BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Dalam membangun perangkat lunak ini dilakukan beberapa tahap analisis yaitu :

1. Analisis Masalah
2. Analisis Algoritma
3. Spesifikasi kebutuhan perangkat lunak
4. Analisis Kebutuhan Non Fungsional
5. Analisis Kebutuhan Fungsional

3.1.1. Analisis Masalah

Dari hasil pengamatan diketahui bahwa rata-rata pengunjung yang datang ke rumah baca buku sunda adalah untuk mencari buku atau bacaan dengan terbitan lawas, karena memang rumah baca buku sunda menyediakan buku-buku dengan terbitan lama. Timbul masalah ketika pengunjung tidak mengingat judul buku atau pengarang buku yang akan mereka cari dan hanya mengingat nama tokoh dalam buku tersebut atau kata yang sering muncul dalam buku sehingga pengunjung mengalami kesuliatan dalam melakukan pencarian.

Oleh sebab itu perlu dibangunnya perangkat lunak pencarian kata yang dapat menyelesaikan permasalah tersebut dengan tepat dan juga cepat. Sehingga dapat membantu calon pembaca menemukan buku yang dicari. Ada sekitar 35 algoritma pencarian kata yang bisa digunakan dalam perangkat baik merupakan algoritma yang diciptakan dari awal maupun berupa pengembangan dari algoritma yang sudah ada [**1**]. Dua di antaranya yaitu algoritma Karp-Rabin dan Algoritma Zhu-Takaoka. Dengan melakukan analisis perbandingan performansi dari algoritma Karp-Rabin dan algoritma Zhu-Takaoka maka akan dapat diketahui cara kerja dan performansi dalam kecepatan dan ketepatan dari kedua algoritma tersebut. Agar selanjutnya algoritma yang lebih mangkus dapat digunakan pada perangkat lunak pencocokan kata.

3.1.2. Analisis Algoritma

Pembuatan program komputer tidak terlepas dari algoritma, apalagi program yang dibuat sangat kompleks. Analisis algoritma sangat membantu di dalam meningkatkan efesiensi program. Kecanggihan suatu program bukan dilihat dari tampilan program, tetapi berdasarkan efisiensi algoritma yang terdapat didalam program tersebut. Program dapat dibuat dengan mengabaikan algoritma, tetapi jangan heran bila ada program yang mirip tetapi memiliki akses yang lebih cepat dan memakai memori yang sangat sedikit. Analisis algoritma adalah bahasan utama dalam ilmu komputer. Dalam menguji suatu algoritma, dibutuhkan beberapa kriteria untuk mengukur efisiensi algoritma. Terdapat dua tipe analisis algoritma [**10**] yaitu :

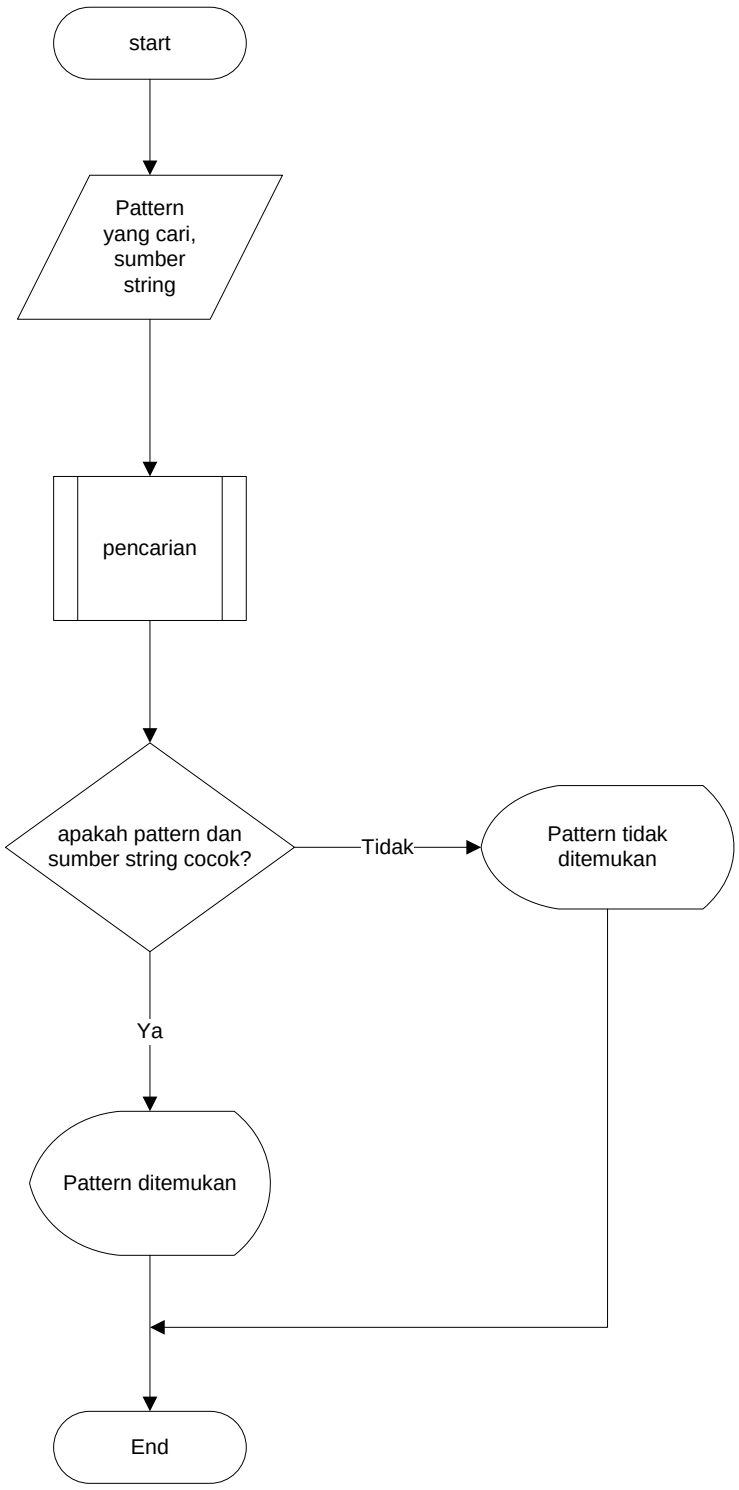
1. Memeriksa kebenaran algoritma dapat dilakukan dengan cara perurutan, memeriksa bentuk logika, implementasi algoritma, pengujian dengan data dan menggunakan cara matematika untuk membuktikan kebenaran.
2. Penyederhanaan Algoritma Membagi algoritma menjadi bentuk yang sederhana.

3.1.2.1.Analisis Algoritma Zhu-Takaoka

Pada Gambar 3.1 Flowchart Algoritma Zhu-Takaoka dapat dilihat alur kerja algoritma Zhu-Takaoka.

Algoritma BM‟ (Algoritma Zhu-Takaoka) yang merupakan modifikasi dari Algoritma Boyer Moore mempunyai ciri-ciri yang sama dalam proses pencarian string. Ciri-ciri tersebut yaitu terbagi dua fase yaitu fase preprocessing dan fase pencarian. Perbedaan antara Algoritma Boyer-Moore dan Algoritma Zhu-Takaoka yaitu terletak pada tahap penentuan bad character rule. Dalam Boyer-Moore, bad character hanya terdiri array satu dimensi, sedangkan dalam Zhu-Takaoka dimodifikasi menjadi array dua dimensi. Karakteristik Algoritma Zhu-Takaoka

1. Pengembangan dari algoritma Boyer-Moore
2. Menggunakan array dua dimensi untuk menghitung nilai pergeseran.
3. Melakukan pencocokan dari kanan ke kiri



Gambar 3.1 Flowchart Algoritma Zhu-Takaoka

Tabel 3-1 Pseudocode Algoritma Zhu-Takaoka dengan notasi Big-O

```
Procedure ZT(input x : array of char,input m :integer,input
y :array of char , input n : integer)
{IS : pencocokan string dengan algoritma zhu-takaoka
 FS : keluaran yang diharapkan hasil dari pencocokan}

Kamus
   i, j    : integer
   ztBc    : array[0..ASIZE][0..ASIZE] of integer
   bmGs    : array[0..XSIZE] of integer

algorimta

   {preprocessing}
   preZtBc(x, m, ztBc);
   preBmGs(x, m, bmGs);
   {pencarian}
   j = 0;                                          O(1)
   while (j <= n - m) {                            O(n)
      i ← m - 1;                                   O(1)
      while (i < m and x[i] = y[i + j]; --i)       O(n)
         if (i < 0) {                              O(1)
         OUTPUT(j);                                O(1)
         j ← j + bmGs[0];                          O(1)
      }
      else
         j ← j + MAX(bmGs[i],ztBc[y[j + m - 2]][y[j + m -
1]]);                                              O(1)
        endif
      endwhile
  endwhile
Endprocedure
```

Perhitungan Big-O

Tabel 3-2 Perhitungan Big-O algoritma Zhu-Takaoka

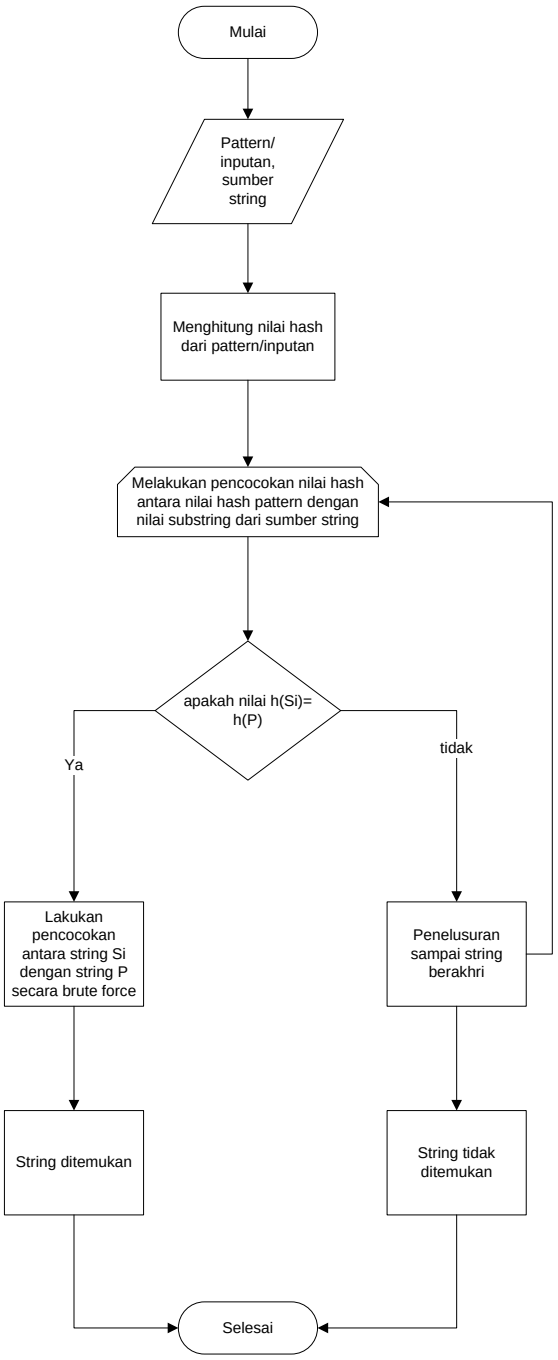
Pseudocode	Nilai Big-O
j = 0;	**O(1)**
while (j <= n - m)	**O(n)**
i ← m - 1;	**O(1)**

while (i < m and x[i] = y[i + j]; --i)	**O(n)**
if (i < 0)	**O(1)**
OUTPUT(j)	**O(1)**
j ← j + bmGs[0];	**O(1)**
j ← j + MAX(bmGs[i],ztBc[y[j + m - 2]][y[j + m - 1]]);	**O(1)**
Jumlah	**O(n^2)**

Berdasarkan hasil perhitungan performansi algoritma Zhu-Takaoka menggunakan notasi Big-O didapat kompleksitas waktu dengan **O(n^2)** dengan n adalah ukuran inputan **.** Yang mempengaruhi nilai kompleksitas waktu dari algoritma Zhu-Takaoka adalah nilai n karena n berpangkat dua.

3.1.2.2.Analisis Algoritma Karp-Rarbin

Pada dapat dilihat Gambar 3.2 Flowchart Algoritma Karp-Rabin alur kerja algoritma Karp-Rabin.



Gambar 3.2 Flowchart Algoritma Karp-Rabin

Algoritma Karp-Rabin ini tidak melakukan pergeseran yang rumit untuk menyelesaikan masalah, algoritma ini mempercepat pengecekan kata pada suatu teks dengan menggunakan fungsi hash dalam mencari suatu string. Fungsi *hash*

adalah fungsi yang menerima masukan string yang panjangnya sembarang dan mengkonversinya menjadi string keluaran yang panjangnya tetap (fixed) umumnya berukuran jauh lebih kecil daripada ukuran string semula [**12**]. Pada algoritma ini untaian string akan diubah menjadi integer berdasarkan bilangan ASCII-nya. Pendekatan utamanya adalah, string yang sama akan memiliki nilai hash yang sama.

Hal yang penting yang harus dilakukan sebelum melakukan pencocokan dengan algoritma Karp-Rabin adalah mengubah sumber string dan pattern yang dicari menjadi untaian integer. Karakteristik algoritma Karp-Rabin

1. Menggunakan Fungsi Rolling Hashing.
2. Melakukan pencocokan dari kiri ke kanan
3. Pergeseran dilakukan secara brute-force

Pseudocode Algoritma Karp-Rabin

Jika algoritma Rabin-Karp ditulis secara keseluruhan dalam *pseudocode* :

Tabel 3-3 Pseudocode Algoritma Karp-Rabin dengan notasi Big-O

```
function RabinKarp (input s: string[1..m], teks: string[1..n])
{IS :  Melakukan pencarian string s pada string teks dengan
algoritma Rabin Karp
FS : keluaran yang diharapkan pencarian string berhasil dilakukan}
}

Deklarasi
    i : integer
    ketemu = boolean

Algoritma:
 ketemu ← false                              O(1)
  hs ← hash(s[1..m])                         O(1)
    for i ← 1 to n-m+1 do                    O(n)
      hsub ← hash(teks[1..i+m-1])            O(1)
        if hsub = hs then                    O(1)
          if teks[i..i+m-1] = s then         O(1)
            ketemu ← true                    O(1)
          else
```

```
            hsub ← hash(teks[i+1..i+m])   O(1)
          endif
        endif
    endfor
          return ketemu                   O(1)
endfunction
```

Perhitungan Big-O

Tabel 3-4 Peritungan Big-O

Pseudocode	Big-O
ketemu ← false	O(1)
hs ← hash(s[1..m])	O(1)
for i ← 1 to n-m+1 do	O(n)
hsub ← hash(teks[1..i+m-1])	O(1)
if hsub = hs then	O(1)
if teks[i..i+m-1] = s then	O(1)
ketemu ← true	O(1)
hsub ← hash(teks[i+1..i+m])	O(1)
return ketemu	O(1)
Jumlah	**O(n)**

Berdasarkan hasil perhitungan performansi algoritma Karp-Rabin menggunakan notasi Big-O didapat kompleksitas waktu dengan **O(n) .** dengan n adalah ukuran inputan.

3.1.2.3. Perbandingan Algoritma

Sample Sinopsi yang diambil dari rumah baca buku sunda yang dijadikan sample penelitian dalam perhitungan performansi algoritma Zhu-Takaoka dan algoritma Karp-Rabin dikategorikan berdasarkan jumlah karakter deskripsi buku bisa dilihat pada tabel Tabel 3-5 Data Buku dan kombinasi inputan pencarian seperti :

1. Huruf kecil , contoh : lengkong
2. Huruf besar, contoh : LENGKONG
3. Huruf besar dilanjutkan huruf kecil, contoh : Lengkong
4. Angka , contoh : 1945
5. Karakter inputan sama, contoh : dan
6. Jumlah karakter sebelum karakter ditemukan sama

Tabel 3-5 Data Buku

Buku ke	**Judul buku**	**Jumlah karakter Deskripsi**
1	Azab dan Sengsara	468
2	Akademi militer dan peristiwa lengkong	1858
3	Hulubalang Raja	3208
4	Salah Asuhan	3900
5	De davinci code	8409

Tabel 3-6 hasil uji coba kata inputan huruf kecil

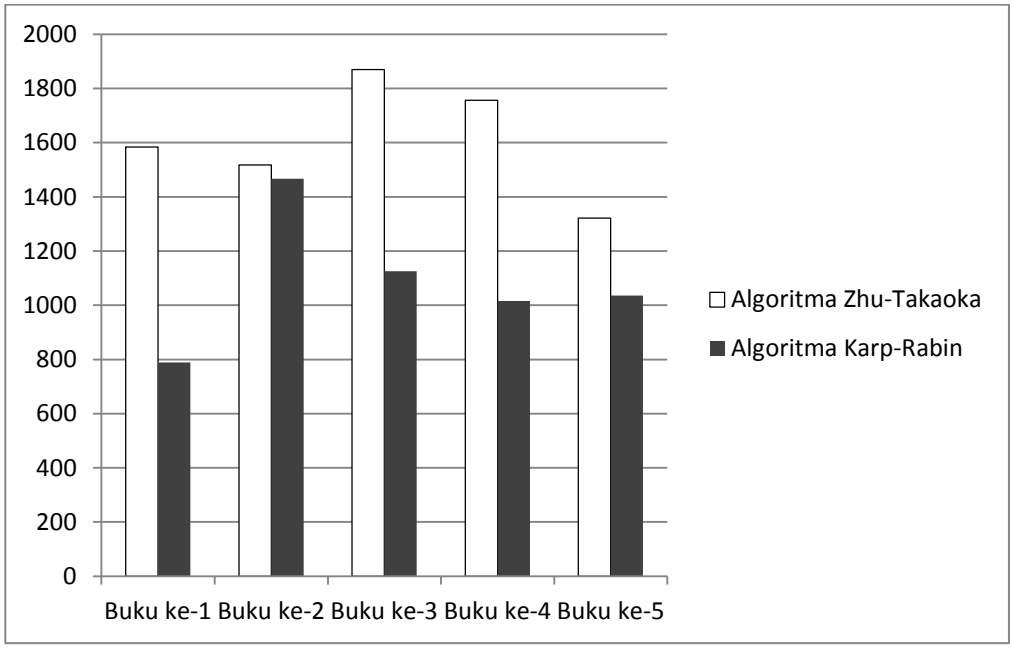
Buku ke	Kata yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
1	miskin	7	215	1	257	Data ditemukan	2090	268392	Data ditemukan	656	384
				2			1896			248	
				3			2737			240	
				4			1347			390	
				5			1245			265	
				Rata-Rata			**1663**			**307**	
2	pasukan	1	65	1	77	Data ditemukan	1592	271160	Data ditemukan	482	440
				2			1181			226	
				3			1322			449	
				4			1288			226	
				5			1498			272	
				Rata-Rata			**1376**			**331**	
3	sultan	9	502	1	540	Data ditemukan	3891	273848	Data ditemukan	282	384
				2			3044			309	
				3			4734			281	

Buku ke	Kata yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			Memori (byte)	Waktu (ms)	hasil	Memori (byte)	Waktu (ms)	hasil
				4			1854			270	
				5			1711			280	
				Rata-Rata			**3046**			**284**	
4	keluarga	18	321	1	397	275344	2126	Data ditemukan	488	1419	Data ditemukan
				2			2110			1353	
				3			1212			1577	
				4			1568			1370	
				5			1264			1362	
				Rata-Rata			**1656**			**1416**	
5	dekade	4	286	1	338	284584	2497	Data ditemukan	384	543	Data ditemukan
				2			1416			276	
				3			1720			283	
				4			1257			275	
				5			1242			956	
				Rata-Rata			**1626**			**466**	

Untuk perhitungan waktu computer dengan arsitektur yang berbeda akan berbeda pula lama waktu untuk setiap jenis operasinya hal ini disebabkan dalam system operasi tertentu untuk mengeksekusi perintah tertentu dilakukan

penjadwalan yang berbeda – beda. Untuk itu model abstrak pengukuran waktu atau ruang besaran yang dipakai adalah kompleksitas algoritma.

Dari hasil percobaan dengan inputan huruf kecil,untuk kecepatan dan efesiensi algoritma Karp-Rabin lebih unggul dari pada algoritma Zhu-Takaoka dan hal yang berpengaruh pada pencocokan yaitu posisi karakter ketika ditemukan dan karakteristik jenis huruf sebelum diketemukan . Sedangkan dari segi ketepatan kedua algoritma sama-sama dapat menyelesaikan pencocokan dan menghasilkan keluaran yang tepat.



Gambar 3.3 Grafik Waktu uji coba kedua Algoritma terhadap inputan huruf kecil(ms)

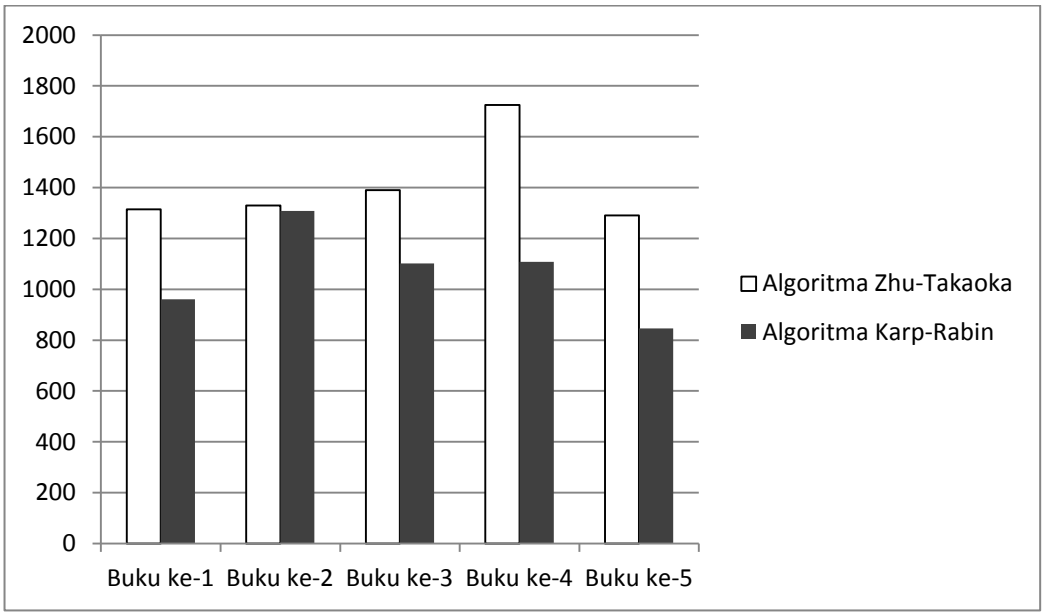
Tabel 3-7 hasil uji coba kata inputan huruf besar

Buku ke	Kata yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
1	SUTAN	9	228	1	277	Data ditemukan	1227	268384	Data ditemukan	654	336
				2			1060			252	
				3			939			242	
				4			1451			245	
				5			1046			255	
				Rata-Rata			**1144**			**329**	
2	TKR	52	919	1	1124	Data ditemukan	1279	271128	Data ditemukan	561	240
				2			1417			313	
				3			1338			537	
				4			1232			331	
				5			999			516	
				Rata-Rata			**1257**			**451**	
3	RAJA	106	2194	1	2725	Data ditemukan	3154	273840	Data ditemukan	929	288
				2			2273			545	
				3			1655			1029	
				4			1634			541	
				5			1961			523	
				Rata-Rata			**2135**			**713**	

Buku ke	Kata yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
4	BELANDA	16	327	1	400	Data ditemukan	1315	275216	Data ditemukan	544	240
				2			1591			309	
				3			1072			316	
				4			1221			304	
				5			1080			296	
				Rata-Rata			**1195**			**353**	
5	LANGDON	13	573	1	700	Data ditemukan	1901	284416	Data ditemukan	1049	440
				2			1177			350	
				3			1145			381	
				4			1073			336	
				5			1166			588	
				Rata-Rata			**1292**			**540**	

Pada percobaan dengan inputan huruf besar, dari segi waktu dan memori yang digunakan algoritma Karp-Rabin lebih cepat dan efesien dibandingkan dengan berpengaruh terhadap pencocokan yaitu karakteristik karakter sebelum ditemukan huruf capital dan huruf kecil berpengaruh terhadap waktu pencarian

algoritma Zhu-Takaoka. Hal yang Sedangkan dari segi ketepatan kedua algoritma sama-sama dapat menyelesaikan pencocokan dan menghasilkan keluaran yang tepat

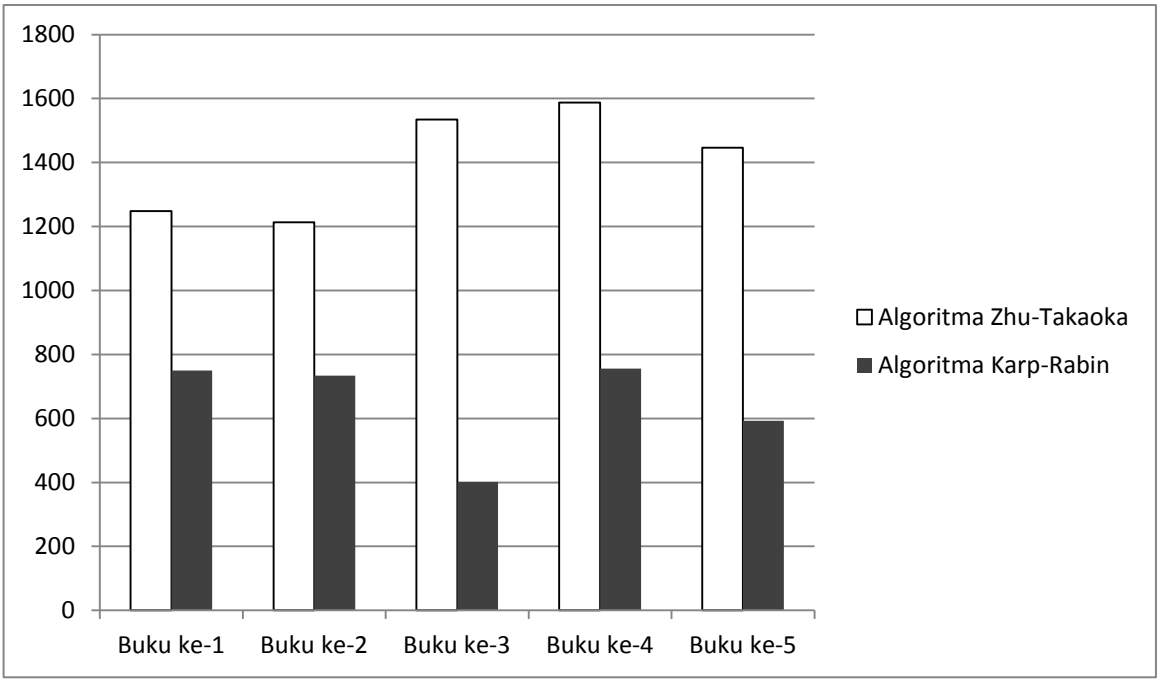


Gambar 3.4 Grafik Waktu uji coba kedua Algoritma terhadap inputan huruf besar

Tabel 3-8 Hasil uji coba inputan Huruf besar dilanjutkan huruf kecil

Buku ke	Kata yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
1	Ayah	7	221	1	263	Data ditemukan	1233	268360	Data ditemukan	507	288
				2			1133			234	
				3			1507			221	
				4			911			218	
				5			1016			242	
				Rata-Rata			**1160**			**284**	
2	Bogor	2	82	1	98	Data ditemukan	1123	271136	Data ditemukan	297	336
				2			1290			444	
				3			1308			391	
				4			1109			228	
				5			975			279	
				Rata-Rata			**1161**			**327**	
3	Inderapura	8	78	1	102	Data ditemukan	1297	273872	Data ditemukan	649	584
				2			1588			847	
				3			1189			393	
				4			1347			415	
				5			1322			368	
				Rata-Rata			**1348**			**534**	

Pada percobaan dengan inputan huruf besar dilanjutkan huruf kecil, dari segi waktu dan memori yang digunakan algoritma Karp-Rabin lebih cepat dan efesien dibandingkan dengan berpengaruh terhadap pencocokan yaitu karakteristik karakter sebelum ditemukan huruf capital dan huruf kecil berpengaruh terhadap waktu pencarian algoritma Zhu-Takaoka. Hal yang Sedangkan dari segi ketepatan kedua algoritma sama-sama dapat menyelesaikan pencocokan dan menghasilkan keluaran yang tepat.

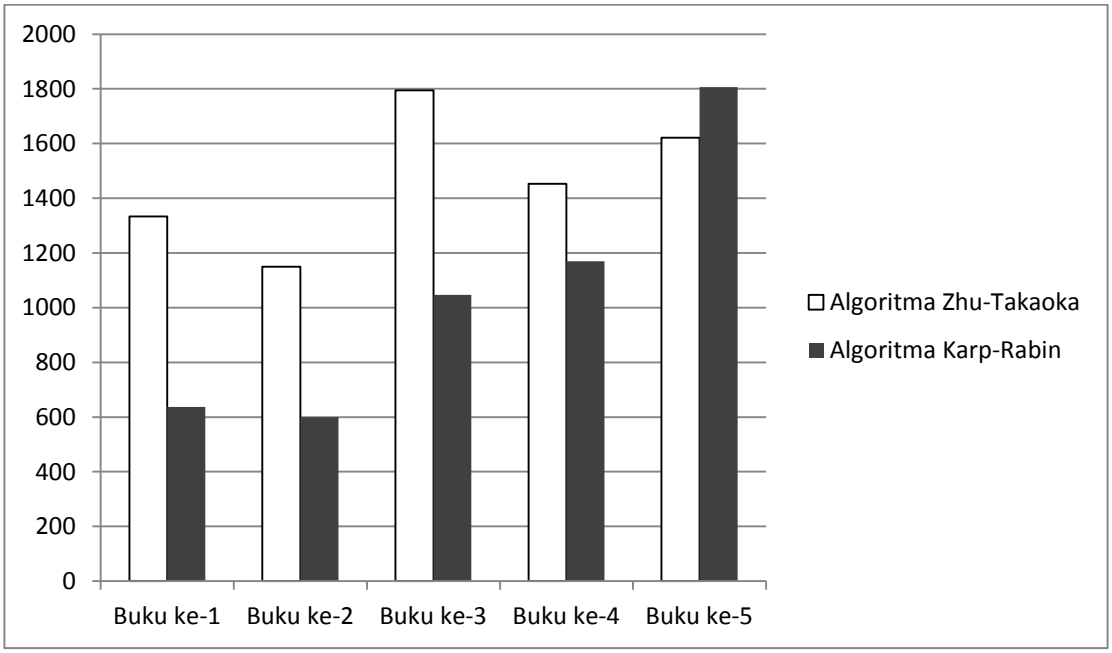


Gambar 3.5 Grafik Waktu uji coba kedua Algoritma terhadap inputan huruf Besar dilanjutkan huruf kecil

Tabel 3-9 Hasil uji coba inputan angka

Buku ke	Angka yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
1	50	15	251	1	311	Data ditemukan	1317	268360	Data ditemukan	2878	184
				2			1287			600	
				3			1106			491	
				4			989			434	
				5			945			628	
				Rata-Rata			**1128**			**1006**	
2	1945	43	814	1	925	Data ditemukan	1335	271136	Data ditemukan	859	288
				2			926			833	
				3			1430			376	
				4			1118			329	
				5			1196			555	
				Rata-Rata			**1201**			**656**	
3	2	54	959	1	1192	Data ditemukan	1358	273840	Data ditemukan	537	136
				2			1134			276	
				3			1106			522	
				4			1190			261	
				5			1116			415	

Buku ke	Angka yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
				Rata-Rata			**1180**			**402**	
4	3	45	832	1	1040	Data ditemukan	1109	275224	Data ditemukan	509	136
				2			1121			228	
				3			1157			242	
				4			1082			373	
				5			1217			255	
				Rata-Rata			**1148**			**321**	
5	1	35	1209	1	1476	Data ditemukan	1716	284376	Data ditemukan	516	136
				2			1451			463	
				3			1167			270	
				4			1784			322	
				5			1375			304	
				Rata-Rata			**1498**			**375**	



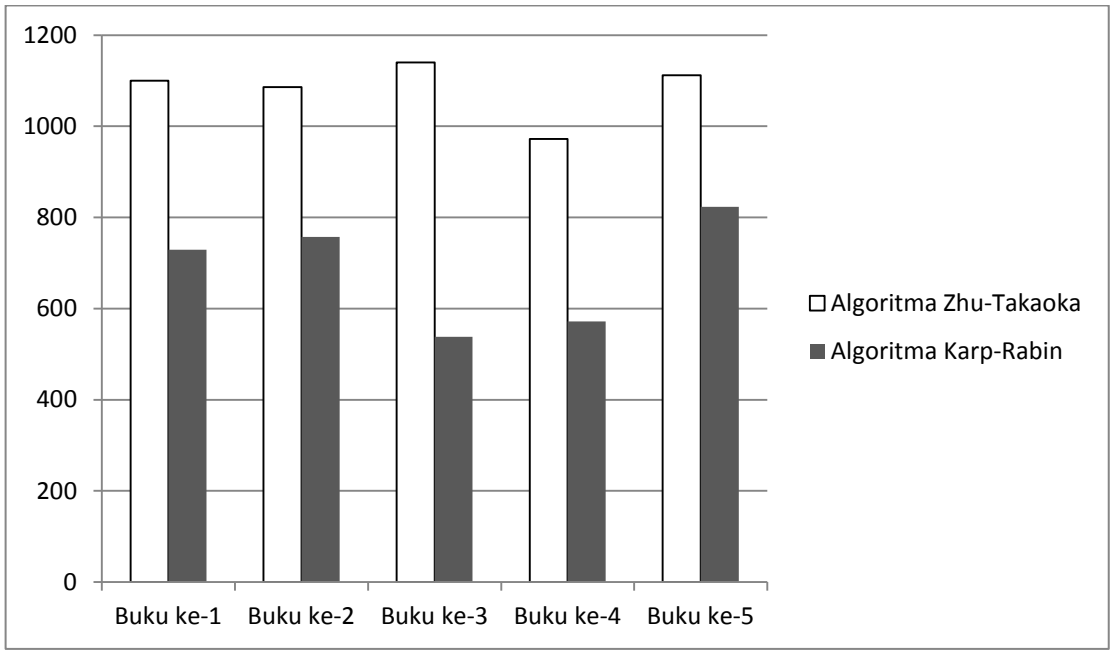
Gambar 3.6 Grafik Waktu uji coba kedua Algoritma terhadap inputan Angka

Pada percobaan dengan inputan angka, dari segi waktu dan memori yang digunakan algoritma Karp-Rabin lebih cepat dan efesien dibandingkan dengan berpengaruh terhadap pencocokan yaitu karakteristik karakter sebelum ditemukan huruf capital dan huruf kecil berpengaruh terhadap waktu pencarian algoritma Zhu-Takaoka. Hal yang Sedangkan dari segi ketepatan kedua algoritma sama-sama dapat menyelesaikan pencocokan dan menghasilkan keluaran yang tepat.

Tabel 3-10 hasil uji coba inputan sama

Buku ke	Angka yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
1	dan	3	73	1	87	Data ditemukan	1018	268352	Data ditemukan	581	240
				2			891			219	
				3			1539			217	
				4			1206			207	
				5			889			213	
				Rata-Rata			**1108**			**287**	
2	dan	4	107	1	129	Data ditemukan	1062	271128	Data ditemukan	241	240
				2			1244			219	
				3			1518			451	
				4			891			479	
				5			1288			490	
				Rata-Rata			**1200**			**376**	
3	dan	6	39	1	54	Data ditemukan	1034	273832	Data ditemukan	209	240
				2			1326			205	
				3			1056			199	
				4			980			266	
				5			971			207	
				Rata-Rata			**1073**			**217**	

Buku ke	Angka yang dicari	Karakteristik karakter sebelum kata ditemukan		Pengujian ke-	Panjang karakter sebelum kata ditemukan	Zhu-Takaoka			Karp-Rabin		
		Jumlah Huruf Besar	Jumlah Huruf Kecil			hasil	Waktu (ms)	Memori (byte)	hasil	Waktu (ms)	Memori (byte)
4	dan	20	376	1	457	Data ditemukan	1056	275216	Data ditemukan	266	240
				2			986			251	
				3			994			252	
				4			1243			273	
				5			970			261	
				Rata-Rata			**1049**			**260**	
5	dan	1	80	1	95	Data ditemukan	1417	284392	Data ditemukan	213	240
				2			1301			204	
				3			1109			206	
				4			1093			216	
				5			1086			230	
				Rata-Rata			**1201**			**213**	



Gambar 3.7 Grafik Waktu uji coba kedua Algoritma terhadap inputan sama

Dari hasil uji coba keseluruhan terhadap kedua algoritma ,keduanya memiliki ke akuratan yang sama-sama baik. Algoritma Karp-Rabin cendrung lebih cepat dan efesien dalam penggunaan ruang memori, hal yang mempengaruhi pencarian waktu dan penggunaan ruang adalah panjangnya karakter sebelum kata pencarian ditemukan dan karakteristik karakter sebelum kata pencarian ditemukan, huruf besar dan huruf kecil ikut dan dari hasil perhitungan kompleksitas menggunakan metode Big –O algoritma Karp-Rabin menghasilikan kompleksitas O(n) sedangkan untuk algoritma Zhu-Takaoka menghasilkan kompleksitas $O(n^2)$. Maka dapat ditarik kesimpulan algoritma Karp-Rabin cendrung lebih baik dari pada algoritma Karp-Rabin.

3.1.3. Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak yang akan dibangun berdasarkan kebutuhan pengguna pada jurnal dan artikel serta hasil observasi. Spesifikasi kebutuhan perangkat lunak akan dibagi kedalam dua bagian yaitu SKPL-F (Spesifikasi Kebutuhan Perangkat Lunak Fungsional) dan SKPL-NF (Spesifikasi kebutuhan perangkata lunak non-fungsional) Berikut ini adalah tabel Spesifikasi kebutuhan perangkat lunak pencocokan string :

Tabel 3-11 Spesifikasi Kebutuhan Perangkat Lunak Fungsional

Kode	Kebutuhan
SKPL-F001	Perangkat lunak dapat malakukan inputan string atau pola yang diinginkan oleh user
SKPL-F002	Perangkat lunak dapat memproses hasil inputan untuk kemudian dimulai pencocokan string atau pola.
SKPL-F003	Perangkat lunak dapat mengukur waktu eksekusi ketika memulai pencarian.
SKPL-F004	Perangkat lunak dapat menampilkan penggunaan ruang memori saat melakukan pencocokan string.
SKPL-F005	Perangkat lunak dapat menampilkan hasil dari pencocokan string.

Tabel 3-12 Spesifikasi kebutuhan perangakat lunak non-fungsional

Kode	Kebutuhan
SKPL-NF001	Pengguna atau user yang menggunakan perangkat lunak ini adalah user yang ingin melakukan pencarian buku.
SKPL-NF002	Perangkat lunak yang dibangun berbasis desktop
SKPL-NF003	Perangkat keras yang digunakan adalah komputer dengan spesifikasi minimal processor Intel Pentium 4 2.6 GHz , memori 1024 MB,keyboard,dan mouse.
SKPL-NF004	Bahasa pemograman yang digunakan adalah java

3.1.3.1. Analisis Perangkat Lunak

Rumah baca buku sunda menggunakan sistem operasi Windows 7 . Kebutuhan perangkat lunak dalam membangun dan menerapkan sistem yang akan dibuat di rumah baca buku sunda adalah sebagai berikut :

1. Sistem Operasi Windows XP
2. Software :
    a. NetBeans IDE 6.9.1
    b. Java Runtime Edition, sebagai *platform* untuk menjalankan sistem
    c. Java Development Kit versi 5 atau 6 , untuk kompilasi kode – kode program

Spesifikasi kebutuhan perangkat lunak tersebut dipilih karena kemudahannya, *familiar* dan *Interaktif* serta mudah dalam memahami cara kerjanya.

3.1.3.2. Analisis Perangkat pikir

Analisa dan spesifikasi kebutuhan diperlukan agar kemampuan perangkat lunak yang dibangun menjadi jelas. Beberapa analisa dan kebutuhan yang berkaitan dengan perangkat lunak yang akan dibangun nanti yaitu analisa dan kebutuhan pengguna. Adapun karakteristik pengguna pada perangkat lunak pencocokan string yang akan dibangun yaitu user sebagai berikut :

1. User dapat menggunakan komputer, minimal mampu menggunakan keyboard sebagai sarana penginputan data dan kata kunci pencarian.

2. User dapat membaca.
3. Jenjang pendidikan dimulai dari Anak sekolah (SD,SMP dan SMA), Mahasiswa

Berdasarkan analisis pada *user*, dapat diambil kesimpulan bahwa pengguna (*user)* yang ada cukup memenuhi syarat sebagai pengguna sistem yang akan dikembangkan, sehingga tidak diperlukan pelatihan khusus mengenai penggunaan komputer, cukup berupa dokumen atau buku panduan untuk membantu menjalankan perangkat lunak.

3.2. Perancangan Sistem

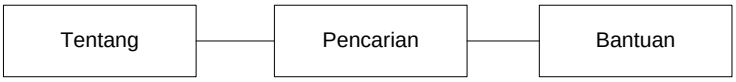
Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi mengkonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem. Adapun perancangan sistem dari sistem informasi kepegawaian yang dibuat dijelaskan sebagai berikut.

3.2.1. Perancangan Arsitektural Perangkat Lunak

Perancangan arsitektur adalah tahap yang dilakukan dalam merancang stuktur menu , merancang antarmuka bagi pengguna, perancangan pesan dalam perangkat lunak serta jaringan semantik.

3.2.2. Perancangan Struktur Menu

Perancangan struktur menu berisikan menu dan submenu yang berfungsi memudahkan pengguna didalam menggunakan sistem. Pada perangkat lunak ini perancangan struktur menu pada user menggunakan struktur menu hirarki atau sekuensial untuk jelasnya dapat dilihat pada Gambar 3.8 struktur menu pada User



Gambar 3.8 struktur menu pada User

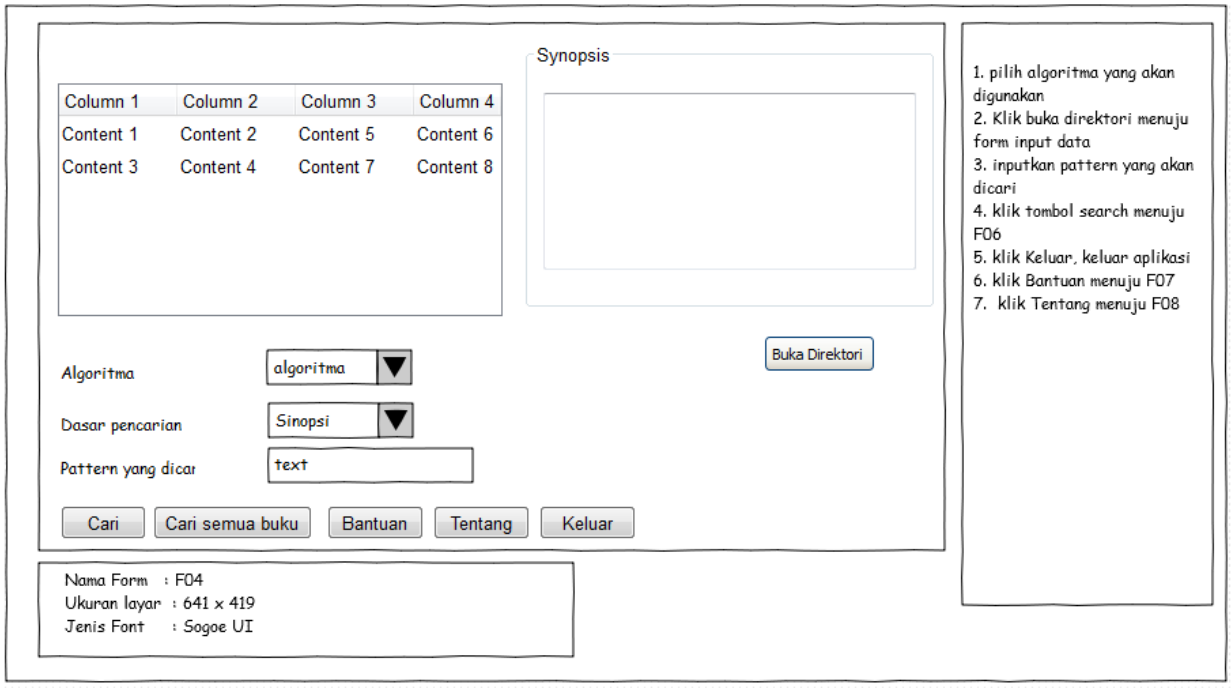
3.2.3. Perancangan Antarmuka Perangkat Lunak

Perancangan antarmuka merupakan sebuah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Adapun perancangan antarmuka perangkat lunak pencocokan string adalah sebagai berikut :

1. Desain Form pencarian

   Form pencarian merupakan form yang digunakan sebagai tampilan user pada saat akan mulai pencarian dimana algoritma di implementasikan. Desain tampilan form dan deskripsi objek dari aplikasi ini dapat dilihat pada Gambar 3.9 Desain tampilan form pencarian

   1. Tampilan Form Pencarian



Gambar 3.9 Desain tampilan form pencarian

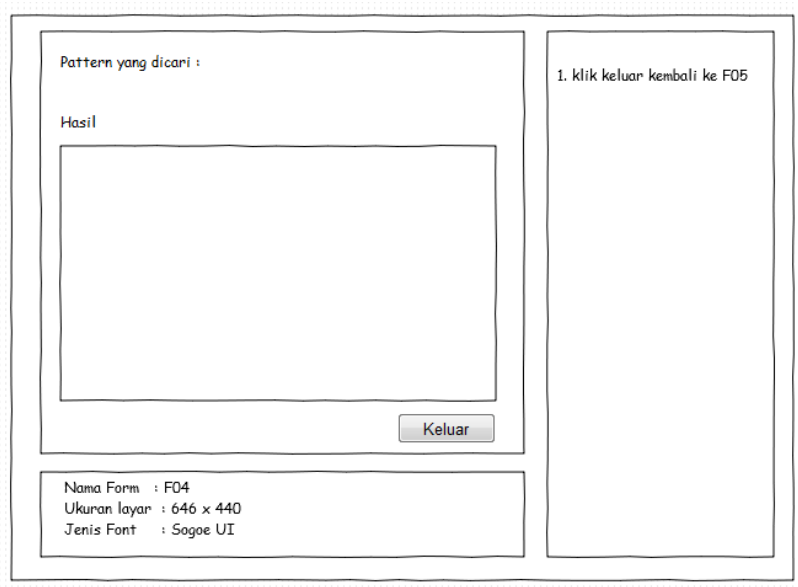
Tabel 3-13 Deskripsi objek form pencarian

Objek	Jenis	Keterangan
Column 1,Content 2	Tabel	Tempat menampilkan data buku yang telah tersedia dari data inputan
Algoritma	Combo Box	Memilih algoritma yang akan digunakan
Buka Direktori	Button	Menginputkan data buku yang akan menjadi sumber pencarian
Cari	Button	Memulai pencarian dari buku yang dipilih muncul F06
Cari Semua Buku	Button	Memulai pencarian dari semua buku yang terdapat di data base
Keluar	Button	Keluar aplikasi
Tentang	Button	Menuju F08, melihat tentang pembuat
Bantuan	Button	Menuju F07, melihat bantuan cara penggunaan

2. Desain Form proses pencarian

Form proses pencarian merupakan form yang digunakan sebagai tampilan dimana proses pencarian dikerjakan. Desain tampilan form dan deskripsi objek dari aplikasi ini dapat dilihat pada Gambar 3.10 Desain tampilan form proses pencarian

2. Tampilan Form Proses Pencarian



Gambar 3.10 Desain tampilan form proses pencarian

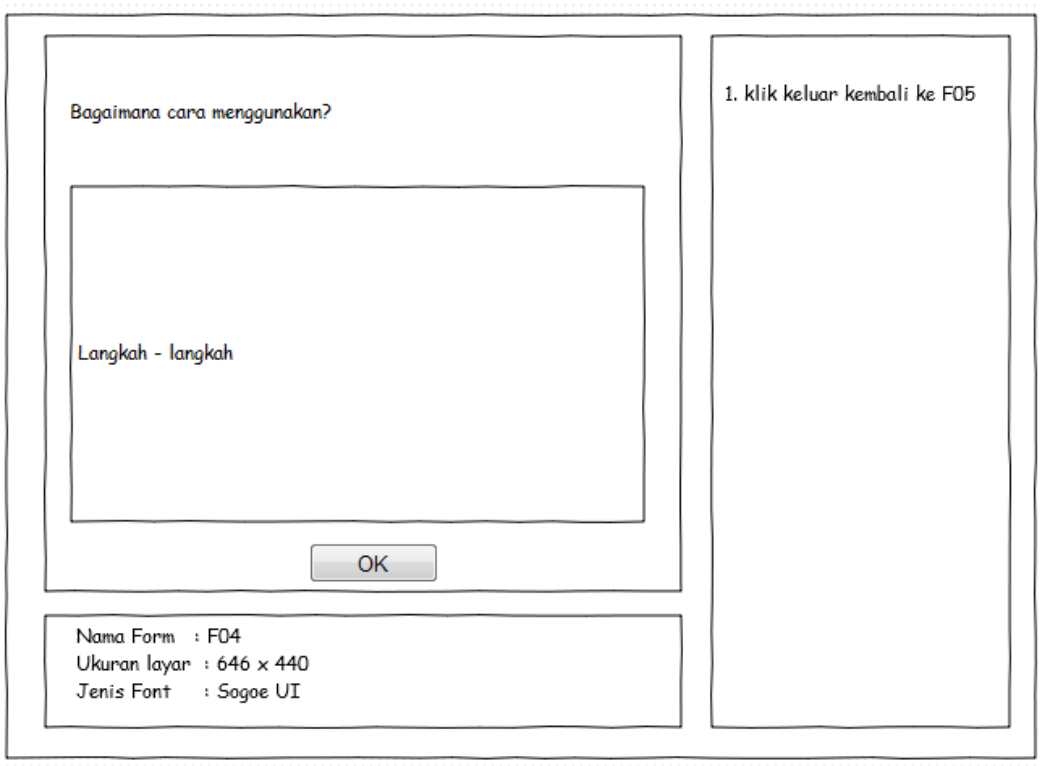
Tabel 3-14 Deskripsi objek form proses pencarian

Objek	Jenis	Keterangan
Pattern yang dicari	Text	Menampilkan pattern yang dicari
Hasil	Text area	Menampilkan hasil pencarian lengkap dengan waktu dan memori yang digunakan.
Keluar	Button	Keluar aplikasi

3. Desain Form bantuan

Form bantuan merupakan form yang menampilkan tatacara penggunaan perangkat lunak. Desain tampilan form dan deskripsi objek dari aplikasi ini dapat dilihat pada gambar dibawah.

3. Tampilan Form Bantuan



Gambar 3.11 Desain tampilan form bantuan

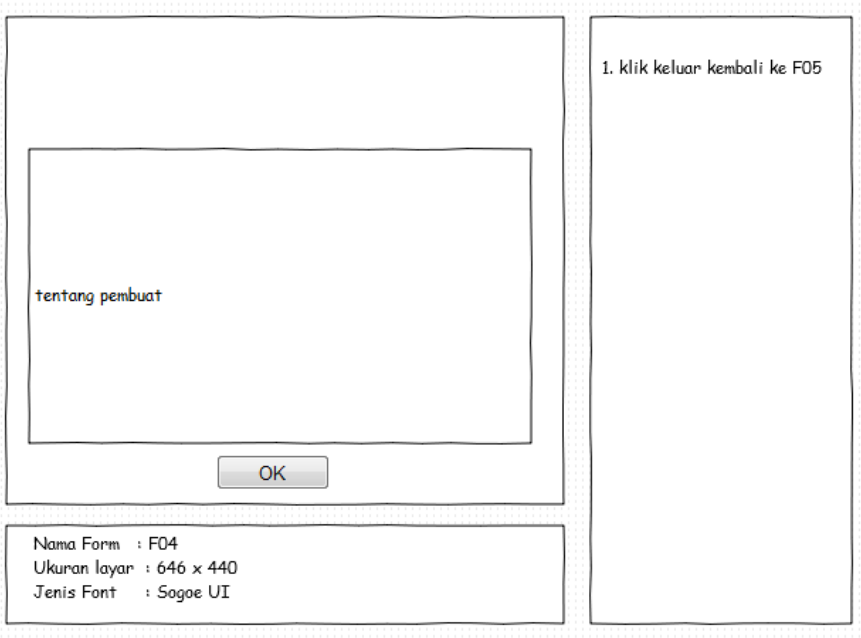
Tabel 3-65 Deskripsi objek form bantuan

Objek	Jenis	Keterangan
bagaimana	label	Menampilkan judul form
Langkah-langkah	Text area	Menampilkan langkah – langkah untuk menggunakan perangkat lunak
Keluar	Button	Keluar aplikasi

4. Desain Form tentang

Form bantuan merupakan form yang menampilkan tentang pembuat perangkat lunak. Desain tampilan form dan deskripsi objek dari aplikasi ini dapat dilihat pada gambar dibawah.

4. Tampilan Form tentang



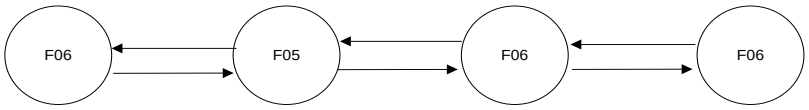
Gambar 3.12 Desain tampilan form tentang

Tabel 3-76 Deskripsi objek form tentang

Objek	Jenis	Keterangan
Langkah-langkah	Text area	Menampilkan langkah – langkah untuk menggunakan perangkat lunak
Keluar	Button	Keluar aplikasi

3.2.4. Perancangan Jaringan Semantik

Berikut ini adalah gambar jaringan semantik yang menggambarkan hubungan antar modul perangkat lunak pencocokan string untuk user :

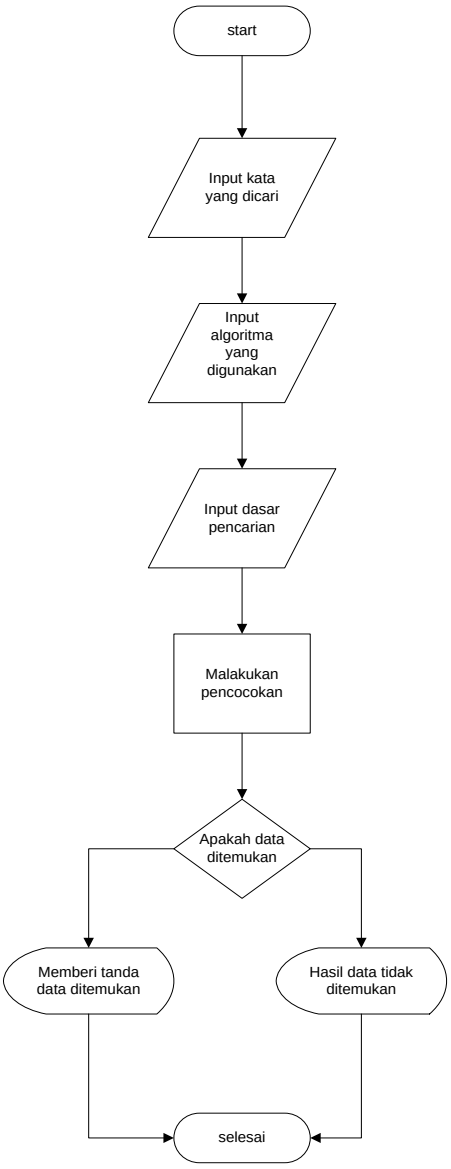


Gambar 3.13 jaringan semantik user

3.2.5. Perancangan Prosedural

Perancangan prosedural adalah tata cara atau urutan langkah – langkah untuk melakukan suatu proses. Perancangan prosedural menggambarkan algoritma dari prosedur - prosedur yang terdapat pada perangkat lunak yang dibangun. Adapun perancangan prosedural pada perangkat lunak pencocokan string adalah sebagai berikut :

1. Prosedure Uji Algoritma



Gambar 3.14 Prosedure pencarian kata