



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan salah satu algoritma pencarian *string* yang dipublikasikan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977.

Anany Levitin (2007: 259-262) menentukan banyaknya pergeseran algoritma Boyer Moore dengan mempertimbangkan 2 parameter, *Bad-Symbol-Shift* dan *Good-Suffix-Shift*.

2.1.1. Bad Symbol Shift

Parameter pertama ditentukan oleh karakter teks *c* yang menyebabkan ketidakcocokan pada *pattern* yang disebut dengan *bad symbol shift*. Jika *c* tidak terdapat pada *pattern*, maka *pattern* akan digeser melewati *c* pada *text*. Banyaknya pergeseran dapat ditentukan dengan

$$d = t1(c) - k \quad \dots \text{ Rumus 2.1}$$

dimana $t1(c)$ merupakan hasil yang didapat dari tabel *bad symbol shift* dan k adalah banyaknya karakter yang cocok antara teks dan *pattern*.

Tabel 2.1 Tabel *Bad Symbol Shift* untuk BARBER
(Sumber: Anany Levitin, 2007)

c	A	B	C	D	E	F	...	R	...	Z	
t1(c)	4	2	6	6	1	6	6	3	6	6	6

Sebagai contoh pencarian *pattern* BARBER pada teks memiliki kecocokan pada 2 huruf terakhir dan terjadi ketidakcocokan pada huruf S, dapat dilakukan pergeseran *pattern* sebanyak $t1(S) - 2 = 6 - 2 = 4$ huruf, seperti pada contoh

SER
BARBER
BARBER

Rumus 2.1 dapat digunakan pula saat terjadi ketidakcocokan antara teks dan *pattern* jika $t1(c) - k > 0$. Jika $t1(c) - k \leq 0$, tidak mungkin terjadi pergeseran sebanyak 0 atau bahkan posisi negatif (bergeser ke kiri). Oleh sebab itu dapat dilakukan pergeseran *pattern* sebanyak 1 huruf ke kanan.

Dapat disimpulkan untuk mencari *bad symbol shift* $d1$ didapat

$$d1 = \max\{t1(c) - k, 1\} \quad \dots \text{Rumus 2.2}$$

2.1.2. Good-Suffix-Shift

Parameter kedua ditentukan dari pencocokan yang berhasil dari $k > 0$ antara *pattern* dan teks, dengan menggunakan bagian akhir *pattern* / sufiks berukuran k dinyatakan dengan $\text{suff}(k)$. Pergeseran ini disebut dengan *good-suffix-shift*.

Terdapat kasus dimana $\text{suff}(k)$ berulang dalam *pattern*, tetapi tidak didahului karakter yang sama seperti sebelumnya. (Akan sia-sia menggeser *pattern* untuk dicocokkan dengan $\text{suff}(k)$ yang didahului karakter yang sama karena akan mengulang ketidakcocokan). Dalam kasus ini dapat dilakukan penggeseran *pattern* sebanyak $d2$ diantara kejadian kedua yang paling kanan dari $\text{suff}(k)$ dan kejadian paling kanan. Contoh, untuk *pattern* ABCBAB, pergeseran $d2$ untuk $k = 1$ dan 2 adalah 2 dan 4 .

Tabel 2.2. Tabel $d2$ untuk *pattern* ABCBAB
(Sumber: Anany Levitin, 2007)

k	Pattern	$d2$
1	ABC B AB	2
2	AB CBAB	4

Jika tidak ada $\text{suff}(k)$ yang didahului karakter yang sama dengan kejadian sebelumnya, dapat dilakukan pergeseran *pattern* sebanyak panjang *pattern* m . Sebagai contoh, untuk *pattern* DBCBAB dan $k = 3$, dapat dilakukan pergeseran sebanyak panjang *pattern* yaitu 6 karakter. Pergeseran dapat dilihat pada contoh

```

c B A B
D B C B A B
      D B C B A B

```

Pergeseran sebanyak 6 karakter benar untuk DBCBAB, namun tidak untuk ABCBAB, karena *pattern* ABCBAB memiliki substring AB sebagai prefiks (awalan *pattern*). Untuk menghindari kesalahan pergeseran berdasar sufiks berukuran k , dimana tidak ada kejadian lain di *pattern* yang tidak diawali karakter yang sama, dapat dicari prefiks terpanjang L yang sama dengan sufiks dengan panjang yang sama. Jika prefiks tersebut ada, pergeseran d_2 dihitung berdasar jarak diantara prefiks dan sufiks yang bersangkutan. Jika tidak ada, d_2 ditetapkan sebanyak panjang *pattern* m . Berikut adalah contoh daftar lengkap nilai d_2 *good-suffix* algoritma Boyer-Moore untuk *pattern* ABCBAB

Tabel 2.3 Tabel d_2 lengkap untuk *pattern* ABCBAB
(Sumber: Anany Levitin, 2007)

k	pattern	d_2
1	ABC B AB	2
2	AB CBAB	4
3	ABC BAB	4
4	<u>ABC</u> BAB	4
5	ABC BAB	4

Anany Levitin (2007:262) merumuskan cara kerja algoritma Boyer-Moore sebagai:

- a. Untuk huruf yang digunakan pada *pattern* dan teks, buat tabel *bad-symbol-shift*.

- b. Dari *pattern*, buat tabel *good-suffix-shift*.
- c. Sejajarkan *pattern* dengan awal teks.
- d. Ulangi langkah berikut hingga ditemukan *substring* yang cocok atau *pattern* melebihi karakter terakhir teks.

Dimulai dari karakter terakhir *pattern*, bandingkan antara karakter *pattern* dan teks hingga seluruh pasangan m karakter cocok (kemudian berhenti) atau terjadi ketidakcocokan setelah $k \geq 0$. Geser *pattern* ke kanan sebanyak $d = d_1$ jika $k = 0$, atau $d = \max\{d_1, d_2\}$ jika $k > 0$, dimana $d_1 = \max\{t_1(c) - k, 1\}$.

Contoh: Cari BAOBAB dari BESS KNEW ABOUT BAOBABS

Membuat tabel *bad symbol*

Tabel 2.4 Tabel *bad symbol* antara BAOBAB dan text
(Sumber: Anany Levitin, 2007)

C	A	B	C	D	...	O	...	Z	
t1(c)	1	2	6	6	6	3	6	6	6

Membuat tabel *good suffix shift*

Tabel 2.5 Tabel *good suffix shift* dari *pattern* BAOBAB
(Sumber: Anany Levitin, 2007)

k	pattern	d2
1	BAO <u>B</u> AB	2
2	<u>B</u> AOBAB	5
3	<u>BA</u> OBAB	5
4	<u>BAOB</u> AB	5
5	<u>BAOBAB</u>	5

Sejajarkan dan bandingkan *pattern* dengan teks.

Tabel 2.6 Tabel pencocokan *pattern* BAOBAB dengan teks
(Sumber: Anany Levitin, 2007)

B	E	S	S		K	N	E	W		A	B	O	U	T		B	A	O	B	A	B	S		
B	A	O	B	A	B																			
d1 = t1(k)-0=6					B	A	O	B	A	B														
					d1=t1()-2=4					B	A	O	B	A	B									
					d2=5					d1=t1()-1= 5														
					d=max{4,5}=5					d2=2														
																d=max{5,2}=5								
																B	A	O	B	A	B			

B tidak cocok dengan K, $t1(k) = 6$ didapat dari tabel *bad symbol*, dan dilakukan pergeseran *pattern* sebanyak $d1 = \max\{t1(k)-0,1\} = 6$ ke kanan. Berikutnya didapat 2 pasang huruf yang cocok. Pada karakter ketiga (spasi/_) terjadi ketidakcocokan, sehingga $t1(_)=6$ dari tabel *bad symbol* dan $d2=5$ dari tabel *good suffix*. Oleh karena itu dilakukan pergeseran sebanyak $\max\{d1,d2\} = \max\{6-2,5\} = 5$.

Pencocokan berikutnya hanya didapat 1 karakter B yang cocok. $t1(_)=6$, $d2=2$. Pergeseran berikutnya dilakukan sebanyak $\max\{d1,d2\} = \max\{6-1,2\} = 5$. Setelah bergeser sebanyak 5 didapat kecocokan antara seluruh huruf pada *pattern* dengan teks.

2.2. Website

Menurut Coupey (2001:127) *website* merupakan jaringan antar dokumen digital yang disebut dengan halaman web. Halaman web dapat berisi teks, grafik/gambar, format suara, serta format video. Dokumen digital ini terintegrasi dengan *hyperlink* yang memungkinkan pengguna untuk mendapat informasi yang diinginkan dari satu atau beberapa halaman lainnya dengan mudah.

Coupey (2001) membagi *Website* menjadi 3 komponen dasar:

- *HomePage*
Home Page merupakan halaman awal yang muncul ketika pengguna mengakses suatu *website*.
- *Link*
Link merupakan penghubung antar halaman yang digunakan untuk berpindah dari halaman yang satu ke halaman lainnya.

- *Content*

Content merupakan bagian dari *website* yang berupa teks, gambar, suara, video, ataupun hyperlink ke situs lain yang berhubungan.

2.3. World Wide Web dan HTML

Robert W. Sebesta(2009:6) menyatakan *World Wide Web* bermula pada tahun 1989 diprakasai grup kecil yang dipimpin Tim Berners-Lee saat CERN (*Conseil European pour la Recherche Nucleaire* atau *European Laboratory for Particle Physics*). *World Wide Web* atau WWW merupakan sistem informasi hipermedia kolaboratif terdistribusi, sebuah protokol untuk menghubungkan keanekaragaman dokumen yang ditempatkan pada komputer manapun di dalam internet. *Hyper Text Transfer Protocol* (HTTP) yang baru ini memberikan peraturan untuk transaksi sederhana antara dua komputer pada internet yang terdiri dari membangun sebuah koneksi, permintaan pengiriman dokumen, pengiriman dokumen, dan penutupan koneksi. Sistem ini juga memerlukan sebuah format dokumen sederhana yang disebut *Hypertext Markup language* (HTML) untuk mempresentasikan teks terstruktur yang bercampur dengan image inline.

Sebuah dokumen HTML dapat berisi *hyperlink* atau anchor yang mengacu pada dokumen lain yang mirip. Dengan perangkat lunak browser, pengguna selanjutnya dapat mengklik pada area *hot text* suatu dokumen yang didesain dan pindah ke dokumen lainnya. Pengguna dapat melakukan penjelajahan dari satu dokumen ke dokumen lain melalui Web, dengan HTML sebagai kerangka pokok.(Vaughan, 2004)

Menurut Vaughan (2004:294-295), cara kerja web melibatkan komunikasi antara dua komputer, satu server dan satu klien. Server mengirimkan dokumen

ketika klien meminta, sedangkan web browser merupakan aplikasi yang dijalankan komputer pengguna (klien) untuk menyediakan antarmuka grafis yang interaktif untuk mencari, menemukan, dan melihat dokumen teks, suara, animasi, dan sumber-sumber multimedia lain pada Web.

2.4. PHP

Luke Welling dan Laura Thomson (2010:2) mengartikan PHP sebagai sebuah bahasa pemrograman yang berbasis server (*server-side*) yang dikhususkan untuk *web*. Dalam halaman web yang menggunakan HTML, dapat disisipkan kode PHP yang akan dijalankan ketika halaman tersebut dibuka.

PHP merupakan projek *open source*, dimana setiap orang dapat mengakses kode program dan dapat menggunakan, mengubah, dan mendistribusikan tanpa dikenai biaya. Awalnya PHP diartikan sebagai *Personal Home Page*, namun sekarang diubah menjadi *PHP Hypertext Preprocessor*. (Luke Welling dan Laura Thomson, 2010)

2.5 Entity Relationship Diagram(ERD)

ERD atau diagram entitas/relasi merupakan sebuah teknik untuk menggambarkan struktur logis dari sebuah basisdata. ERD menyediakan sebuah arti yang sederhana dan siap dimengerti dari mengkomunikasikan desain dari suatu basis data yang diberikan. (Date,2004)

Date, C. J. (2004) mendefinisikan ERD dalam beberapa elemen:

1. Entitas

Setiap jenis kesatuan diperlihatkan sebagai sebuah persegi panjang yang berisi nama dari entitas tersebut.

2. Properti

Diperlihatkan sebagai elips yang berisi nama properti dan menempel dengan entitas relevan atau relasi dengan garis yang kokoh.

3. Relasi

Berbentuk wajik dan berisi nama dari jenis relasi.

2.6 FlowChart

Menurut Krismiaji (2005), *flowchart* atau bagan alir merupakan teknik analitis yang digunakan untuk menjelaskan aspek-aspek sistem informasi secara jelas, tepat dan logis. Bagan alir menggunakan serangkaian simbol standar untuk menguraikan prosedur pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran data dalam sebuah sistem.

Krismiaji (2005) membedakan jenis bagan alir yang biasa digunakan, yaitu sebagai berikut:

1. Bagan Alir Sistem (*System Flowchart*)

Bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem serta menunjukkan apa yang dikerjakan di dalam sistem.

2. Bagan Alir Dokumen (*Document Flowchart*)

Disebut juga bagan alir formulir (*Form Flowchart / paperwork Flowchart*). Bagan alir ini menunjukkan arus dari laporan dan formulir termasuk tembusannya. Menggunakan simbol yang sama dengan bagan alir sistem.

3. Bagan Alir Skematik(*Schematic Flowchart*)

Bagan ini menggambarkan prosedur dalam sistem. Bagan ini menggunakan simbol yang sama dengan bagan alir sistem dan ditambah dengan gambar-gambar komputer serta peralatan lain yang digunakan.

4. Bagan Alir Program(*Program Flowchart*)

Terdiri atas 2, Bagan alir logika program(*Program Flowchart*) yang digunakan untuk menggambarkan langkah di dalam program komputer secara logika, dan bagan alir program komputer terinci(*Detailed Computer Program Flowchart*). Bagan alir logika program dipersiapkan oleh analis sistem.

5. Bagan Alir Proses(*Process Flowchart*)

Bagan alir yang digunakan analis sistem untuk menggambarkan proses dalam suatu prosedur.

2.7 Sampling Data

2.7.1 Pengertian Sampel

Sampel adalah bagian dari populasi (sebagian atau wakil populasi yang diteliti). Sampel penelitian adalah sebagian dari populasi yang diambil sebagai sumber data dan dapat mewakili seluruh populasi. (Indriani, 2013)

Indriani (2013) menyatakan keuntungan dalam menggunakan sampel yaitu: memudahkan peneliti, penelitian lebih efisien, lebih teliti dan cermat dalam pengumpulan data, serta penelitian lebih efektif.

2.7.2 Syarat Sampel yang Baik

Menurut (Indriani, 2013), syarat dari sampel yang baik antara lain adalah:

1. Akurasi atau ketepatan

Tingkat ketidakadaan “bias” (kekeliruan) dalam sample. Makin sedikit tingkat kekeliruan yang ada dalam sampel, makin akurat sampel tersebut.

2. Presisi

Kriteria kedua sampel yang baik adalah memiliki tingkat presisi estimasi. Presisi mengacu pada persoalan sedekat mana estimasi kita dengan karakteristik populasi. Presisi=*standard error*, Nilai rata-rata populasi dikurangi nilai rata-rata sampel.

Indirani (2013) juga menjelaskan teknik (metode) penentuan sampel yang ideal memiliki ciri-ciri dapat memberikan gambaran yang akurat tentang populasi, dapat menentukan presisi, sederhana sehingga mudah dilaksanakan, dan dapat memberikan keterangan sebanyak mungkin dengan biaya murah.

2.7.3 Teknik Sampling

Indriani (2013) mendefinisikan Teknik pengambilan sampel atau teknik sampling sebagai suatu cara pengambilan sampel yang representatif dari populasi. Untuk menentukan sampel dalam penelitian, terdapat berbagai teknik sampling yang digunakan.

Teknik sampling dikelompokkan menjadi *Probability Sampling* dan *Nonprobability Sampling*. (Indriani, 2013).

Probability Sampling adalah teknik pengambilan sampel yang memberikan peluang yang sama bagi setiap unsur (anggota) populasi untuk dipilih menjadi

anggota sampel. Sedangkan *NonProbability Sampling* adalah teknik pengambilan sampel yang tidak memberi peluang/kesempatan sama bagi setiap unsur atau anggota populasi untuk dipilih menjadi sampel. (Indriani, 2013)

Menurut Indirani(2013) salah satu jenis *sampling* yang merupakan *nonprobability sampling* adalah *sampling jenuh*. *Sampling Jenuh* adalah teknik penentuan sampel bila semua anggota populasi digunakan sebagai sampel. Hal ini sering dilakukan bila jumlah populasi relatif kecil, kurang dari 30 orang, atau penelitian yang ingin membuat generalisasi dengan kesalahan yang sangat kecil.



U M N