BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Masalah

Beberapa algoritma yang termasuk dalam metode *fingerprinting* yaitu Algoritma Rabin Karp, Algoritma Winnowing dan Algoritma Manber. Dalam penelitian ini akan dilakukan analisis performansi dari dua buah algoritma yang termasuk dalam metode *fingerprinting* yaitu Algoritma Winnowing dan Algoritma Manber yang dapat diterapkan untuk mendeteksi similaritas pada dokumen teks. Analisis performansi dilakukan dengan tujuan mendapatkan informasi dari kedua algoritma yaitu berupa informasi kecepatan, ketepatan, dan jumlah langkah dari Algoritma Winnowing dan Algoritma Manber dalam penerapannya pada sebuah sistem.

Dari sedikitnya penelitian dalam lingkup analisis performansi kedua algoritma ini maka menjadi dasar penelitian ini. Terdapat penelitian sebelumnya mengenai perbandingan pendekatan kedua algoritma ini yang berjudul "Perbandingan Pendekatan Deteksi Plagiarisme Dokumen Dalam Bahasa Inggris" oleh Ana Kurniawati dan I Wayan Simri Wicaksana dalam jurnalnya yang menganalisis perbandingan dan penerapannya pada dokumen berbahasa inggris. Berdasarkan hasil penelitian tersebut, disimpulkan bahwa algoritma winnowing lebih baik dari algoritma manber karena memberikan jaminan terdeteksinya dokumen sama [2]. Penerapan penelitian tersebut dilakukan pada sistem yang memproses teks berbahasa inggris. Berdasarkan hal tersebut maka akan dilakukan analisis yang nantinya akan diterapkan pada dokumen berbahasa Indonesia.

Penelitian lainnya dari kedua algoritma yaitu hanya berfokus pada implementasi algoritma sehingga informasi dari parameter-parameter yang digunakan tidak memiliki standar optimal untuk diterapkan pada rumus yang dipakai oleh kedua algoritma tersebut. Ketika ditemukan jumlah parameter optimal diharapkan dapat

mendapatkan akurasi yang lebih tepat dalam mendeteksi similaritas dari dokumen yang dibandingkan. Parameter yang perlu diuji yaitu jumlah gram, jumlah window, basis bilangan prima yang di gunakan dan proses pemilihan nilai hash yang berpengaruh pada persentasi kemiripan dari dokumen yang dibandingkan.

3.2 Analisis Input dan Output

Analisis Input dan Output sistem menjelaskan tentang bagaimana sistem menerima inputan dan menghasilkan output dari proses yang telah dilakukan. Berikut ini merupakan analisis input dan output dari sistem yang akan dibuat.

3.2.1 Analisis Input dan Output Algoritma Winnowing

Algoritma Winnowing memiliki input dan output data sebagai berikut.

3.2.1.1 Analisis Input

Pada penerapan Algoritma Winnowing dalam sebuah sistem membutuhkan beberapa inputan meliputi:

1. Dokumen Teks berektensi .txt
2. Jumlah Karakter Minimal 100 Huruf
3. Jumlah Gram = 5 , diambil berdasarkan jumlah penelitian sebelumnya [4].
4. Jumlah Window = 4, diambil berdasarkan jumlah penelitian sebelumnya [4].

3.2.1.2 Analisis Output

Output yang dihasilkan dari sistem berupa:

1. Persentasi Kemiripan dari dokumen yang dibandingkan
2. Informasi Waktu Proses

3.2.2 Analisis Input dan Output Algoritma Manber

Algoritma Manber memiliki input dan output data sebagai berikut:

3.2.2.1 Analisis Input

Pada penerapan Algoritma Manber dalam sebuah sistem membutuhkan beberapa inputan meliputi:

1. Dokumen Teks berektensi .txt
2. Jumlah Karakter Minimal 100 Huruf
3. Jumlah Gram = 5, diambil berdasarkan jumlah penelitian sebelumnya [2].
4. Jumlah Ukuran P = 4, diambil berdasarkan jumlah penelitian sebelumnya [2].

3.2.2.2 Analisis Output

Output yang dihasilkan dari sistem berupa:

1. Persentasi Kemiripan dari dokumen yang dibandingkan
2. Informasi Waktu Proses

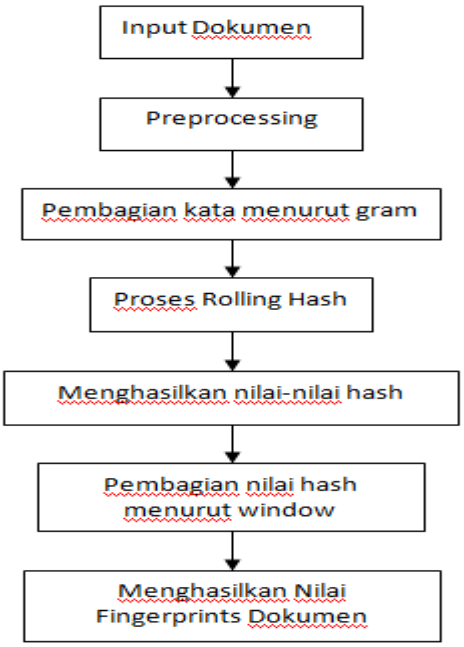
3.3 Analisis Algoritma

Analisis algoritma yaitu tahapan analisis algoritma apa saja yang akan dianalisis dalam penelitian ini yaitu :

1. Analisis Algoritma Winnowing
2. Analisis Algoritma Manber

3.3.1 Analisis Algoritma Winnowing

Tahapan pada Algoritma Winnowing dapat dilihat pada G*ambar 3.1*



Gambar 3.1 Langkah Algoritma Winnowing

Gambar 3.1 menjelaskan tahapan dari Algoritma Winnowing sampai pada proses menghasilkan nilai hash. Tahapan tersebut dilakukan pada dokumen pertama, setelah itu dilakukan tahapan yang sama untuk dokumen kedua sehingga menghasilkan dua nilai-nilai *fingerprints* dari kedua dokumen. Setelah itu proses dilanjutkan dengan perhitungan persentasi kemiripan dari kedua dokumen berdasarkan nilai *fingerprints* yang telah diperoleh. Persentasi kemiripan dihitung menggunakan metode *jaccard similarity coefficient.*

Input dari proses *document fingerprinting* adalah file teks. Kemudian *output*-nya berupa sekumpulan nilai *hash* yang disebut *fingerprint*. *Fingerprint* inilah yang akan dijadikan dasar pembanding antara *file-file* teks yang telah dimasukkan .

Algoritma Winnowing menghapus seluruh karakter-karakter yang tidak dibutuhkan seperti: tanda baca, spasi dan juga karakter lain, sehingga hanya karakter-

karakter yang berupa huruf yang akan diproses lebih lanjut . Berikut merupakan contoh dari alur proses Algoritma Winnowing:

1. Langkah 1:

Contoh kalimat : Hari Minggu sangat cerah!!!!.

Langkah awal pada Algoritma Winnowing yaitu melakukan *lowercase* pada setiap karakter dan membuang karakter-karakter yang tidak dibutuhkan seperti spasi dan symbol, sehingga teks menjadi bersih dari symbol-simbol selain huruf [a-z]. maka kalimat diatas akan menjadi

hariminggusangatcerah

2. Langkah 2:

Langkah berikutnya yaitu pembagian kata menurut gram yang diinputkan oleh pengguna. Sebagai contoh nilai *gram*=5. Maka teks akan terbagi menjadi :

harim	mingg	gusan	ngatc	cerah
arimi	inggu	usang	gatce	
rimin	nggus	sanga	atcer	
iming	ggusa	angat	tcera	

3. Langkah 3:

Langkah berikutnya setelah dibagi menurut gram yaitu dilakukan proses *Rolling Hash* untuk menghasilkan nilai hash dari setiap gram yang dibentuk. Proses *Rolling Hash* menggunakan persamaan (2.1) dan persamaan (2.2).

Contoh kata "**harim**"

Sesuai persamaan (2.1) nilai hash yang diperoleh yaitu:

h=104, **a**=97, **r**=114, **i**=105, **m**=109 (nilai ascii karakter)

$104*3^4 + 97*3^3 + 114*3^2 + 105*3 + 109$

8424 + 2619 + 1026 + 315 + 109 = 12493

Dengan menggunakan rumus tersebut maka dapat dihasilkan nilai-nilai hash dari proses pembentukan gram sebelumnya

12493, 12312, 13475, 12826, 13066, 12828, 13084, 12619, 12938, 13888, 13330, 12161, 13011, 12404, 12297, 13417, 12167

4. Langkah 4:

Setelah mendapatkan nilai *hash* dari proses *Rolling Hash* maka langkah berikutnya adalah membagi nilai-nilai *hash* tersebut menurut window, nilai window dinputkan oleh pengguna sebagai contoh nilai window = 4. Pemilihan window hampir sama seperti pembentukan gram. Maka akan didapatkan window sebagai berikut:

{ 12493 12312 13475 12826 }, { 12312 13475 12826 13066 }
{ 13475 12826 13066 12828 }, { 12826 13066 12828 13084 }
{ 13066 12828 13084 12619 }, { 12828 13084 12619 12938 }
{ 13084 12619 12938 13888 }, { 12619 12938 13888 13330 }
{ 12938 13888 13330 12161 }, { 13888 13330 12161 13011 }
{ 13330 12161 13011 12404 }, { 12161 13011 12404 12297 }
{ 13011 12404 12297 13417 }, { 12404 12297 13417 12167 }

5. Langkah 5:

Langkah berikutnya yaitu memilih nilai terkecil dari setiap window untuk dijadikan *fingerprints*.

{ 12493 **12312** 13475 12826 }, { 12312 13475 12826 13066 }
{ 13475 **12826** 13066 12828 }, { 12826 13066 12828 13084 }
{ 13066 12828 13084 **12619** }, { 12828 13084 12619 12938 }
{ 13084 12619 12938 13888 }, { 12619 12938 13888 13330 }
{ 12938 13888 13330 **12161** }, { 13888 13330 12161 13011 }
{ 13330 12161 13011 12404 }, { 12161 13011 12404 12297 }
{ 13011 12404 **12297** 13417 }, { 12404 12297 13417 **12167** }

Dari hasil pemilihan nilai terkecil dari setiap window maka didapatkan nilai *fingerprints* yaitu

12312, 12826, 12619, 12161, 12297,12167

Setelah itu dilakukan penambahan posisi dari fingerprints menjadi:

[12312, 1], [12826, 3], [12619, 8], [12161, 12], [12297, 15], [12167, 17]

6. Langkah 6:

Langkah berikutnya yaitu perhitungan persentasi kemiripan menggunakan *jaccard's similarity coefficient* dengan rumus (2.3)

Sebagai contoh untuk mengetahui persentasi kesamaan dari dua dokumen sebagai berikut:

a. Dokumen A berisi kalimat : Hari minggu sangat cerah !!!
b. Dokumen B berisi kalimat : Hari minggu mendung

Menggunakan proses *Rolling hash* dan pemilihan window didapat hasil *fingerprints*

Dokumen A : 12312, 12826, 12619, 12161, 12297,12167

Dokumen B : 12312, 12826, 12605, 12512

Dokumen B menghasilkan nilai *hash* dengan dilakukan proses yang sama seperti proses Dokumen A sehingga didapatkan hasil *fingerprints* 12312, 12826, 12605, 12512 . Berdasarkan rumus *jaccard similarity coefficient* maka dapat dilakukan perhitungan berikut ini berdasarkan hasil dari *fingerprints* Dokumen A dan B:

$|A \cap B|$ = 12312, 12826

$|A \cup B|$ = 12312, 12826, 12619, 12161, 12297,12167, 12605, 12512

$D(A, B)$ = (2/8)*100 = 25%

Jadi total persentasi kemiripan dari Dokumen A dan B yaitu sebesar 25%

3.3.1.1 Analisis Kompleksitas Algoritma Winnowing

Perhitungan kompleksitas waktu asimptotik Algoritma Winnowing dapat dihitung dengan menelusuri setiap langkah algoritma winnowing *pada pseudocode* pada tabel 3.1

Table 3.1 Pseudocode Winnowing

No	Algoritma	Jumlah Eksekusi
1	int: gram	1
2	array: tampung	1
3	**for** (length = 0; length < value.Length; length++) **do**	n
4	**if** (length == gram) **then**	1
5	**for** (start = 0; start <= value.Length - length; start++) **do**	n x n
7	substring = value.Substring(start, length)	n x n
8	tampung.Add(substring)	1
9	**endfor**	
10	**endif**	
11	**endfor**	
12	{Didapatkan hasil proses pertama berupa pembagian kata	

	menurut gram yang diinputkan}	
13	{Proses selanjutnya perhitungan nilai hash}	
14	array: tampungnilaihash	1
15	**for** (k = 0; k < tampung.Count; k++) **do**	n
16	double : hitung	1
17	Jumlah=0	1
18	d = tampung[k]	n
19	**for** (l = 0; l < gram; l++) **do**	n x n
20	s = d.Substring(l, 1)	n
21	w = s	1
22	asciiCodee = (int)w	1
23	hitung = asciiCodee * (Math.Pow(3, (gram - (l + 1))))	n x n
24	jumlah = jumlah + hitung;	1
25	**endfor**	
26	tampungnilaihash.Add(jumlah)	1
27	**endfor**	
28	{Mendapatkan hasil hashing dari setiap gram}	
29	{tahap selanjutnya pembagian window dari hasil hashing}	
30	array : tampungwindow	1
31	**for** (m = 0; m < (tampungnilaihash.Count) - (window - 1); m++) **do**	n
32	max = 100000	1
33	gabungin = “”	1
34	**for** (n = 0; n < window; n++) **do**	n x n
35	bts2 = n + m	1
36	window1 = tampungnilaihash[bts2]	n
37	min = tampungnilaihash[bts2]	n

38	**if** (min < max) **then**	1
39	max = min	1
40	**endif**	
41	tampungwindow.Add(window1)	1
42	**endfor**	
43	tampungfingerprints.Add(max)	1
44	**endfor**	

Algoritma Winnowing terdiri dari banyak perulangan *for* atau lebih dikenal sebagai *nested for loop.* Perhitungan jumlah T(*n*) pada *nested for loop* memiliki dua aturan, yaitu :

1. Dianalisis dari *loop* terdalam kemudian keluar.
2. Waktu eksekusi total sebuah *statement* adalah waktu eksekusi *statement* tersebut dikalikan hasil kali dari banyaknya iterasi semua *loop* yang ada di luarnya.

Dari perhitungan pada tabel 3.1 didapatkan hasil kompleksitas waktu T(n) = $5n^2$+ 7n + 18 dan termasuk pada kelompok algoritma kuadratik sesuai dengan aturan *nested for loop* dengan kompleksitas waktu asimptotik yaitu T(n) = n^2.

Perhitungan waktu yang dibutuhkan sesuai kerja mesin yaitu jika misalkan mesin dapat memproses 1 kali eksekusi membutuhkan waktu 10^{-6} detik, maka untuk memasukan data jika T(n) = 1000, maka T(n) = (n-1) menjadi T(1000) = (1000 – 1) x 10^{-6} = 0,000999 detik untuk n=1000.

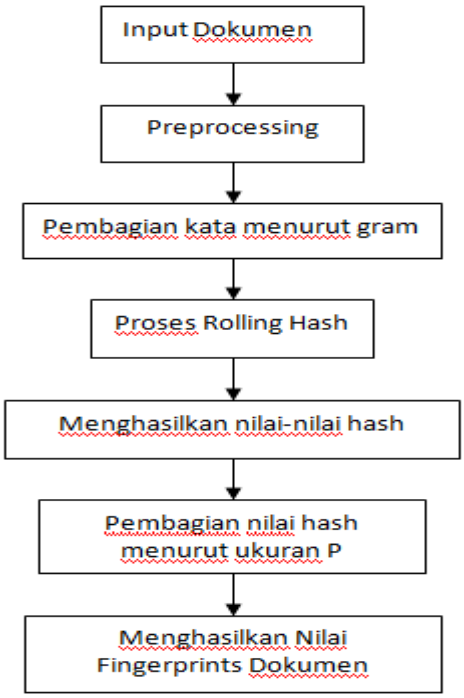
Tabel 3.2 Pertumbuhan n terhadap T(n) Winnowing

n	T(n) = $5n^2$ + 7n + 18	n^2
10	750	100
100	52.500	10.000
1000	5.025.000	1.000.000
10.000	500.250.000	100.000.000

Dari tabel 3.2 dapat dikatakan bahwa T(n) tumbuh sama seperti n^2 ketika n bertambah.

3.3.2 Analisis Algoritma Manber

Tahapan pada Algoritma Manber hampir sama dengan Algoritma Winnowing yang membedakan adalah proses pemilihan *fingerprints* dari hasil perhitungan *Rolling Hash*. Contoh tahapan Algoritma Manber dapat dilihat pada gambar 3.2.



Gambar 3.2 Tahapan Algoritma Manber

Gambar 3.2 menjelaskan tahapan dari Algoritma Manber sampai pada proses menghasilkan nilai hash. Tahapan tersebut dilakukan pada dokumen pertama, setelah itu dilakukan tahapan yang sama untuk dokumen kedua sehingga menghasilkan dua nilai-nilai fingerprints dari kedua dokumen. Setelah itu proses dilanjutkan dengan perhitungan persentasi kemiripan dari kedua dokumen berdasarkan nilai *fingerprints*

yang telah diperoleh. Persentasi kemiripan dihitung menggunakan metode *jaccard similarity coefficient.* Berikut ini merupakan contoh dari tahapan Algoritma Manber:

1. Langkah 1:

Contoh kalimat : Hari Minggu sangat cerah!!!!.

Langkah awal pada Algoritma Winnowing yaitu melakukan lowercase pada setiap karakter dan membuang karakter-karakter yang tidak dibutuhkan seperti spasi dan symbol, sehingga teks hanya menjadi bersih dari symbol-simbol selain huruf [a-z]. maka kalimat diatas akan menjadi

hariminggusangatcerah

2. Langkah 2:

Langkah berikutnya yaitu pembagian kata menurut gram yang diinputkan oleh pengguna. Sebagai contoh nilai gram=5. Maka teks akan terbagi menjadi :

harim	mingg	gusan	ngatc	cerah
arimi	inggu	usang	gatce	
rimin	nggus	sanga	atcer	
iming	ggusa	angat	tcera	

3. Langkah 3:

Langkah berikutnya setelah dibagi menurut gram yaitu dilakukan proses Rolling Hash untuk menghasilkan nilai hash dari setiap gram yang dibentuk. Proses Rolling Hash menggunakan persamaan (2.1) dan persamaan (2.2).

Contoh kata "**harim**"

Sesuai persamaan (2.1) nilai hash yang diperoleh yaitu:

h=104, **a**=97, **r**=114, **i**=105, **m**=109 (nilai ascii karakter)

$104*3^4 + 97*3^3 + 114*3^2 + 105*3 + 109$

8424 + 2619 + 1026 + 315 + 109 = 12493

Dengan menggunakan rumus tersebut maka dapat dihasilkan nilai-nilai *hash* dari proses pembentukan gram sebelumnya

12493, 12312, 13475, 12826, 13066, 12828, 13084, 12619, 12938, 13888, 13330, 12161, 13011, 12404, 12297, 13417, 12167

4. Langkah 4:

Langkah berikutnya pemilihan nilai *hash* yang memenuhi kriteria H *mod* P=0, ukuran P merupakan inputan pengguna. Sebagai contoh nilai P=5 sehingga dari hasil nilai hash sebelumnya dapat diambil nilai fingerprints yang memenuhi kriteria H *mod* P=0 yaitu:

13475, 13330

5. Langkah 5:

Langkah berikutnya yaitu perhitungan persentasi kemiripan menggunakan *jaccard's similarity coefficient* dengan rumus:

$$D(A,B) = \frac{|A \cap B|}{|A \cup B|} \times 100$$

Sebagai contoh untuk mengetahui persentasi kesamaan dari dua dokumen sebagai berikut:

Dokumen A berisi kalimat : Hari minggu sangat cerah !!!

Dokumen B berisi kalimat : Hari minggu mendung

Dari hasil proses didapat hasil *fingerprints* dari kedua dokumen yaitu:

Dokumen A : 13475, 13330

Dokumen B : 13475,12605

Dokumen B menghasilkan nilai hash dengan dilakukan proses yang sama seperti proses Dokumen A sehingga didapatkan hasil *fingerprints* 12312, 12826, 12605, 12512 . Berdasarkan rumus *jaccard similarity coefficient* sebelumnya maka dapat dilakukan perhitungan berikut ini berdasarkan hasil dari *fingerprints* Dokumen A dan B:

$|A \cap B|$ = 13475

$|A \cup B|$ = 13475,13330,12605

$D(A, B)$ = (1/3)*100 = 33,33%

Jadi total persentasi kemiripan dari Dokumen A dan B yaitu sebesar 33,33%

3.3.2.1 Analisis Kompleksitas Manber

Perhitungan kompleksitas waktu asimptotik Algoritma Manber dapat dihitung dengan menelusuri setiap langkah algoritma winnowing pada *pseudocode* metode ini.

Tabel 3.3 Pseudocode Manber

No	Algoritma	Jumlah Eksekusi
1	int: gram	1
2	array: tampung	1
3	**for** (length = 0; length < value.Length; length++) **do**	n
4	**if** (length == gram) **then**	1
5	**for** (start = 0; start <= value.Length - length; start++) **do**	n x n
7	substring = value.Substring(start, length)	n x n
8	tampung.Add(substring)	1
9	**endfor**	
10	**endif**	

11	**endfor**	
12	{Didapatkan hasil proses pertama berupa pembagian kata menurut gram yang diinputkan}	
13	{Proses selanjutnya perhitungan nilai hash}	
14	array: tampungnilaihash	1
15	**for** (k = 0; k < tampung.Count; k++) **do**	n
16	double : hitung	1
17	Jumlah=0	1
18	d = tampung[k]	n
19	**for** (l = 0; l < gram; l++) **do**	n x n
20	s = d.Substring(l, 1)	n
21	w = s	1
22	asciiCodee = (int)w	1
23	hitung = asciiCodee * (Math.Pow(3, (gram - (l + 1))))	n x n
24	jumlah = jumlah + hitung;	1
25	**endfor**	
26	tampungnilaihash.Add(jumlah)	1
27	**endfor**	
28	{Mendapatkan hasil hashing dari setiap gram}	
29	{tahap selanjutnya pemilihan nilai *fingerprints*}	
30	array: tampungfingerprints	1
31	**for** (n = 0; n < tampungnilaihash.Count; n++) **do**	n
32	mod = tampungnilaihash[n] % window	n
33	if(mod == 0)	1
34	tampungfingerprints.Add(tampungnilaihash[n])	n
35	**endif**	
36	**endfor**	

Pseudocode Manber terdiri dari banyak perulangan *for* atau lebih dikenal sebagai *nested for loop*. Perhitungan jumlah T(n) pada *nested for loop* memiliki dua aturan, yaitu :

1. Dianalisis dari *loop* terdalam kemudian keluar.
2. Waktu eksekusi total sebuah *statement* adalah waktu eksekusi *statement* tersebut dikalikan hasil kali dari banyaknya iterasi semua *loop* yang ada di luarnya

Dari perhitungan pada tabel didapatkan hasil kompleksitas waktu T(n) = $4n^2$+ 7n + 13 dan termasuk pada kelompok algoritma kuadratik sesuai dengan aturan *nested for loop* dengan kompleksitas waktu asimptotik yaitu T(n) = n^2.

Perhitungan waktu yang dibutuhkan sesuai kerja mesin yaitu jika misalkan mesin dapat memproses 1 kali eksekusi membutuhkan waktu 10^{-6} detik, maka untuk memasukan data jika T(n) = 1000, maka T(n) = (n-1) menjadi T(1000) = (1000 – 1) x 10^{-6} = 0,000999 detik untuk n=1000.

Tabel 3.4 Pertumbuhan n terhadap T(n) Manber

n	T(n) = $4n^2$ + 7n + 13	n^2
10	600	100
100	42.000	10.000
1000	4.020.000	1.000.000
10.000	400.200.000	100.000.000

Dari tabel 3.4 dapat dikatakan bahwa T(n) tumbuh sama seperti n^2 ketika n bertambah. Berdasarkan analisis kompleksitas yang telah dilakukan, kedua algoritma termasuk kedalam algoritma yang sama.

3.4 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dilakukan bertujuan untuk mensimulasikan kedua algoritma kedalam sebuah perangkat lunak. Analisis kebutuhan perangkat lunak terbagi menjadi dua bagian yaitu:

1. Analisis Kebutuhan Non-Fungsional
2. Analisis Kebutuhan Fungsional

3.4.1 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional adalah langkah dimana seorang pembangun perangkat lunak (*software developer*) menganalisis sumber daya yang akan digunakan dan menggunakan perangkat lunak yang dibangun. Perangkat keras dan perangkat lunak yang dimiliki harus sesuai dengan kebutuhan sehingga dapat ditentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada.

Analisis kebutuhan non fungsional yang dilakukan dibagi dalam tiga tahap, yaitu:

1. Analisis perangkat keras (*hardware*).
2. Analisis perangkat lunak (*software*).
3. Analisis pengguna *(user*)

3.4.1.1 Analisis Perangkat Keras

Spesifikasi perangkat keras minimum yang butuhkan untuk menjalankan sistem Deteksi similaritas dokumen dapat dilihat pada tabel 3.5.

Tabel 3.5 Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Prosesor	Kecepatan Minimal 1.8 GHz
2	Monitor	Monitor 14.1’’
3	Memori	1024 MB

3.4.1.2 Analisis Perangkat Lunak

Komponen perangkat lunak pendukung yang dibutuhkan untuk menjalankan sistem Deteksi Similaritas Dokumen ini, dapat dilihat pada table 3.6

Tabel 3.6 Spesifikasi Perangkat Lunak

No	Perangkat Lunak
1	Sistem operasi Windows XP/ 7
2	Visual Studio 2010
3	Notepad++
4	Wampserver
5	Mysql

3.4.1.3 Analisis Perangkat Pikir

Analisis kebutuhan perangkat pikir merupakan uraian mengenai siapa saja yang akan menggunakan sistem dan terlibat dalam pengolahan data beserta karakteristiknya sehingga dapat diketahui tingkat pengalaman dan pemahaman pengguna terhadap sistem. Adapun karakteristik dari pengguna adalah sebagai berikut:

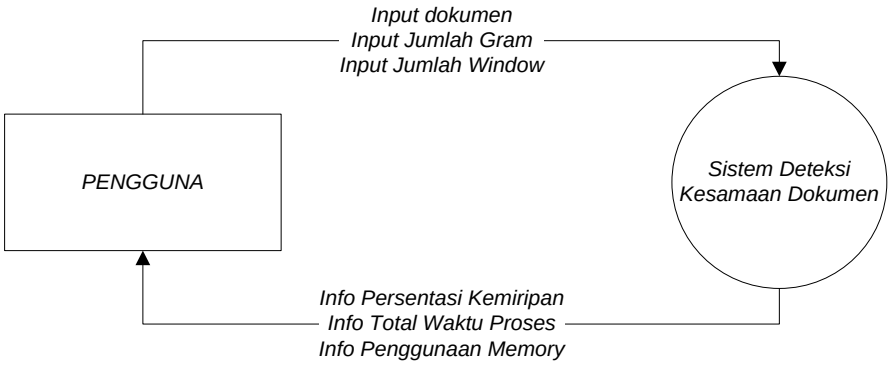
1. Pengguna dapat membaca tulisan.
2. Pengguna minimal mampu menggunakan *keyboard* dan *mouse* sebagai untuk berinteraksi dengan sistem.
3. Pengguna sudah terbiasa menggunakan komputer.

3.4.2 Analisis Kebutuhan Fungsional

Spesifikasi kebutuhan fungsional adalah spesifikasi tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran.

3.4.2.1 Diagram Konteks

Diagram konteks adalah diagram yang menggambarkan *input* , *process* dan *output* secara umum yang terjadi pada sistem perangkat lunak yang akan di bangun . Berikut Diagram konteks dari Sistem Deteksi Kesamaan Dokumen:



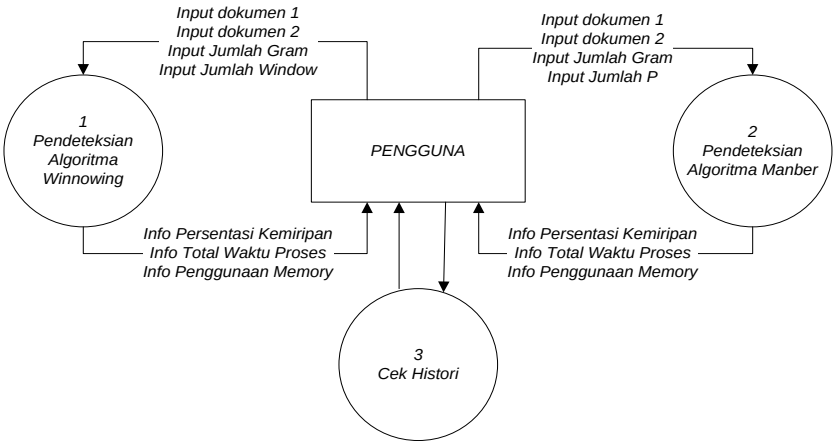
Gambar 3.3 Diagram Konteks

3.4.2.2 Data Flow Diagram

Data Flow Diagram merupakan suatu media yang digunakan untuk menggambarkan aliran data yang mengalir pada suatu sistem informasi. Dalam *Data Flow Diagram* (DFD) terdiri dari entitas luar, aliran data, proses, dan penyimpanan data.

3.4.2.2.1 Data Flow Diagram Level 1

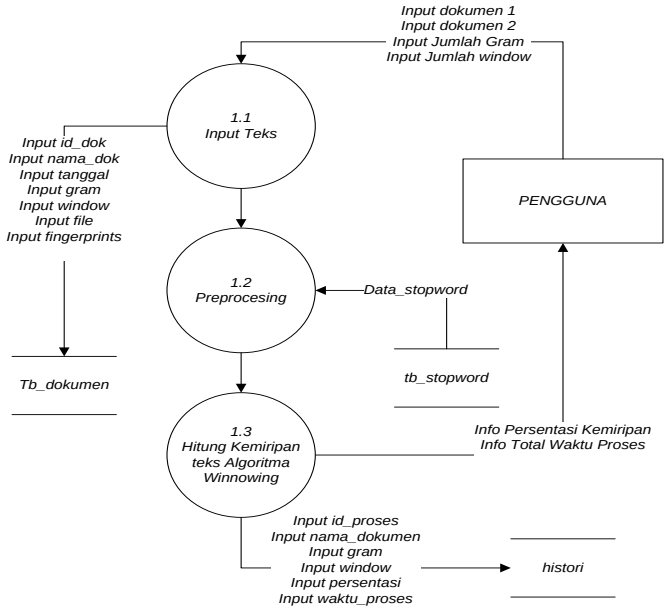
Berikut ini adalah Data Flow Diagram (DFD) Level 1 dari Sistem Deteksi Kesamaan Dokumen ditunjukan pada gambar 3.4.



Gambar 3.4 DFD Level 1

3.4.2.2.2 Data Flow Diagram Level 2 Pendeteksian Algoritma Winnowing

Berikut ini adalah Data Flow Diagram (DFD) Level 2 proses pendeteksian algoritma winnowing dari Sistem Deteksi Kesamaan Dokumen ditunjukan pada gambar 3.5.

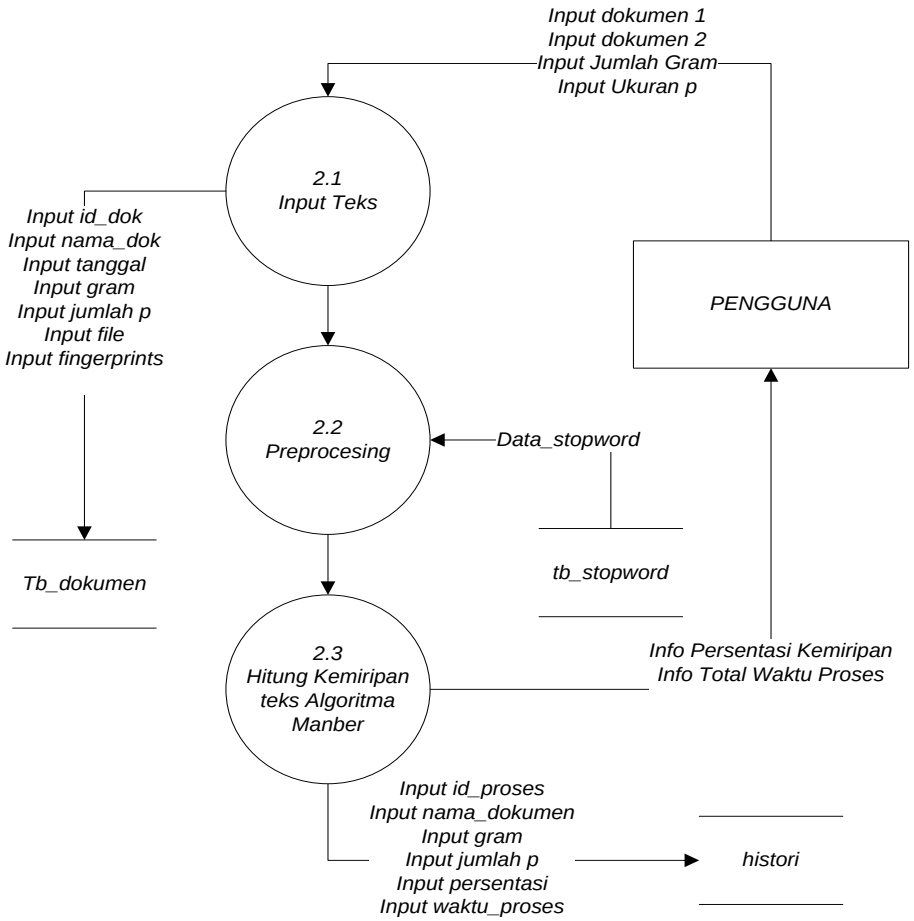


Gambar 3.5 DFD Level 2 Proses Pendeteksian Winnowing

Ket : tb_dokumen dan histori merupakan tabel yang terdapat dalam database data_dokumen

3.4.2.2.3 Data Flow Diagram Level 2 Pendeteksian Algoritma Manber

Berikut ini adalah Data Flow Diagram (DFD) Level 2 proses pendeteksian algoritma manber dari Sistem Deteksi Kesamaan Dokumen ditunjukan pada gambar 3.6.

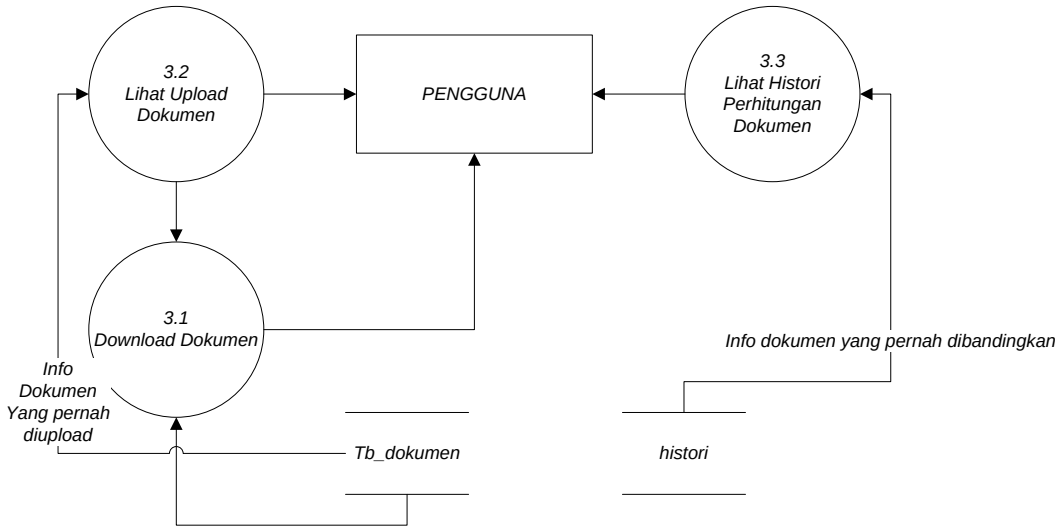


Gambar 3.6 DFD Level 2 Proses Pendeteksian Manber

Ket : tb_dokumen dan histori merupakan tabel yang terdapat dalam database data_dokumen

3.4.2.2.4 Data Flow Diagram Level 2 Proses Cek Histori

Berikut ini adalah Data Flow Diagram (DFD) Level 2 proses cek histori dari Sistem Deteksi Kesamaan Dokumen ditunjukan pada gambar 3.7.



Gambar 3.7 DFD Level 2 Proses Cek Histori

Ket : tb_dokumen dan histori merupakan tabel yang terdapat dalam database data_dokumen

3.4.2.3 Spesifikasi Proses

Spesifikasi proses merupakan deskripsi dari setiap elemen proses yang terdapat dalam program, yang meliputi nama proses, input, output dan keterangan dari proses. Berikut ini spesifikasi proses dari Sistem Deteksi Kesamaan Dokumen ditunjukan pada tabel berikut ini

3.4.2.3.1 Spesifikasi Proses Input Teks

Spesifikasi Proses pada proses 1.1 Input Teks dapat dilihat pada table 3.7.

Tabel 3.7 Spesifikasi Proses Input Teks

No	Proses	Keterangan

1	No Proses	1.1
	Nama	Input Teks
	Source	Pengguna
	Input	Input Dokumen 1, Input Dokumen 2, Input Jumlah Gram, Input Jumlah Window
	Output	Info Persentasi Kemiripan Info Total Waktu Proses
	Logika Proses	1. Pengguna Melakukan input dokumen teks atau input manual teks pada form yang disediakan sistem 2. Pengguna melakukan input jumlah gram dan window 3. Sistem menampung inputan dan menyimpan info ke tabel tb_dokumen dalam database data_dokumen

3.4.2.3.2 Spesifikasi Proses Preprocessig

Spesifikasi Proses pada proses 1.2 *Preprocessing* dapat dilihat pada tabel 3.8 berikut:

Tabel 3.8 Spesifikasi Proses Preprocessing

No	**Proses**	**Keterangan**
1	No Proses	1.2
	Nama	Preprocessing
	Source	1.1 Input Teks
	Input	Teks 1 , Teks 2
	Output	String yang sudah dilakukan proses penghapusan kata stopword dalam dokumen
	Logika Proses	1. Sistem menampung inputan teks 1 dan teks 2

		2. Sistem menghapus kata-kata stopword yang bersumber dari database tabel tb_stopword 3. Sistem memberikan output berupa string yang bebas dari kata-kata stopword

3.4.2.3.3 Spesifikasi Proses Hitung Kemiripan

Spesifikasi Proses pada proses 1.3 Hitung Kemiripan Teks Algoritma Winnowing dapat dilihat pada table 3.9.

Tabel 3.9 Spesifikasi Proses Hitung Kemiripan

No	Proses	Keterangan
1	No Proses	1.3
	Nama	Hitung Kemiripan Teks Algoritma Winnowing
	Source	1.1 Input Teks
	Input	Dokumen 1, Dokumen 2, Jumlah Gram, Jumlah Window
	Output	Info Persentasi Kemiripan Info Total Waktu Proses Info Penggunaan Memory
	Logika Proses	1. Pengguna Menginput dokumen teks atau input manual teks pada form yang disediakan sistem 2. Sistem menampung inputan dan menyimpan info ke database dokumen 3. Sistem menghitung kemiripan dokumen menggunakan algoritma winnowing 4. Sistem menampilkan info perhitungan dan menyimpan data ke database dokumen

3.4.2.3.4 Spesifikasi Proses Download Dokumen

Spesifikasi Proses pada proses 1.2 Hitung Kemiripan Teks Algoritma Winnowing dapat dilihat pada table 3.10.

Tabel 3.10 Spesifikasi Proses Download Dokumen

No	Proses	Keterangan
1	No Proses	3.1
	Nama	Download Dokumen
	Source	3.2 Lihat Upload Dokumen, tb_dokumen
	Input	-
	Output	Isi File
	Logika Proses	1. Pengguna memilih link download pada tabel dokumen 2. Sistem menampilkan file yang bersumber dari tb_dokumen dalam database data_dokumen

3.4.2.3.5 Spesifikasi Proses Lihat Upload Dokumen

Spesifikasi Proses pada proses 3.2 Lihat Upload Dokumen dapat dilihat pada table 3.11.

Tabel 3.11 Spesifikasi Proses Lihat Upload Dokumen

No	Proses	Keterangan
1	No Proses	3.2
	Nama	Lihat Upload Dokumen
	Source	tb_dokumen
	Input	-
	Output	Dokumen yang pernah di upload
	Logika Proses	1. Pengguna memilih menu hitori

		2. Sistem menampilkan data dokumen yang pernah diupload yangbbersumber dari tb_dokumen dalam database data_dokumen

3.4.2.3.6 Spesifikasi Proses Lihat Histori Perhitungan Dokumen

Spesifikasi Proses pada proses 3.3 Lihat Histori Perhitungan Dokumen dapat dilihat pada table 3.11.

Tabel 3.11 Spesifikasi Proses Lihat Histori Perhitungan Dokumen

No	**Proses**	**Keterangan**
1	No Proses	3.3
	Nama	Lihat Histori Perhitungan Dokumen
	Source	Tabel histori
	Input	-
	Output	Info Dokumen yang pernah dilakukan perhitungan
	Logika Proses	1. Pengguna memilih menu histori 2. Sistem menampilkan data dokumen yang pernah dilakukan perhitungan yang bersumber dari tabel histori dalam database data_dokumen

3.4.3 Kamus Data

Kamus data dibuat berdasarkan aliran data yang ada pada DFD, dengan kamus data data-data yang mengalir dalam DFD didefinisikan secara detail. Kamus data pada sistem deteksi similaritas dokumen dapat dilihat pada tabel 3.12

Tabel 3.12 Tabel Kamus Data

No	Kamus	Keterangan
1	Nama Aliran Data	Data Input Teks
	Digunakan Pada	Proses 1.1, Proses 2.1
	Deskripsi	Berisi data input yang dilakukan pengguna
	Struktur Data	nama_dok+ +gram+window
	nama_dok gram window file	[a-z|A-Z] [0-9] [0-9] [a-z|A-Z|0-9]
2	Nama Aliran Data	Data Dokumen
	Digunakan Pada	Proses 1.1, Proses 2.1, Proses 3.1, Proses 3.2
	Deskripsi	Berisi data hasil input pengguna sebelum diolah
	Struktur Data	Id_dok+nama_dok+gram+window+tanggal+file+ fingerprints
	Id_dok nama_dok gram window tanggal file fingerprints	[0-9] [a-z] [0-9] [0-9] [0-9|-] [a-z|A-Z|0-9] [0-9|,]
3	Nama Aliran Data	Data Hasil Perhitungan
	Digunakan Pada	Proses 1.3, Proses 2.3, Proses 3.3
	Deskripsi	Berisi data hasil perhitungan sistem
	Struktur Data	Id_proses+nama_dokumen+gram+window+ algoritma+persentasi+waktu_proses

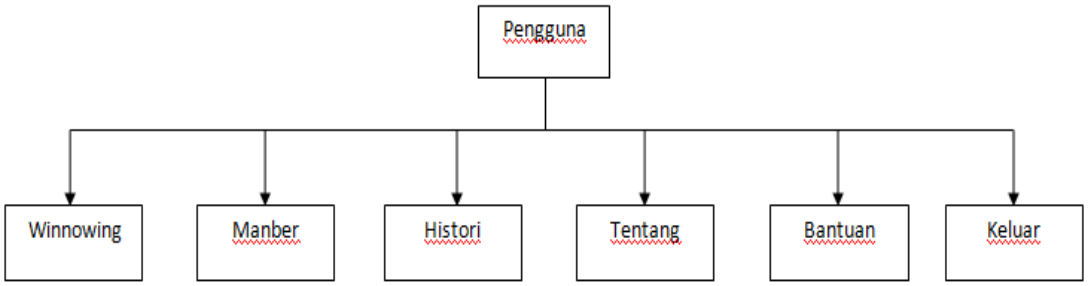
	id_proses	[0-9]
	nama_dokumen	[a-z]
	gram	[0-9]
	window	[0-9]
	algoritma	[Algoritma Winnowing|Algoritma Manber]
	persentasi	[0-9|%]
	waktu_proses	[0-9]

3.4.4 Perancangan Sistem

Pada tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

3.4.4.1 Perancangan Struktur Menu

Perancangan struktur menu dilakukan sebagai panduan pengguna dalam memilih menu dan fungsi yang tersedia di halaman tampilan. Struktur menu pengguna dapat dilihat pada gambar 3.8.



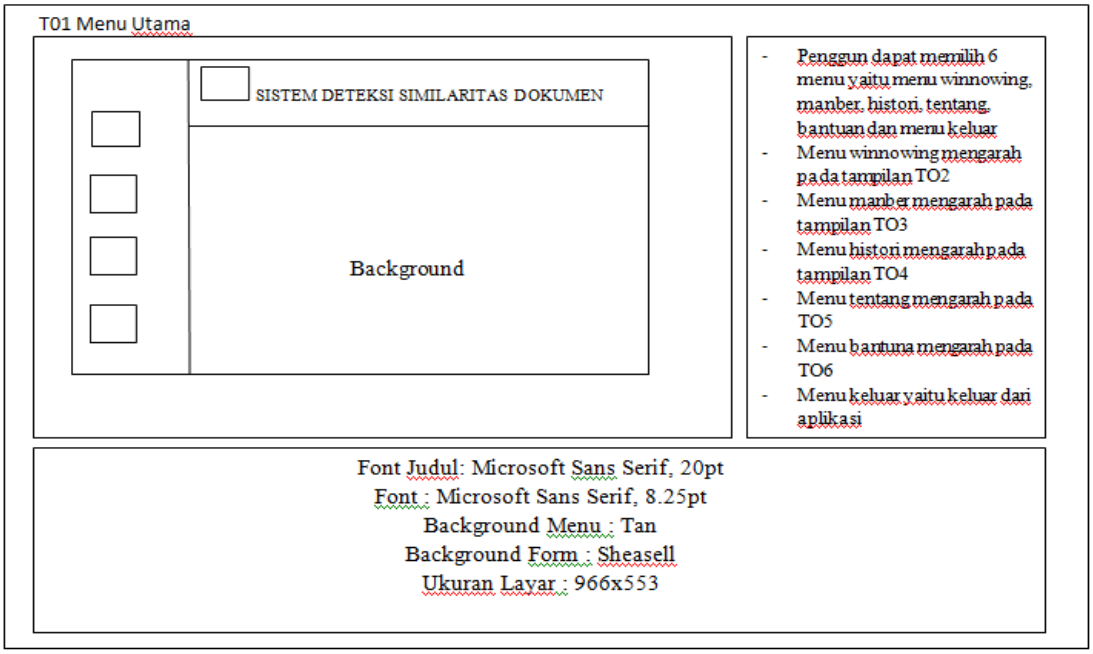
Gambar 3.8 Perancangan Struktur Menu Pengguna

3.4.4.2 Perancangan Antar Muka

Perancangan antar muka Sistem Deteksi Similaritas Dokumen dilakukan agar membantu dalam proses implementasi antar muka sistem dari hasil analisis yang telah dilakukan.

3.4.4.2.1 Perancangan Antar Muka T01

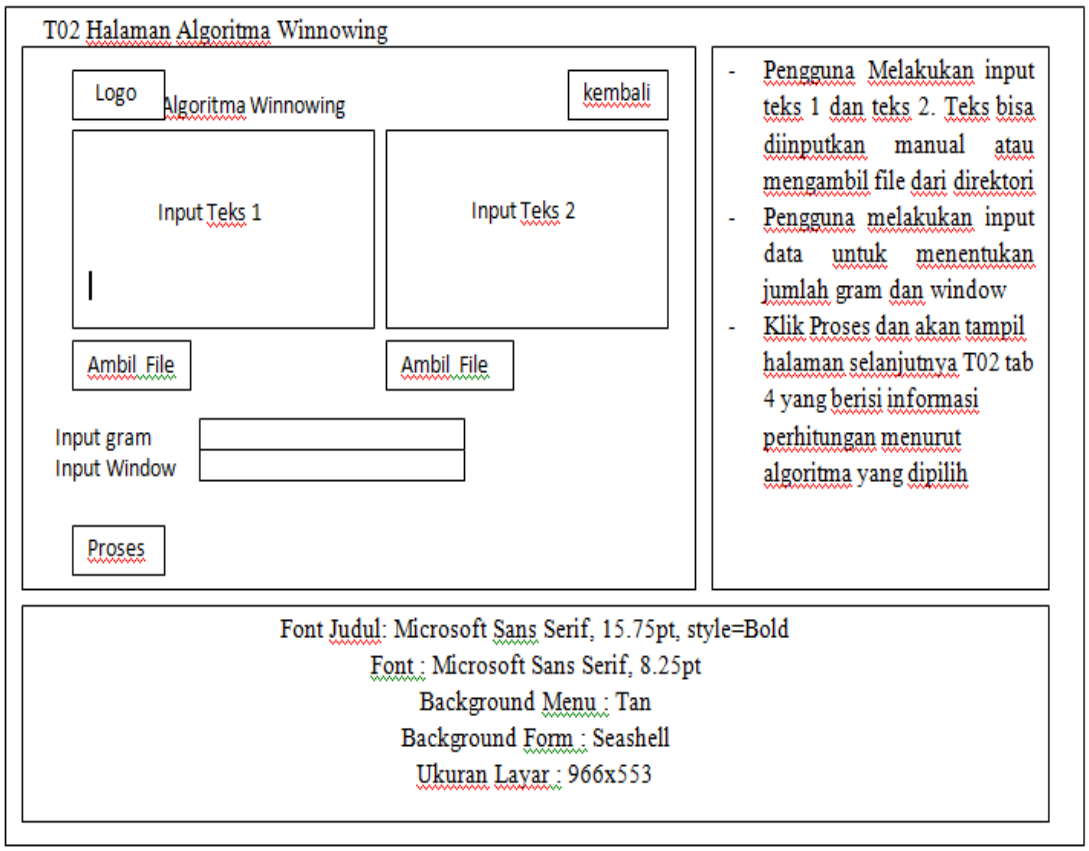
Antar muka halaman pertama yaitu berisi menu-menu pilihan yang dapat dilihat pada gambar 3.9.



Gambar 3.9 Antar Muka Menu Utama

3.4.4.2.2 Perancangan Antar Muka T02

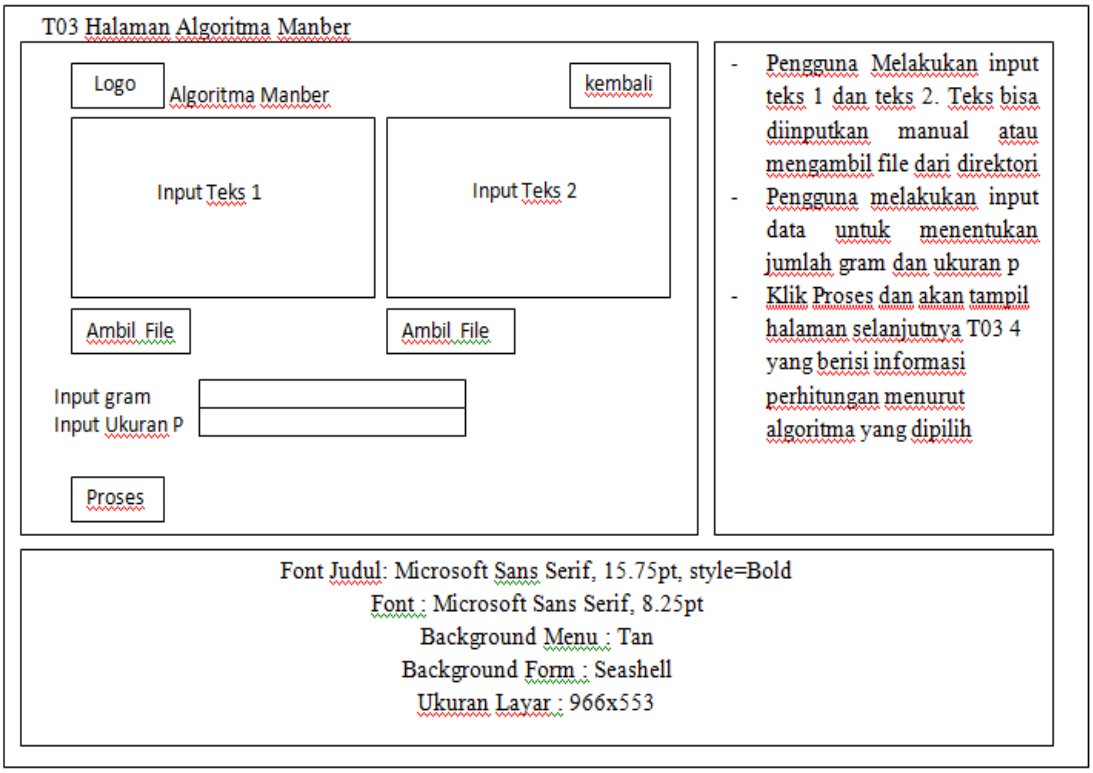
Antar Muka halaman kedua yaitu berisi form inputan teks pertama, teks kedua, jumlah gram, dan jumlah window yang diperlukan untuk perhitungan persentasi kemiripan menggunakan Algoritma Winnowing. Antar muka T02 dapat dilihat pada gambar 3.10.



Gambar 3.10 Antar Muka Winnowing

3.4.4.2.3 Perancangan Antar Muka T03

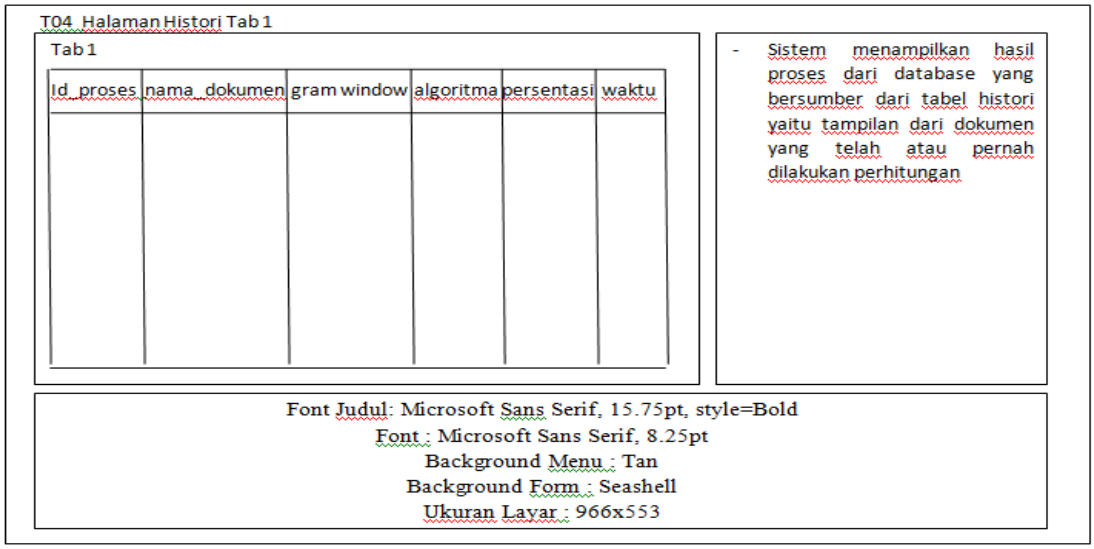
Antar Muka halaman kedua yaitu berisi form inputan teks pertama, teks kedua, jumlah gram, dan jumlah window yang diperlukan untuk perhitungan persentasi kemiripan menggunakan Algoritma Winnowing. Antar muka T02 dapat dilihat pada gambar 3.11.



Gambar 3.11 Antar Muka Manber

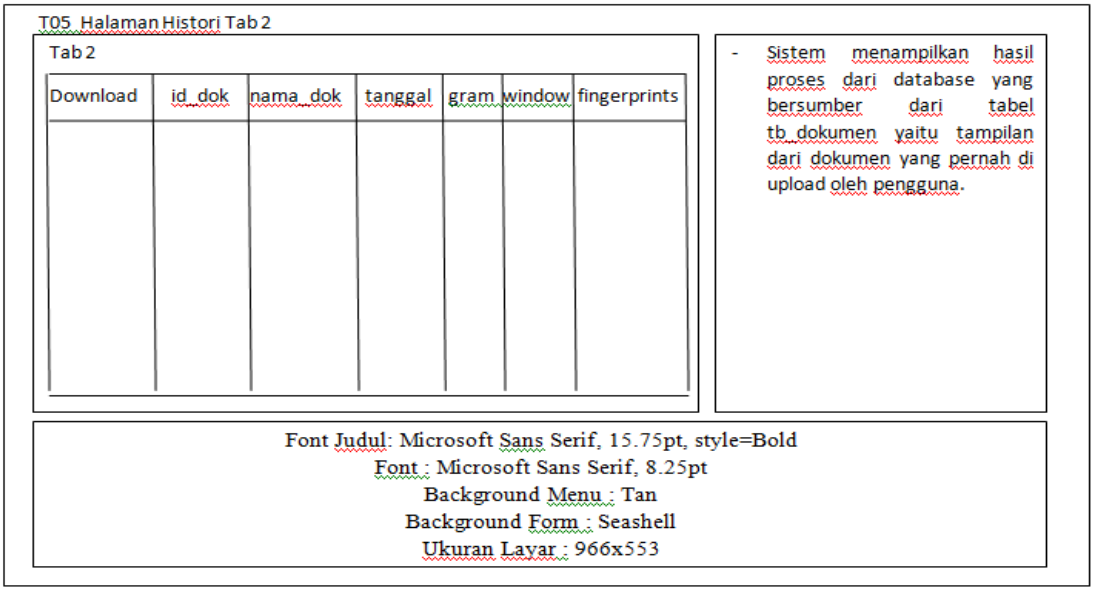
3.4.4.2.4 Perancangan Antar Muka T04

Antar muka T04 berisi tampilan antar muka riwayat hasil dari proses yang telah dilakukan. Halaman T04 menampilkan keseluruhan proses yang telah dilakukan berupa table dan keterangan-keterangan. Halaman TO4 memiliki 2 tab yaitu tab Histori Proses dan tab Histori Dokumen. Halaman T04 dapat dilihat pada gambar 3.12.



Gambar 3.12 Antar Muka Histori Tab 1

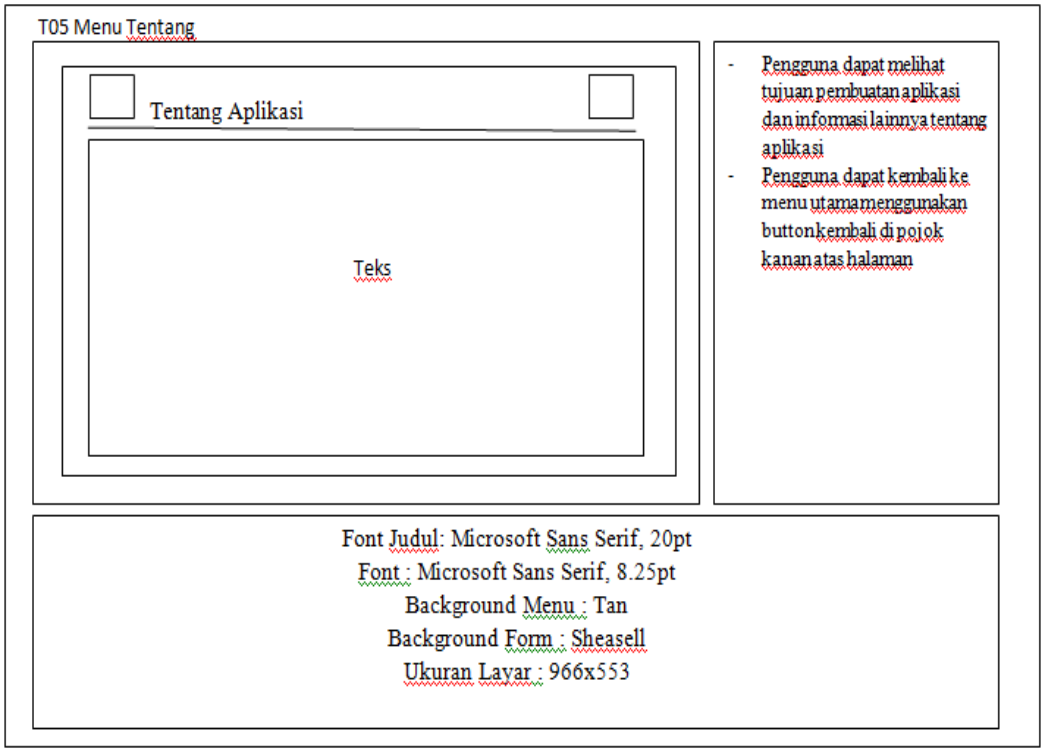
Tampilan berikutnya yaitu tampilan TO4 tab 2 yaitu berisi tentang dokumen apa saja yang pernah di unggah pengguna. Tampilan TO4 tab 2 dapat dilihat pada gambar 3.13.



Gambar 3.13 Antar Muka Histori Tab 2

3.4.4.2.5 Perancangan Antar Muka T05

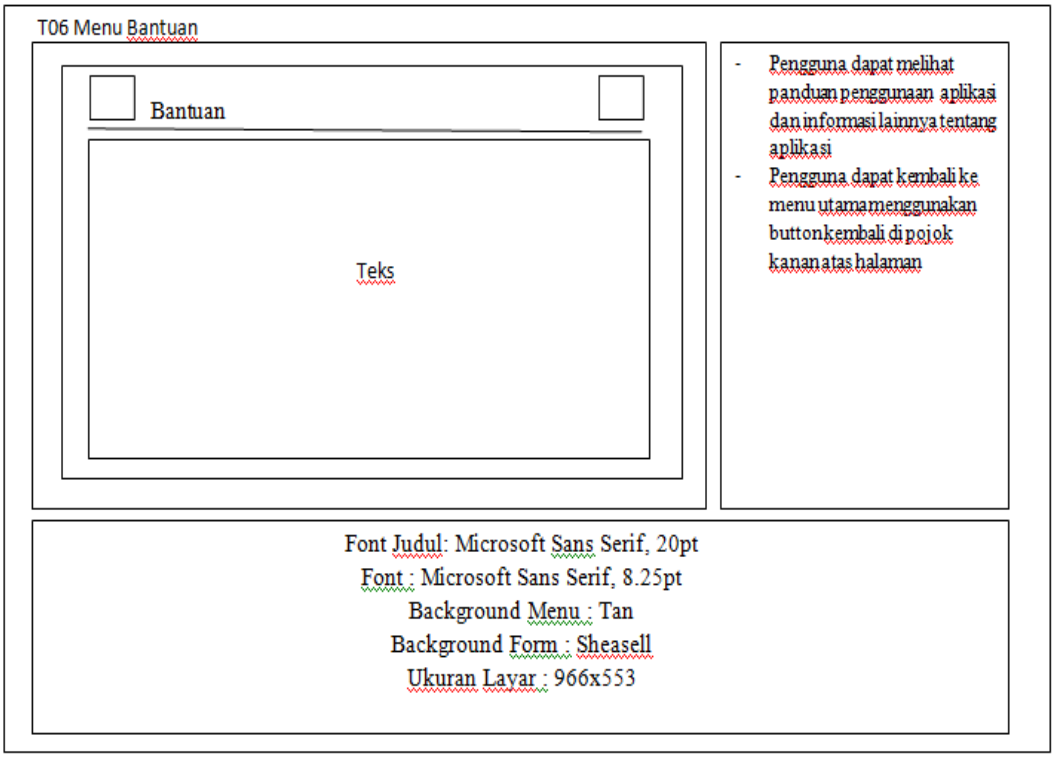
Antar muka T05 merupakan antar muka yang menampilkan menu tentang yang berisi tentang teks tujuan dibuatnya aplikasi dan deskripsi aplikasi. Antar muka TO5 dapat dilihat pada gambar 3.14.



Gambar 3.14 Antar Muka Tentang Aplikasi

3.4.4.2.6 Perancangan Antar Muka T06

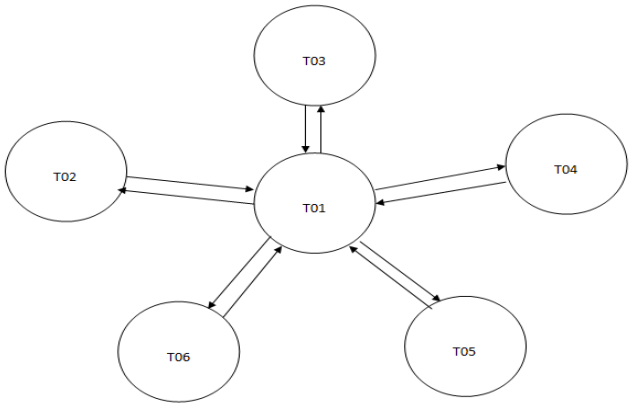
Antar muka T06 merupakan antar muka yang menampilkan menu bantuan yang berisi tentang cara penggunaan aplikasi. Antar muka TO6 dapat dilihat pada gambar 3.15.



Gambar 3.15 Antar Muka Bantuan

3.4.4.3 Jaringan Semantik

Jaringan semantik menggambarkan keterkaitan antar halaman yang dibangun. Jaringan semantik sistem dapat dilihat pada gambar 3.16.



Gambar 3.16 Jaringan Semantik

3.4.4.4 Perancangan Basis Data

Perancangan basis data merupakan perancangan database yang meliputi struktur data dari suatu tabel dalam database. Tabel di bawah ini merupakan tabel beserta *field* dan *data type* yang digunakan dalam *database* dengan menggunakan *MYSQL Database*

3.4.4.4.1 Tabel tb_dokumen

Tabel *tb_dokumen* merupakan tabel untuk menampung data dokumen yang di upload oleh pengguna. Dapat dilihat pada tabel 3.13.

Tabel 3.13 Tabel tb_dokumen

Atribut	Type	Keterangan
Id_dok	Int(10)	Primary key dari tabel
nama_dok	Varchar(30)	Nama dari dokumen yang di upload
tanggal	DateTime	Tanggal ketika pengguna upload
gram	Int(10)	Jumlah gram yang diinputkan
window	Int(10)	Jumlah window yang diinputkan
file	Varbinary(50000)	Berisi file yang di upload pengguna
fingerprints	Varchar(10000)	Nilai fingerprints dari proses perhitungan dari dokumen yang diupload

3.4.4.4.2 Tabel histori

Tabel *histori* merupakan tabel untuk menampung data dokumen yang di upload oleh pengguna. Tabel histori dapat dilihat pada tabel 3.14.

Tabel 3.14 Tabel Histori

Atribut	Type	Keterangan

Id_proses	Int(11)	Primary key dari tabel
nama_dokumen	Varchar(50)	Nama dari dua dokumen yang telah dilakukan perhitungan
gram	Int(10)	Jumlah gram yang diinputkan
window	Int(10)	Jumlah window yang diinputkan
algoritma	Varchar(30)	Nama algoritma yang dipilih ketika proses perbandingan dua dokumen
persentasi	Varchar(30)	Nilai persentasi dari perbandingan dua dokumen
waktu_proses	Varchar(30)	Total waktu proses ketika perbandingan dua dokumen berlangsung

3.4.4.4.3 Tabel Stopword

Tabel *stopword* merupakan tabel untuk menampung kata-kata yang masuk dalam kategori kata yang tidak memiliki makna namun sering digunakan (kata *stopword*). . tabel stopword dapat dilihat pada tabel 3.15.

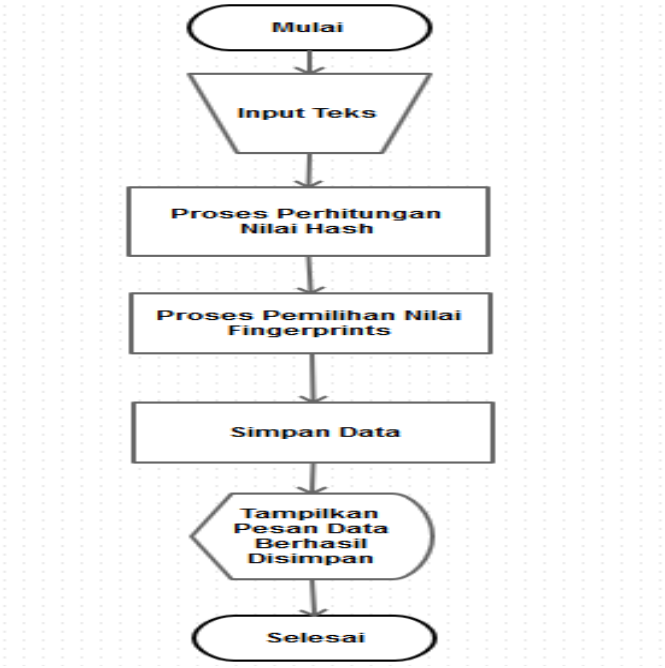
Tabel 3.15 Tabel Stopword

Atribut	Type	Keterangan
Id_stopword	Int(10)	Primary key dari tabel
stopword	Varchar(30)	Kata stopword

3.4.4.5 Perancangan Prosedural

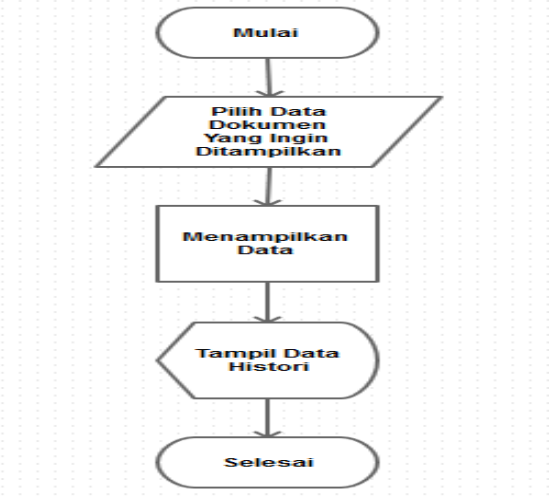
Perancangan prosedural adalah untuk menggambarkan rancangan prosedur yang ada pada pembuatan aplikasi ini. Terdapat dua prosedur yang dapat dilakukan pengguna yaitu prosedur input data dan prosedur cek histori .Perancangan prosedural dapat dilihat sebagai berikut:

1. Prosedur Input Data



Gambar 3.17 Prosedur Input Teks

2. Prosedur Cek Histori



Gambar 3.18 Prosedur Cek Histori