Penggunaan Algoritma Commentz-Walter dalam String Matching

Vicko Novianto | 13513092[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13513092@std.stei.itb.ac.id

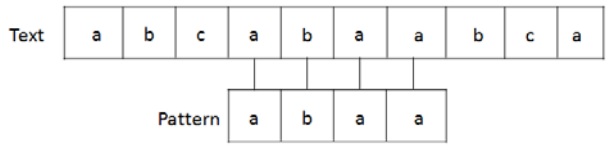
***Abstract*—Algoritma pencocokan string adalah algoritma untuk menemukan pola di dalam teks. Banyak algoritma pencocokan string yang terkenal, salah satunya adalah algoritma Commenz-Walter. Algoritma ini menggabungkan ide Boyer-Moore dan Aho-Corasick. Algoritma ini memiliki kompleksitas linear pada umumnya. Algoritma ini menggunakan struktur trie untuk menyimpan kata-kata kunci.**

***Index Terms*—Commentz-Walter, Pola, Shift, Teks, Trie**

I. PENDAHULUAN

Algoritma pencocokan string adalah algoritma untuk menemukan satu atau beberapa string/pola di dalam string yang lebih besar atau teks. [1] Pencocokan string mempunyai peran yang besar dalam menyelesaikan persoalan di berbagai bidang, misalnya deteksi intrusi dalam jaringan, aplikasi bioinformatik, deteksi plagiarisme, kemananan informasi, pengenalan pola, pencocokan dokumen dan penambangan data(*text mining*). Algoritma pencocokan string bisa dibagi dua, yaitu algoritma pencocokan string eksak (*exact string matching algorithm*) dan algoritma pencocokan string non-eksak (*approximate string matching algorithm*). Algoritma pencocokan string ada yang hanya bisa mencocokkan satu pola dan ada yang bisa mencocokkan banyak pola.

Misalkan Σ adalah set alfabet yang terbatas. Secara formal, baik pola maupun teks adalah vektor dari elemen Σ. Asumsikan teks adalah array T[1..n] dengan panjang n dan pola adalah array P[1..m] dengan panjang m dan m <= n. Pola P ditemukan pada pergeseran (*shift*) s di teks T (dengan kata lain P ditemukan mulai pada posisi s+1 di teks T) jika 0 <= s <= n-m dan T[s+1..s+m] = P[1..m]. Jika P ditemukan pada s yang berada pada range 0 <= s <= n-m maka dikatakan valid. Algoritma pencocokan string adalah masalah menemukan semua s yang valid dengan pola P di dalam teks T. [1]



Gambar 1. Ilustrasi pencocokan string

II. DASAR TEORI

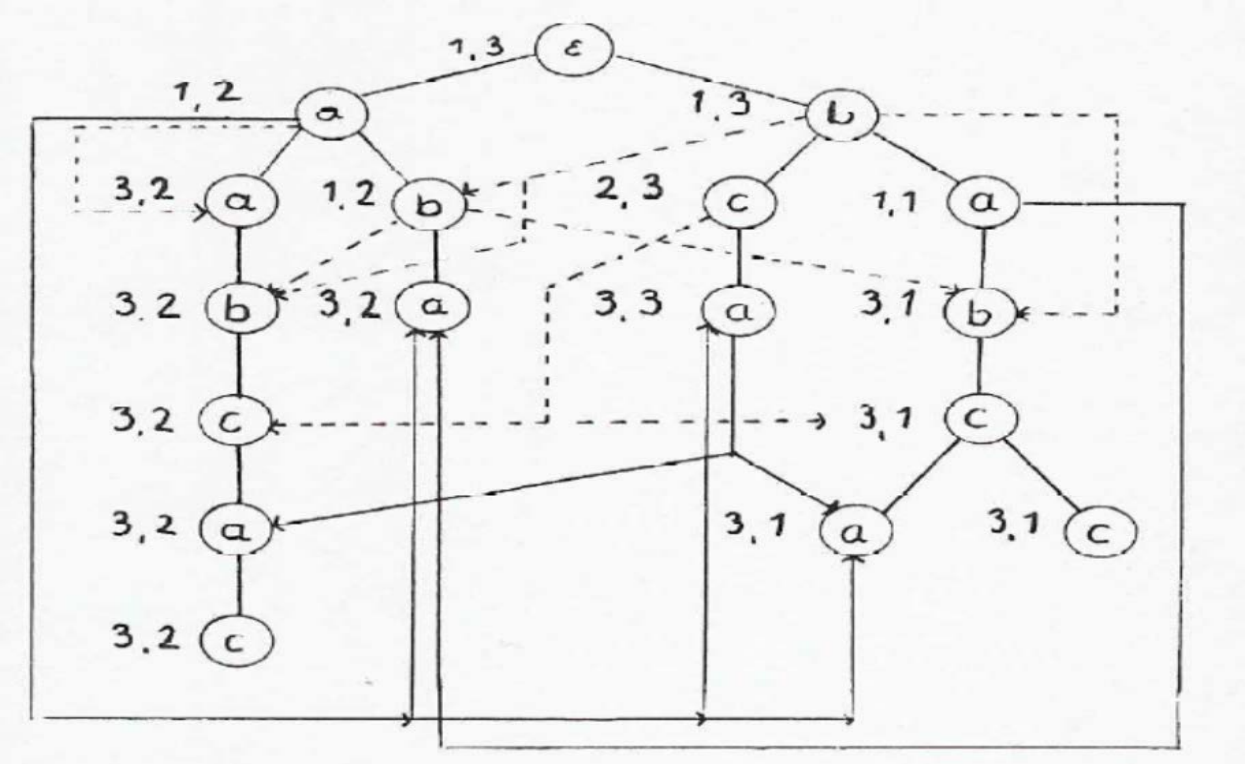
Algoritma pencocokan string yang menggunakan banyak pola menggunakan teks T = $t_1t_2…t_n$ dimana t_i adalah karakter ke-i dan kita ingin mencari bersamaan untuk sebuah himpunan string P = {p1, p2, …, pr} dengan pi adalah string dengan panjang mi dengan i=1..r[2]. Ada banyak algoritma pencocokan string yang menggunakan banyak pola yang bervariasi dalam kecepatannya dihitung dari kompleksitasnya. Beberapa contohnya :

1. Algoritma pencocokan string Aho-Corasick
   Kompleksitas dari algoritma ini adalah linear terhadap panjang pola ditambah panjang dari teks yang dicari ditambah jumlah kecocokan. Algoritma ini membangun sebuah finite state machine yang merupai trie dengan sisi tambahan antara simpul internalnya. Sisi tambahan ini memungkinkan transisi yang cepat bila terjadi ketidakcocokkan(misalkan saat mencari cat dalam trie yang tidak mengandung cat namun mengandung cart, maka akan gagal pada simpul ca) ke cabang trie yang lain yang memiliki prefiks yang sama. Ini memungkinkan kita tidak perlu melakukan runut-balik.
2. Algoritma pencocokan string Rabin Karp
   Algoritma ini menggunakan hash untuk menemukan himpunan pola dalam sebuah teks. Kompleksitas terburuknya adalah O(nm).
3. Algoritma pencocokan string Commentz-Walter

III. ALGORITMA COMMENTZ-WALTER

A. Struktur data

Untuk menggambarkan beberapa kata kunci dalam cara yang baik, struktur data yang digunakan adalah *trie*.



Gambar 2. Trie

Sebuah *trie* adalah pohon T sehingga :

1. Setiap simpul v pada T, kecuali akar r diberi label beberapa karakter a = l(v), sebuah elemen dari beberapa alfabet A.
2. Akar r diberi label ε yang melambangkan kata yang kosong.
3. Jika simpul v' dan v'' adalah saudara(memiliki orang tua yang sama yaitu v), v' ≠ v'' maka l(v') ≠ l(v'').

Sebuah jalur v1,…,vm pada T dengan vi+1 adalah anak dari vi melambangkan kata l(v1) l(v2) … l(vm). Kata ini kita lambangkan dengan w(vm) jika dan hanya jika v1 = r, dengan r adalah akar.

Untuk setiap simpul, kedalamannya dilambangkan dengan :

d(v) = 0 jika v = r, dengan r adalah akar dan d(v) = d(v') + 1 jika v adalah anak dari v'. Kedalaman trie T dilambangkan dengan d(T) = max(d(v); v adalah elemen T).

Misalkan K= W1,…,Wr adalah himpunan kata kunci dengan beberapa alfabet A yang ingin kita cari dalam beberapa dokumen D. Sama dengan algoritma pencocokan string Aho-Corasic, kita menggmbarkan K dengan trie T. Namun, kita basiskan trie T dengan kata kunci yang dibalik (*reversed keywords*) :

Untuk h = 1..r ada sebuah simpul vh pada T menggambarkan reserved keyword W_h^R. Misalkan w(vh) = $w_{h,1}$, … , $w_{h,|Wh|}$ dimana Wh = $w_{h,|Wh|}$, …, $w_{h,1}$.

Untuk setiap simpul kita tambahkan fungsi keluaran out(v) = { W; W^R = w(v), W in K }.

Untuk trie ini kita tambahkan fungsi shift1 dan shift2 yang akan memetakan setiap simpul trie ke sebuah bilangan bulat. Tujuan fungsi shift1 dan shift2 akan jelas di bagian fase pencarian algoritma Commentz-Walter. Definsi shift1 dan shift2 adalah berdasarkan himpunan-himpunan dari simpul :

Untuk setiap v ≠ r pada T :

set1(v) = {v'; w(v) adalah sufiks yang layak dari w(v'), misalnya w(v') = u w(v) untuk beberapa kata tidak kosong u }

set2(v) = { v'; v' adalah elemen dari set1(v) dan out(v') ≠ ø }

Definisi shift1 dan shif2 :

shift1(v) = 1 jika v = r atau

min({k; k = d(v') – d(v), v' adalah elemen dari set1(v)} u {w_{min}})

shift2(v) = w_{min} atau

min({k; k = d(v') – d(v), v' adalah elemen dari set2(v)} u shift2(ayah dari v'))

Misalkan wmax = max { |Wh|; h = 1..r } dan wmin = min { |Wh|; h = 1..r }.

Terakhir kita tambahkan fungsi :

Char : A → N dengan

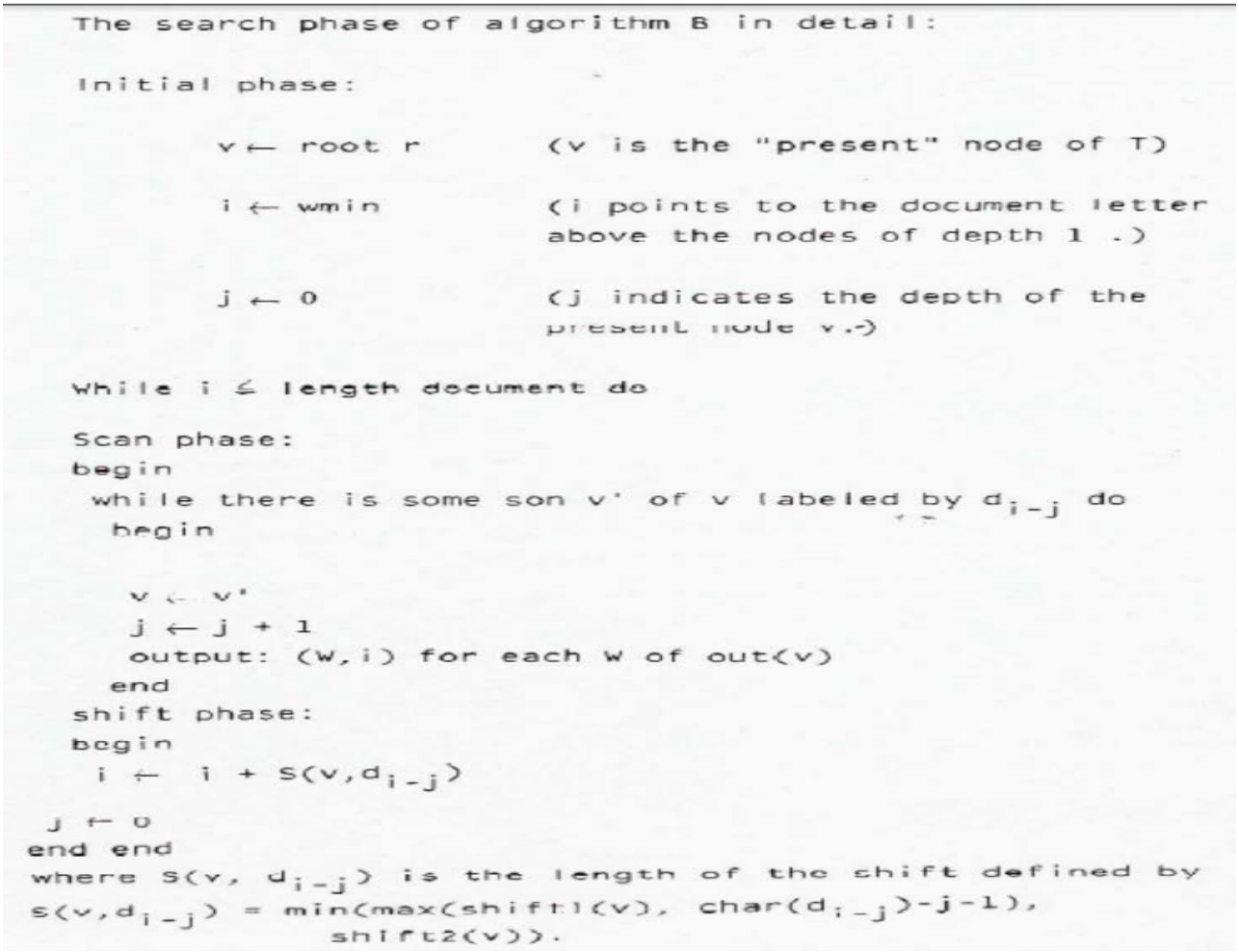
Char(a) = min({d(v); l(v) – a} {wmin + 1})

Misalkan k = { cacbaa, acb, aba, acbab, ccbab {, wmin = 3

Dua buah bilangan bulat di samping setiap simpul menggambarkan hasil dari shift1(v) dan shift2(v).

B.Fase pencarian

Masukan dari algoritma pencocokan string Commentz-Walker adalah beberapa dokumen D dan untuk beberapa himpunan kunci K, trie T yang sudah diproses sebelumnya dan fungsi out, shift1, shift2, dan char.

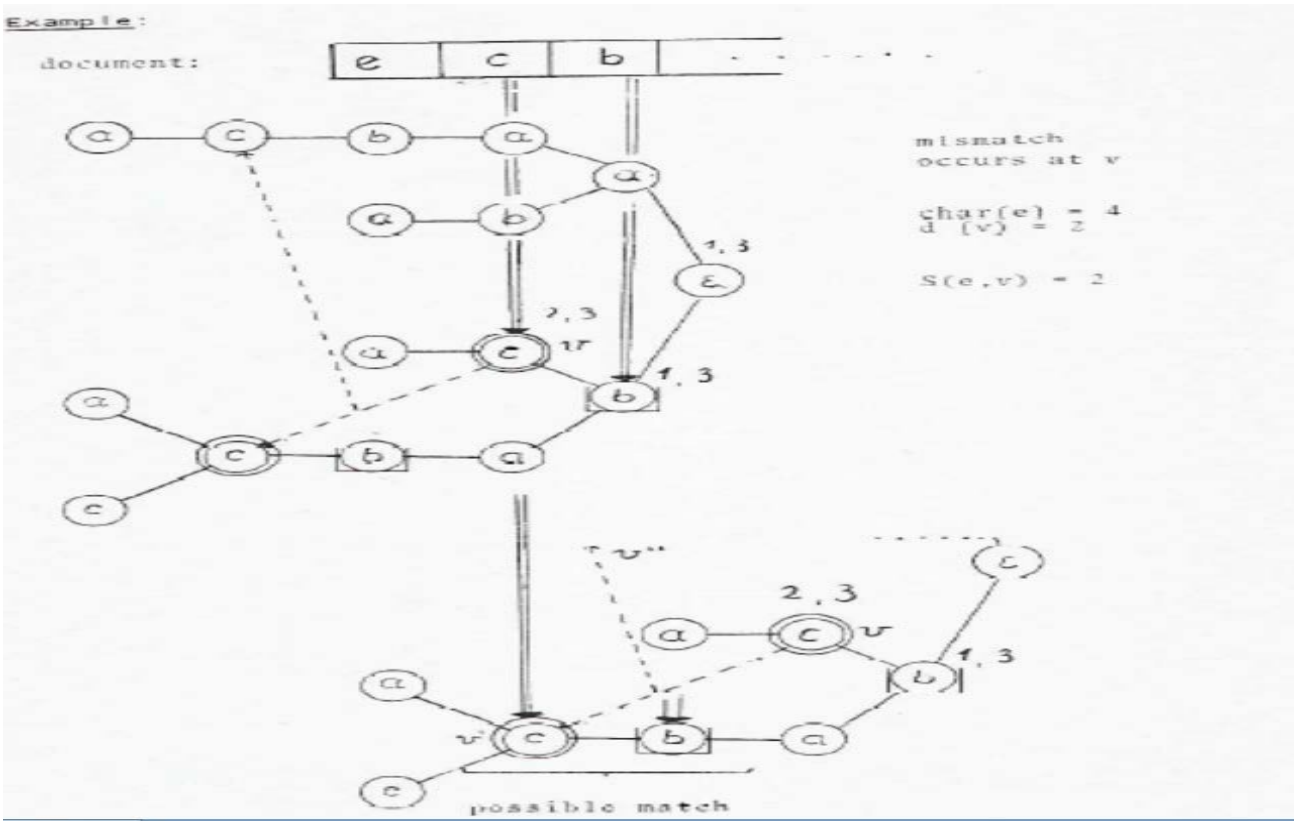


```
The search phase of algorithm B in detail:

Initial phase:

        v ← root r          (v is the "present" node of T)

        i ← wmin            (i points to the document letter
                             above the nodes of depth 1 .)

        j ← 0               (j indicates the depth of the
                             present node v.)

While i ≤ length document do

Scan phase:
begin
 while there is some son v' of v labeled by d_{i-j} do
  begin

   v ← v'
   j ← j + 1
   output: (W,i) for each W of out(v)
  end
shift phase:
begin
 i ← i + S(v,d_{i-j})

 j ← 0
end end
where S(v, d_{i-j}) is the length of the shift defined by
s(v,d_{i-j}) = min(max(shift1(v), char(d_{i-j})-j-1),
                   shift2(v)).
```

Gambar 3. Algoritma fase pencarian

Keluaran dari fase pencarian algoritma pencocokan string Commentz-Walker adalah list dari pasangan(w,i) dengan w adalah sebuah kata dan I adalah sebuah bilangan bulat yang menggambarkan kemunculan W, misalnya (W,i) elemen dari keluaran algoritma ini dan W adalah kata kunci dari K dan $d_{i\,|W|+1}$, …, d_i = W.

Algoritma ini menggabungkan ide dari algoritma pencocokan string Aho-Corasick dan Boyer-Moore. Kita mendasarkan trie dengan kata kunci yang dibalik. Misalkan wmin melambangkan panjang minimal dari beberapa kata kunci. Algoritma ini mulai menaruh akar r pada T dibawah d_{wmin+1}. Selanjutnya kita kan memindai dokumen dari kanan ke kiri sampai terjadi ketidakcocokkan(*mismatch*). Asumsikan kita baru saja memindai kata-kata dokumen yang cocok d_{i-m+1},…,d_i dan sebuah ketidakcocokkan terjadi pada huruf d_{i-m} lalu kita geser akar dari trie ke kanan sebesar beberapa angka dari huruf S yang dihitung dari huruf-huruf dalam dokumen d_{i-m}, …, d_i.



Gambar 4. Ilustrasi fase pencarian

Tentunya, setiap pasangan (w, i) yang ditemukan menggambarkan beberapa kemunculan dari kata kunci W. Jadi algoritma ini menemukan setiap kemunculan dari beberapa kata kunci dalam dokumen D.

C. Fase preprocessing

Masukan dari fase pra-pemrosesan adalah himpunan kata kunci K = {W1,…,Wn}. Keluarannya adalah trie T yang mengandung kata kunci yang dibalik dan fungsi out, shift1, shift2, dan char.

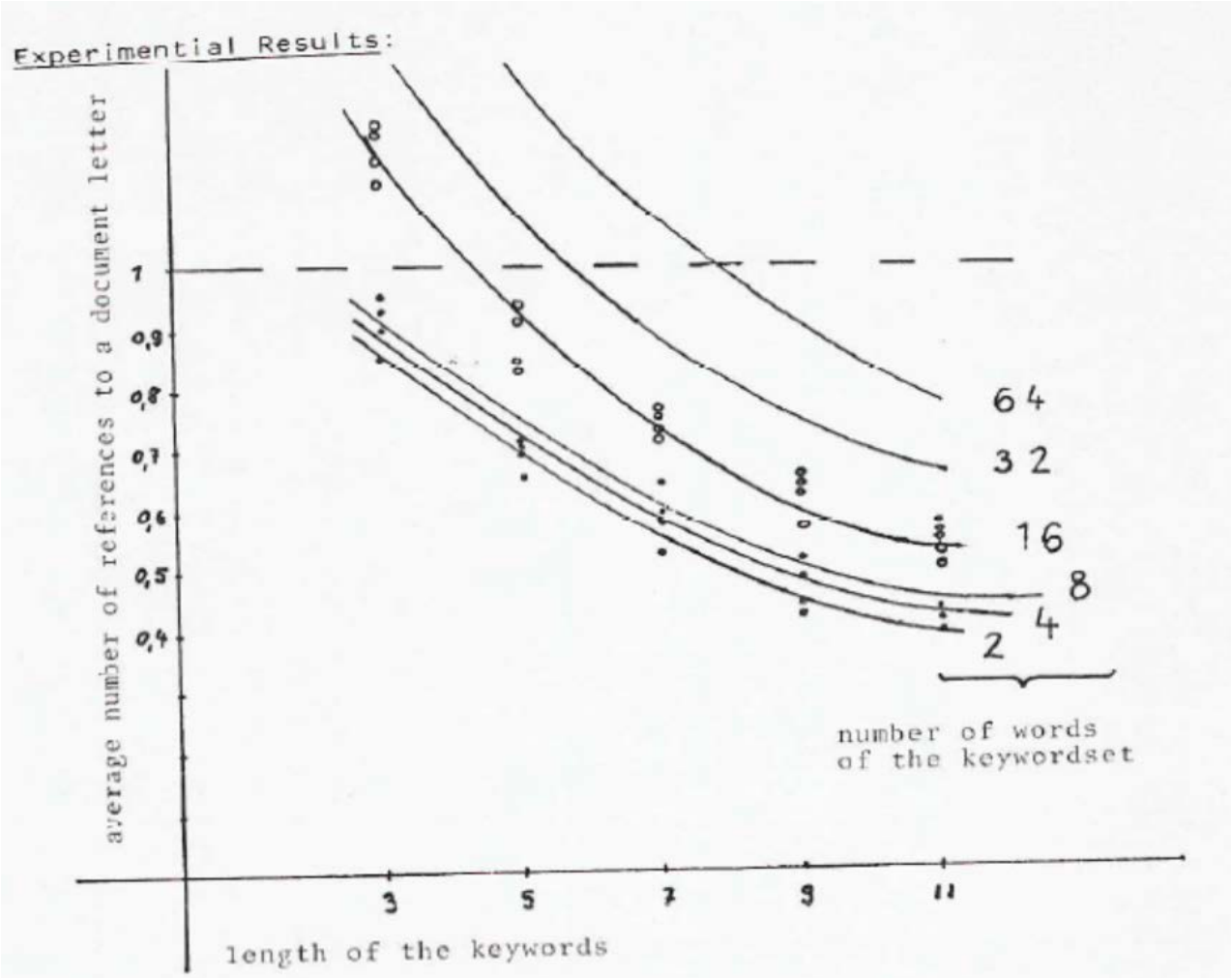
Waktu untuk fase ini adalah linear terhadap total panjang dari kata kunci yaitu W1,…,Wr.

D. Waktu eksekusi

Waktu eksekusi dibagi menjadi dua bagian, yaitu untuk melakukan fase memindai dan melakukan perhitungan shift S(v, d_{i-j}) jika dibutuhkan. Total waktunya adalah linear terhadap total dari banyaknya perbandingan karakter.

Sama seperti algoritma pencocokan string Boyer-Moore, jika ukuran alfabet A nya besar, hanya perlu diperiksa |D| / wmin huruf pada kasus terburuk.

Berdasarkan eksperimen yang dilakukan terhadap 100 judul buku Computer Science dalam bahasa Inggris dan Jerman dan subjek yang terkait. Judul-judul buku adalah dokumen. Alfabet yang digunakan adalah A-Z dan 0-9 dan blank.



Gambar 5. Hasil eksperimen

Gambar di atas menggambarkan bahwa algoritma ini memiliki kompleksitas rata-rata sublinear.

IV. KESIMPULAN

Algoritma pencocokan string Commentz-Walter adalah algoritma yang cukup baik karena menggabungkan ide Boyer-Moore ditambah Aho-Corasick. Algoritma ini cocok untuk teks yang alfabetnya banyak dan bisa mencari beberapa pola sekaligus.

V. UCAPAN TERIMA KASIH

Saya ucapkan terimakasih pada dosen IF2211 karena telah membimbing saya dalam membuat makalah ini.

REFERENSI

[1] Thomas H Corman, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein “Introduction to Algorithms-String matching”, EEE Edition, 2nd Edition, Page no 906-907.

[2] D. Huson , “multiple string matching”, Comp. Sequence Analysis, Nov 17 , 2004

[3] http://www.hs-albsig.de/studium/wirtschaftsinformatik/Documents/commentzwalterextab.pdf

[4] https://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCIQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.402.6234%26rep%3Drep1%26type%3Dpdf&ei=nPVHVaq8JeO1mwWchYHAAw&usg=AFQjCNGVVZg7hGkPufcC5pICWpS8kNVv-Q&sig2=ajxL840CuYHXvP08pNfNuQ&bvm=bv.92291466,d.dGY

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, 4 Mei 2015

Vicko Novianto / 13513092