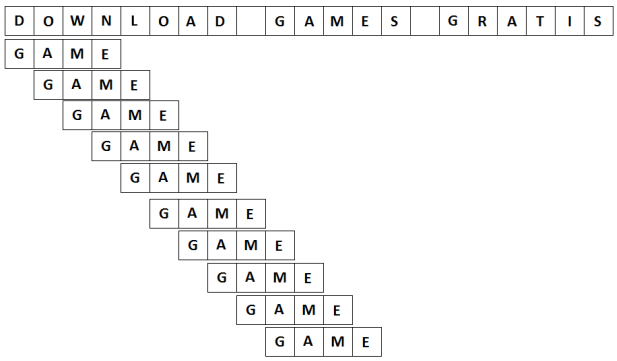
Dengan menggunakan algoritma Boyer-Moore



Banyak perbandingan adalah panjang teks-panjang pattern+1 yaitu 16-4+1 = 13.

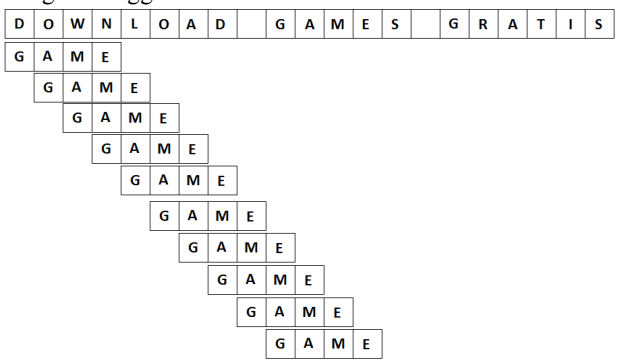
Ternyata kasus yang sama juga berlaku untuk informasi ini . Karena masih belum ada string yang cocok , maka dilanjutkan pencarian untuk informasi selanjutnya

Dengan menggunakan brute force :



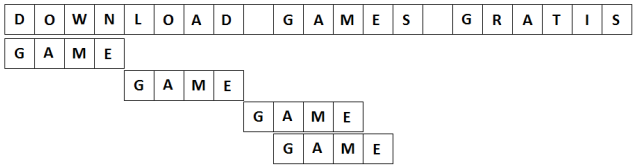
Banyak perbandingan adalah 13.Untuk step terakhir melakukan perbandingan empat kali (mencocokkan GAME)

Dengan menggunakan KMP :



Banyak perbandingan yang dilakukan adalah 13 . terakhir melakukan perbadingan empat kali.

Dengan menggunakan Boyer-Moore :



Banyak perbandingan yang dilakukan adalah 7 kali . Untuk step 4 melakukan perbadingan empat kali.

Ternyata untuk pencarian ketiga telah ditemukan string yang cocok. Artinya URL yang memiliki bagian string " …DOWNLOAD GAMES GRATIS " ini akan ditampilkan ke list hasil pencarian. Selanjutnya dilanjutkan juga pencarian yang sama untuk data-data berikutnya pada database. Dan jika cocok akan ditampilkan sebagai hasil pencarian yang kedua , ketiga , dan seterusnya.

IV Analisis

Ternyata setelah dilakukan ketiga pencarian string yaitu brute force , KMP , dan juga Boyer-Moore . Untuk brute force melakukan perpindahan pattern selalu bertambah satu. Sementara untuk KMP perpindahan pattern bergantung padaborder . Untuk Boyer-Moore bergantung hubungan suffix dengan bagian di dalam pattern.

Banyaknya perbandingan karakter yang terjadi pada KMP dapat sama dengan banyak perbandingan karakter yang terjadi pada brute force. Hal ini terjadi ketika banyak prefix yang sama antara pattern terhadap teks untuk tiap pergeseran adalah nol atapun satu. Contoh :

- Teks : MAKAN MAKANAN BERGIZI
  Pattern : MINUM

  Saat berjumpa dengan "M" pada teks, tidak pernah didampingi oleh "I".
- Teks : MINUMAN YANG BERGIZI
  Pattern: KUMAN
  Tidak pernah berjumpa dengan "K".

V Simpulan

Untuk melakukan pencarian pada mesin pencarian dapat digunakan algoritma-algoritma seperti : bruteforce, KMP , Boyer-Moore ataupun algoritma lainnya. Setelah melakukan uji coba terhadap tiga kasus dengan algoritma brute force , KMP , dan Boyer-Moore didapatkan kesimpulan algoritma Boyer-Moore lebih mangkus daripada algoritma KMP. Dan juga algoritma KMP lebih mangkus daripada algoritma brute force. Dan ada juga kasus banyak

perbandingan karakter pada KMP sama dengan brute force. Hal ini terjadi ketika banyak prefix yang sama untuk setiap pergeseran pattern adalah nol ataupun satu.

VI Ucapan Terima Kasih

Penulis ingin mengucapkan terimakasih kepada Tuhan Yang Maha Kuasa karena berkat-Nya lah penulis dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada dosen yang membina kami dalam kuliah IF2210 Strategi Algoritma yaitu Bapak Rinaldi Munir dan Ibu Nur Ulva Maulidevi . Penulis juga berterima kasih kepada teman-teman yang mendukung Penulis dalam menyelesaikan makalah ini.

Referensi

https://id.wikipedia.org/wiki/Mesin_pencari_web
https://id.wikipedia.org/wiki/Algoritma_pencarian_string
http://dyardian.heck.in/ini-adalah-cara-kerja-mesin-pencari.xhtml
Munir,Rinaldi. *Pencocokan String*. 2016
http://www8.cs.umu.se/kurser/TDBA59/VT01/mom3/slides/BM-alg.html

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, 6 Mei 2016

Harry Octavianus Purba
13513036