

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Pada perpustakaan sekolah, katalog perpustakaan masih berupa katalog fisik dari kertas karton. Katalog diurutkan berdasarkan klasifikasi katalog perpustakaan. Berikut ini adalah format kartu katalog perpustakaan.

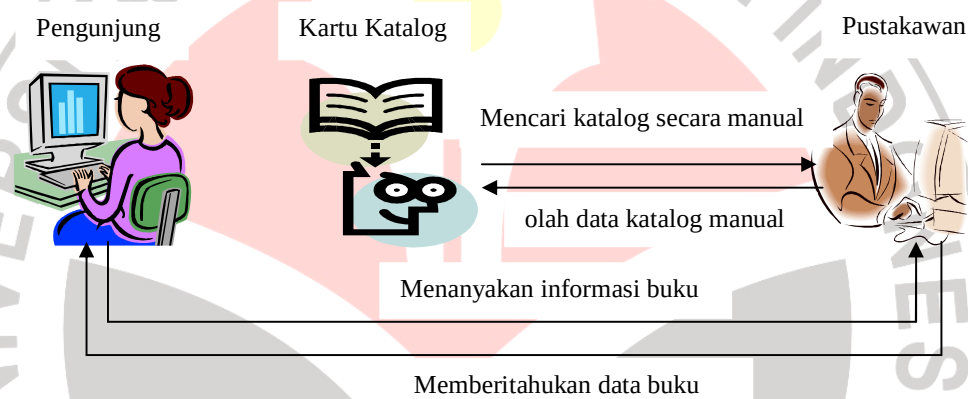
499.221 YWD C	Wulandari, Yayan Cerdas Berbahasa Indonesia Sesuai EYD untuk SD,SMP,SMA : Yudhistira, 2008. ISBN : 9786028661348
	1. Bahasa Indonesia

Gambar 4.1 Format Kartu Katalog Perpustakaan Sekolah

Berdasarkan hasil observasi dan wawancara yang telah dilakukan oleh penulis maka diperoleh beberapa data penelitian berupa format data katalog perpustakaan

sekolah yang meliputi no.isbn, judul buku, jenis buku, penulis, penerbit, jumlah buku, tahun terbit, kode buku, kode penulis, kode penerbit, kode rak dan kondisi.

Penelusuran katalog pada perpustakaan sekolah masih menggunakan cara manual sehingga pencarian katalog buku berjalan lambat. Alur kerja pencarian katalog perpustakaan sekolah tanpa sistem penelusuran katalog yang terkomputerisasi adalah sebagai berikut.



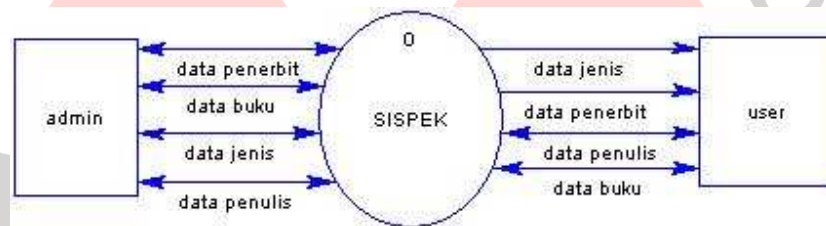
Gambar 4.2 Alur Kerja Penelusuran Katalog Manual

4.2 Pengembangan Perangkat Lunak

Pengembangan perangkat lunak system penelusuran katalog perpustakaan sekolah menggunakan metode sekuensial linier (*waterfall*). Beberapa tahap metode *waterfall* yang dilakukan oleh penulis adalah analisis, desain, coding dan test (pengujian).

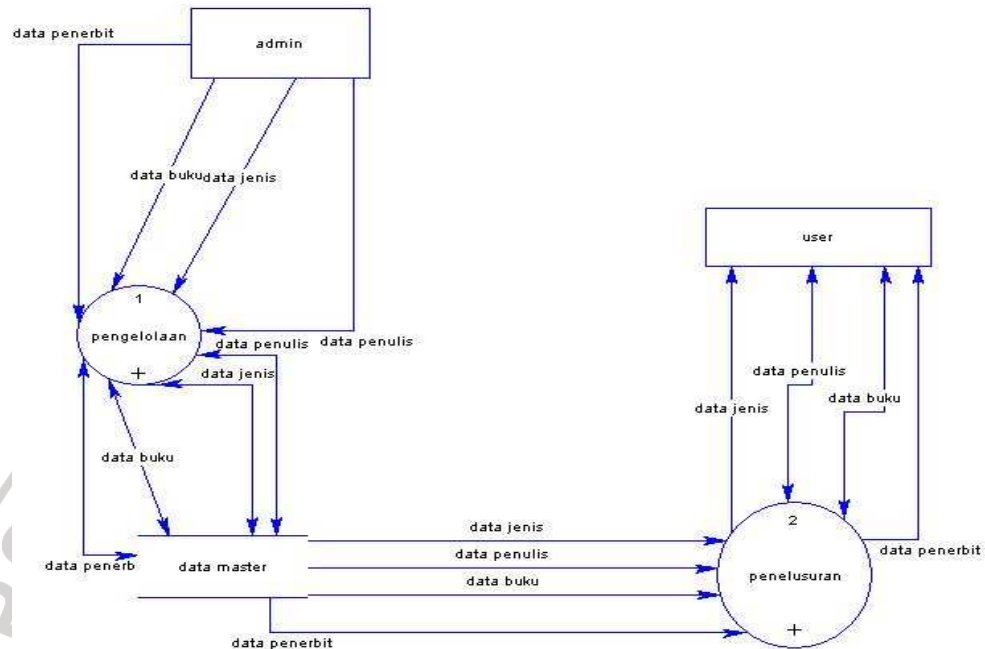
4.2.1 Analisis

Pada tahap ini penulis melakukan analisis kebutuhan sistem agar dapat mengetahui arus proses dan data sistem yang akan dibangun. Model sistem penelusuran katalog perpustakaan sekolah dirancang menggunakan context diagram dan data flow diagram. Context diagram dan data flow diagram dapat dilihat pada Gambar 4.1 dan Gambar 4.2. Analisis sistem selengkapnya dapat dilihat pada Dokumen Teknis.



Gambar 4.3 Context Diagram Sistem Penelusuran Katalog

Dari diagram tersebut dapat dilihat bahwa admin dapat memasukkan data master dan login dengan id admin. Anggota hanya dapat memasukkan kata kunci katalog dan sistem akan memberikan hasil penelusuran berupa data katalog.

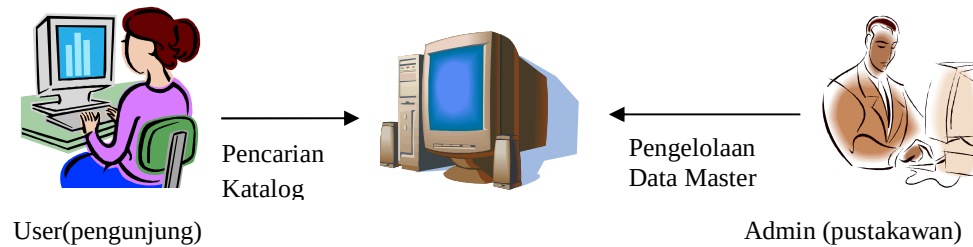


Gambar 4.4 Data Flow Diagram Level 1 Sistem Penelusuran Katalog

Dari diagram diatas terlihat bahwa terdapat empat proses utama yang menyusun sistem penelusuran katalog yaitu proses autentifikasi, setup data, proses penelusuran, dan pembuatan laporan. Keterangan selengkapnya mengenai DFD level 1 dapat dilihat pada Dokumen Teknis.

4.2.2 Desain

Sistem penelusuran katalog perpustakaan sekolah merupakan sebuah sistem yang dibangun untuk membantu pengunjung perpustakaan sekolah untuk mencari dan melihat informasi mengenai buku yang ingin dipinjam. Selain itu sistem ini juga dapat membantu pustakawan dalam mengelola data buku perpustakaan. Berikut ini adalah desain sistem yang akan dibangun.

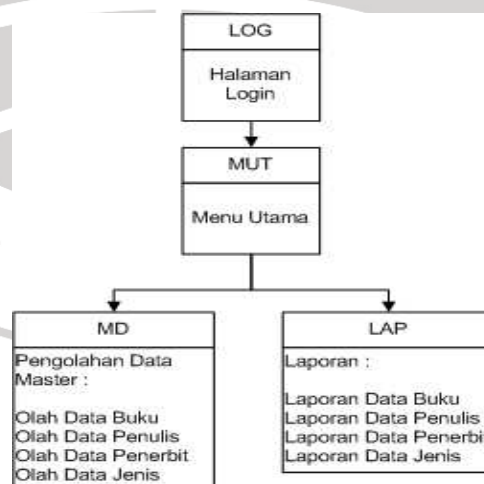


Gambar 4.5 Desain Sistem Penelusuran Katalog

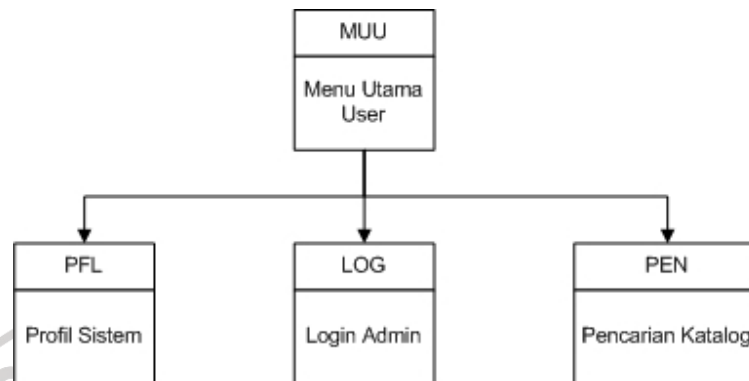
Pada tahap ini penulis melakukan perancangan antarmuka perangkat lunak sistem penelusuran perpustakaan sekolah. Beberapa fitur yang akan dirancang dalam perangkat lunak yaitu :

1. Penelusuran katalog berdasarkan kategori judul dan penulis .
2. Data Master (pengolahan data buku) : judul buku, jenis buku, isbn, penulis, penerbit, tahun terbit, kondisi dan rak buku.

Perancangan antarmuka untuk user dan admin adalah sebagai berikut.



Gambar 4.6 Desain Antarmuka Admin



Gambar 4.7 Desain Antarmuka User

4.2.3 Coding

Dari hasil analisis dan desain yang telah dilakukan, penulis mengubahnya ke dalam bentuk *code* dengan cara mengimplementasikan hasil analisis dan desain tersebut ke dalam bahasa pemrograman web dengan menggunakan algoritma Boyer-Moore. Berikut ini adalah pseudocode algoritma Boyer-Moore pada fase pra-pencarian dan fase pencarian :

1. Fase Preprocessing

```

procedure preBmBc(
  input P : array[0..n-1] of char,
  input n : integer,
  input/output bmBc : array[0..n-1] of integer
)

```

Deklarasi:

i: integer

Algoritma:

```

for (i := 0 to ASIZE-1)
  bmBc[i] := m;
endfor
for (i := 0 to m - 2)
  bmBc[P[i]] := m - i - 1;
endfor

```

```

procedure preSuffixes(
  input P : array[0..n-1] of char,
  input n : integer,

```

```

    input/output suff : array[0..n-1] of integer
)

```

Deklarasi:
 f, g, i: integer

Algoritma:

```

suff[n - 1] := n;
g := n - 1;
for (i := n - 2 downto 0) {
  if (i > g and (suff[i + n - 1 - f] < i - g))
    suff[i] := suff[i + n - 1 - f];
  else
    if (i < g)
      g := i;
    endif
    f := i;
    while (g >= 0 and P[g] = P[g + n - 1 - f])
      --g;
    endwhile
    suff[i] = f - g;
  endif
endfor

```

```

procedure preBmGs(
  input P : array[0..n-1] of char,
  input n : integer,
  input/output bmBc : array[0..n-1] of integer
)

```

Deklarasi:
 i, j: integer
 suff: array [0..RuangAlpabet] of integer

```

preSuffixes(x, n, suff);

```

```

for (i := 0 to m-1)
  bmGs[i] := n
endfor

```

```

j := 0
for (i := n - 1 downto 0)
  if (suff[i] = i + 1)
    for (j:=j to n - 2 - i)
      if (bmGs[j] = n)
        bmGs[j] := n - 1 - i
      endif
    endfor
  endif
endfor

```

```

for (i = 0 to n - 2)
  bmGs[n - 1 - suff[i]] := n - 1 - i;
endfor

```

```

Procedure compute_last_occurrence_function(var p: string,
occ: array of integer, m: integer)
Declaration
c: char
i , j, a: integer;
Begin
    //initialize the occ array
    For i ← 0 to ALPHABET_SIZE do
        occ[i] ← -1
    For j ← 0 to m do
        Begin
            C ← P[j]
            occ[a] ← j
        End_for
    End_of_Procedure
Procedure goodprecomp1(var p: string, s: string,
f:array_of_integer, m: integer)
    Declaration
        i, j : integer
    Begin
        i ← m
        j ← m+1
        f[i] ← j
        while i>0 do
            begin
                while j<=m and p[i=1]?p[j-1] do
                    begin
                        if s[j] = 0 then s[j] ← j-1
                        j ← f[j]
                    end_while
                i ← i-1
                j ← j-1
                f[i] ← j
            end_while

```



```

        end_while
    End_of_procedure

```

2. Fase Pencarian

```

procedure BoyerMooreSearch(
    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, shift, bmBcShift, bmGsShift: integer
BmBc : array[0..255] of integer
BmGs : array[0..n-1] of integer

```

Algoritma:

```

preBmBc(n, P, BmBc)
preBmGs(n, P, BmGs)
i:=0
while (i<= m-n) do
    j:=n-1
    while (j >= 0 and T[i+j] = P[j]) do
        j:=j-1
    endwhile
    if(j < 0) then
        ketemu[i]:=true;
    endif
    bmBcShift:= BmBc[chartoint(T[i+j])]-n+j+1
    bmGsShift:= BmGs[j]
    shift:= max(bmBcShift, bmGsShift)
    i:= i+shift

```

Algoritma Boyer-Moore kemudian diimplementasikan ke dalam sistem penelusuran katalog perpustakaan sekolah menggunakan bahasa pemrograman web yaitu php. Hasil implementasi algoritma Boyer-Moore tersebut berupa skrip php yang digunakan pada fitur Pencarian Katalog yaitu pencarian.php. Skrip pencarian.php tersebut adalah sebagai berikut.

```

<?php

include "include/koneksi.php";

$data_cari = $_REQUEST['data_cari'];

$sql = "SELECT * FROM buku";

$minta= mysql_query($sql);
$pat  = strlen(trim($data_cari));
$text = strlen(trim($pat));
$hasil= ($text/$pat) ; // kompleksitas waktu boyer moore
mysql_query($sql, $connect)
    or die ("SQL Error".mysql_error());
if (! empty($sql)) {
    echo "<table class=table_utama width=99%
    border=0 cellpadding=1 >
    <tr bgcolor=#0066FF>
    <td> Jenis Buku </td>
    <td> Judul Buku </td>
    <td> Penulis </td></tr>";

    $load_time = microtime();
    $load_time = explode(' ', $load_time);
    $load_time = $load_time[1] + $load_time[0];
    $page_end = $load_time;
    $final_time = ($page_end-$page_start);
    $page_load_time = number_format($final_time, 4, '.', '');

    while($hs=mysql_fetch_array($minta))
    {
        echo "<tr>
        <td>$hs[jenis_buku] </td>
        <td><a href=?act=detail_cari&no_buku=$hs[no_buku]> $hs[judul]
        </a></td>
        <td> $hs[penulis] </td></tr>";
    }
    echo "</table><br>";
    echo "Running time penelusuran : <b>$hasil</b> millisecond.";
    echo "<p><a href=javascript:history.back(1)>Back
    </a>";
}

}

?>

```

Hasil implementasi dapat dilihat pada tabel Tabel 4.1 dibawah ini.

Hasil implementasi lebih lengkap dapat dilihat pada Dokumen Teknis.

Tabel 4.1 Hasil Implementasi

No	Nama File	Keterangan
1	index.php	Halaman awal sistem yang merupakan penghubung antarhalaman system
2	administrator.php	Halaman login administrator
3	cek_admin.php	File ini berfungsi sebagai autentifikasi username dan password administrator
4	session.php	File ini berfungsi sebagai perekaman data admin agar dapat mengakses setiap halaman
5	menu_utama.php	Halaman ini menyediakan daftar fitur yang dapat digunakan oleh administrator
6	olah_buku.php	File ini membantu administrator untuk mengedit atau menghapus data buku perpustakaan
7	laporan.php	Halaman ini berfungsi menampilkan data koleksi perpustakaan
8	form_cari.html	File ini berfungsi menampilkan form pencarian katalog
9	pencarian.php	File ini berfungsi menampilkan hasil pencarian katalog secara singkat.
10	cari_detail.php	File ini berfungsi menampilkan data buku secara detail

4.2.4 Test

Pada tahap ini penulis melakukan beberapa pengujian terhadap fitur-fitur perangkat lunak sistem penelusuran katalog perpustakaan sekolah yang telah dibangun. Selain itu untuk mengetahui kecepatan algoritma Boyer-Moore penulis akan menghitung running time waktu algoritma Boyer-Moore.

Lingkungan perangkat keras yang digunakan dalam proses pengujian adalah sebagai berikut :

1. Processor Intel Centrino 2.1 GHz
2. RAM 2 GB
3. Hard Disk 100 GB

Lingkungan perangkat lunak yang digunakan dalam proses pengujian adalah sebagai berikut :

1. Sistem Operasi Windows Vista SP 1
2. Browser Mozilla Firefox 3.6.10
3. XAMPP 1.7.3

Bentuk dari pengujian yang dilakukan adalah pengujian kebenaran fungsional unit perangkat lunak, pengujian yang dilakukan menggunakan teknik pengujian *Black Box*. Pengujian *blackbox* dilakukan untuk menemukan kesalahan yang terjadi seperti fungsi yang tidak benar/hilang, kesalahan *interface*, kesalahan struktur data, kesalahan kinerja, atau kesalahan inisialisasi dan terminasi (Pressman, 2002: 551). Pelaksanaan pengujian dapat dilihat pada Tabel 4.2 dibawah ini. Untuk hasil pengujian selengkapnya dapat dilihat pada Dokumen Teknis.

Tabel 4.2 Pelaksanaan Pengujian

No	Bentuk pengujian	Metode	Data uji	Tujuan
1	Pengujian login admin	Black box	Id Admin dan password Admin	Melihat koneksi database dan keamanan sistem pada fitur admin login
2	Pengujian penelusuran katalog	Black box	Data sembarang dapat mewakili data buku dan data penulis	Melihat proses penelusuran dan hasil penelusuran
3	Pengujian pengelolaan data master	Black box	Data sembarang dapat mewakili data buku, data penulis, data jenis dan data penerbit	Melihat proses input dan output

4.3 Pembahasan

Berikut ini adalah hasil perhitungan kompleksitas waktu dari tiap algoritma menggunakan test case berupa target text dan pattern dengan panjang yang bervariasi. Panjang string text (text size) yang digunakan adalah 100, sedangkan panjang pattern yang digunakan bervariasi antara 1 sampai 18 karakter. Pattern merupakan jumlah string yang dicari dan text adalah jumlah

string yang tersedia pada database sistem penelusuran katalog. Panjang pattern sesuai dengan panjang string dari kata kunci yang dimasukkan pada form pencarian katalog. Hasil pengujian running time algoritma dapat dilihat pada Tabel 4.1 dibawah ini. Satuan running time algoritma adalah millisecond.

Tabel 4.3 Hasil Pengujian Kecepatan Running Time Boyer-Moore

Pattern Size	Running Time Boyer-Moore (millisecond)
3	33.3333
4	25
5	20
9	11.1111
10	10
11	9.09091
12	8.33333
15	6.66667
18	5.55556

Berikut ini adalah keterangan dari perhitungan running time Boyer-Moore pada Tabel 4.1.

Contoh : $m = 4$, m adalah pattern size ; dan $n = 100$, n adalah text size. Maka Running time = $(n/m) = (100/4) = 25$ millisecond

Berdasarkan hasil pengujian yang diperoleh pada Tabel 4.1 penulis memperoleh kesimpulan dalam implementasi algoritma Boyer-Moore ke dalam sistem penelusuran katalog perpustakaan sekolah yaitu semakin besar jumlah pattern yang dicari maka semakin kecil running time algoritma Boyer-Moore. Pada jumlah pattern (m) 6 karakter, maka running time algoritma

Boyer-Moore yang diperoleh sebesar 16.66 millisecond. Sedangkan pada m sebesar 18 karakter, running time algoritma Boyer-Moore yang diperoleh sebesar 5,55 millisecond.

Berikut ini penjelasan mengenai proses pencarian katalog dengan algoritma Boyer-Moore berdasarkan memasukkan kata kunci pada form pencarian katalog. Misalnya string yang akan dicari adalah “musik” dari judul “terampil bermain musik”. Maka proses pencarian string “musik” adalah sebagai berikut.

Pattern : musik Text : Terampil Bermain Musik

1. Preprocessing tahap pertama

Pada preprocessing tahap pertama, pencocokan string pattern dimulai dengan mencocokkan string pattern dengan deretan string text yang pertama “terampil”.

Tabel 4.4 Preprocessing Tahap Pertama

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t	e	r	a	m	p	i	l		b	e	r	m	a	i	n		m	u	s	i	k
m	u	s	i	k																	

Keterangan :

Pada tahap ini tidak ada satu pun karakter pattern yang cocok dengan karakter text sehingga pergeseran string pattern sebesar jumlah string pattern sebanyak 5 posisi, pencarian berikutnya dimulai dari karakter keenam dari text.

2. Preprocessing tahap kedua

Tabel 4.5 Preprocessing Tahap Kedua

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t	e	r	a	m	p	i	l		b	e	r	m	a	i	n		m	u	s	i	k
					m	u	s	i	k												

Keterangan :

Pada tahap ini akhir karakter pattern “k” tidak cocok dengan karakter “b” pada text. Karakter “i” tidak cocok dengan spasi. Tidak ada satu pun karakter yang cocok. Pergeseran string sebesar 5 posisi.

3. Preprocessing tahap ketiga

Tabel 4.6 Preprocessing Tahap Ketiga

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t	e	r	a	m	p	i	l		b	e	r	m	a	i	n		m	u	s	i	k
										m	u	s	i	k							

Keterangan :

Pada tahap ini tidak ada satu pun karakter pattern yang cocok dengan karakter text. Nilai pergeseran string sebesar satu karakter. Pencarian string dilanjutkan pada string ke-12 dari string text, dimulai pada karakter “r”.

4. Bad Character Rule

Tabel 4.7 Bad Character Rule

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t	e	r	a	m	p	i	l		b	e	r	m	a	i	n		m	u	s	i	k
											m	u	s	i	k						

Keterangan :

Pada tahap ini terdapat satu karakter yang memiliki kecocokan yaitu pada posisi string ke-15, karakter “i”. Nilai pergeseran adalah satu karakter.

5. Good Suffix Rule

Tabel 4.8 Good Suffix Rule

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t	e	r	a	m	p	i	l		b	e	r	m	a	i	n		m	u	s	i	k
												m	u	s	i	k					

Keterangan :

Pada tahap ini terdapat satu karakter yang memiliki kecocokan yaitu pada posisi string ke-13, karakter “m”. Nilai pergeseran adalah tiga karakter.

Pencarian string dilanjutkan pada posisi string ke-16 pada karakter “n” string text.

6. Bad Character Shift Rule

Tabel 4.9 Bad Character Shift Rule

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t	e	r	a	m	p	i	l		b	e	r	m	a	i	n		m	u	s	i	k
															m	u	s	i	k		

Keterangan :

Pada tahap ini tidak ada satu pun karakter pattern yang cocok dengan karakter text. Nilai pergeseran string sebesar dua karakter. Pencarian dilanjutkan pada posisi string ke-18.

7. Tahap Good Suffix Shift Rule

Tabel 4.10 Good Suffix Shift Rule

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t	e	r	a	m	p	i	l		b	e	r	m	a	i	n		m	u	s	i	k
																	m	u	s	i	k

Keterangan :

Pada tahap ini semua karakter pattern cocok dengan karakter text. Pencarian string berhasil dilakukan.