BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Masalah

Metode stemming merupakan metode yang digunakan untuk proses pencarian bentuk dasar dari suatu kata pada sebuah dokumen teks. Hal ini telah dibuktikan pada jurnal "Algoritma porter stemmer for bahasa indonesia untuk pre-processing text mining berbasis metode market basket analysis" karya Gregorius S. Budhi, Ibnu Gunawan,dan Ferry Yuwono[3] . Pada jurnal ini, metode stemming digunakan untuk mengatasi masalah pencarian informasi yang tersimpan didalam dokumen secara efektif dan efisien. Penelitian ini mendapatkan kesimpulan bahwa metode stemming dalam tahap pre-processing sangat berguna karena dapat mengurangi lama waktu proses dan kesalahan proses stem kata cukup kecil, yaitu 2% sehingga dapat diatasi dengan cepat menggunakan pemeriksaaan kembali secara manual terhadap hasil stemmer[3].

Algoritma stemming porter memiliki kelebihan dalam hal kecepatan dan memiliki kelemahan dalam hal keakuratan, hal ini dibuktikan pada penelitian sebelumnya yang berjudul "perbandingan Algoritma stemming Porter dan algoritma stemming Adriani Nazief untuk dokumen teks berbahasa indonesia" oleh Ledy Agusta dalam jurnalnya[1].

Algoritma Arifin setiono memiliki kelebihan dalam hal mengatasi overstemming atau dalam hal keakuratan hal ini dibuktikan pada penelitian sebelumnya yang berjudul"Pengaruh Stemming Kata Dalam Peningkatan Untuk Kerja Dokumen Clustering Untuk Dokumen Teks Berbahasa Indonesia" oleh amir hamzah dalam jurnalnya[4].

Dalam penelitian ini akan dilakukan penggabungan dua algoritma tersebut pada dokumen teks berbahasa indonesia agarmenghasilkan hasil yang lebih baik dengan menguji parameter berupa kecepatan dan keakuratan pada algoritma yang diimplementasikan.

3.2 Analisis data masukan

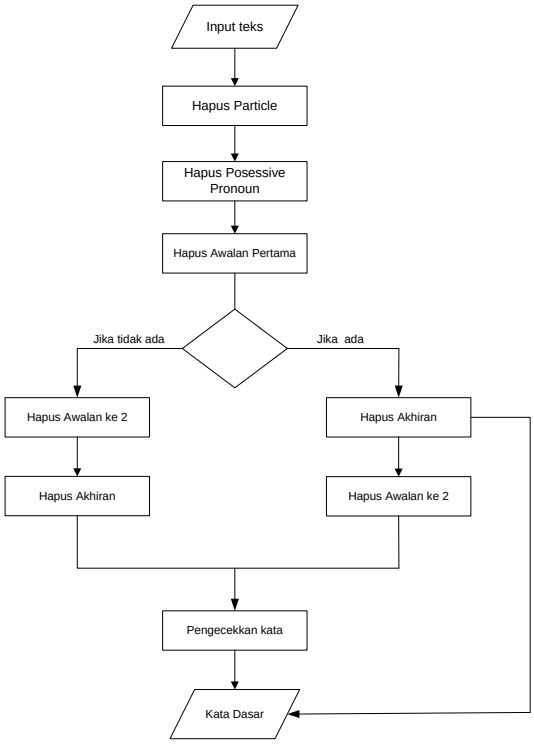
Data yang digunakan dalam menstemming kata dasar pada sistem adalah data berupa teks dari jurnal berekstensi dot(.)*txt. Untuk penelitian ini data diambil kemudian selain teks maka akan dihilangkan dengan tahapan tokenizing dan kamus sebagai pembanding kata yang di *stemming*.

3.3 Analisis metode

Analisis metode menjelaskan tentang bagaimana *stemming* itu terjadi dan menjelaskan tentang tahap-tahap yang dilakukan untuk melakukan proses *stemming*. Secara umum sistem ini terdiri dari beberapa proses, proses-proses tersebut adalah sebagai berikut:

1. Proses *Tokenizing*
2. Proses *Filtering*
3. Proses *Stemming*

Berikut ini *flowchart* dari proses-proses yang akan digunakan untuk melakukan *stemming.*

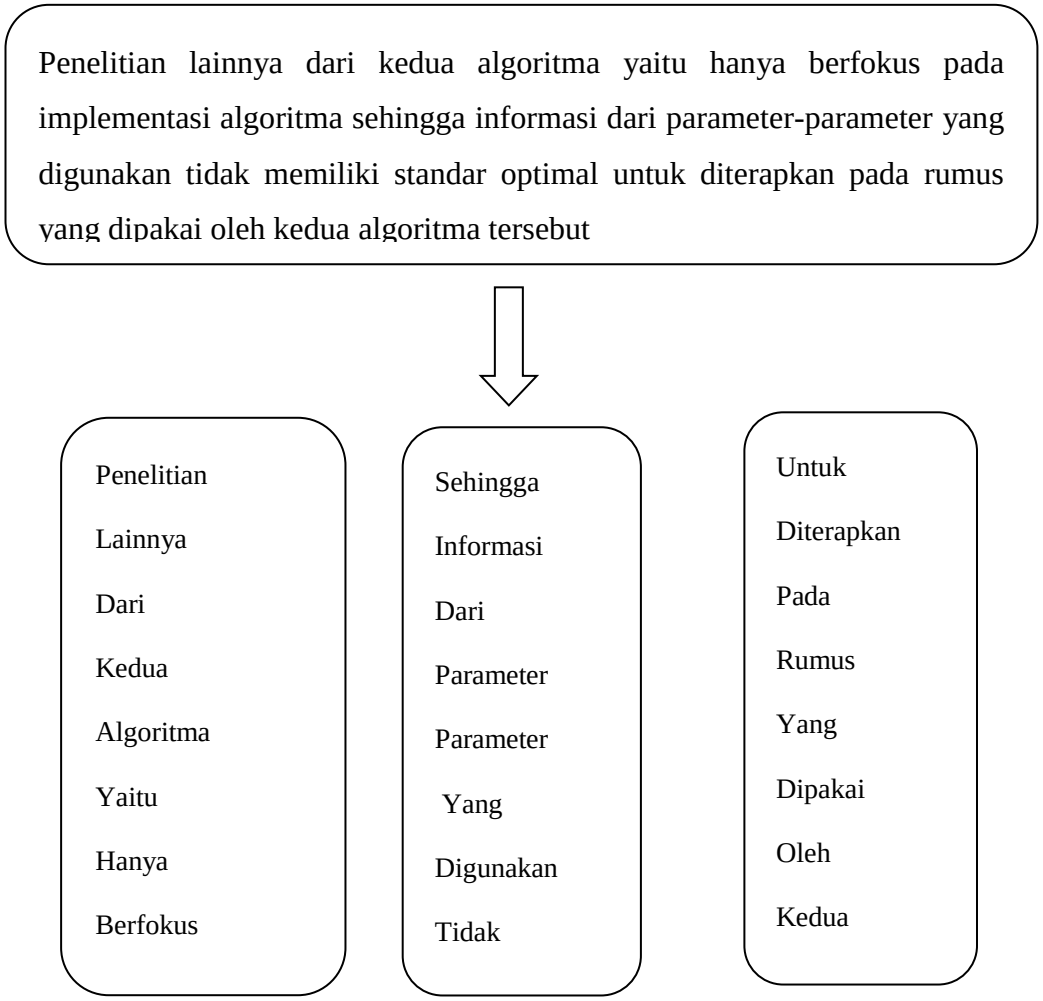


Gambar 3.1 *Flowchart Stemming*

Untuk mengetahui lebih jelas dari proses-proses tersebut maka akan dijelaskan sebagai berikut:

1. Proses *Tokenizing*

Berikut ini adalah langkah-langkah untuk melakukan tokenizing pada sebuah data yang berada dalam dokumen untuk menghilangkan tanda baca, spasi. data dalam dokumen tersebut adalah sebagai berikut:



Gambar 3.2 Proses Tokenizing

2. Proses *Filtering*

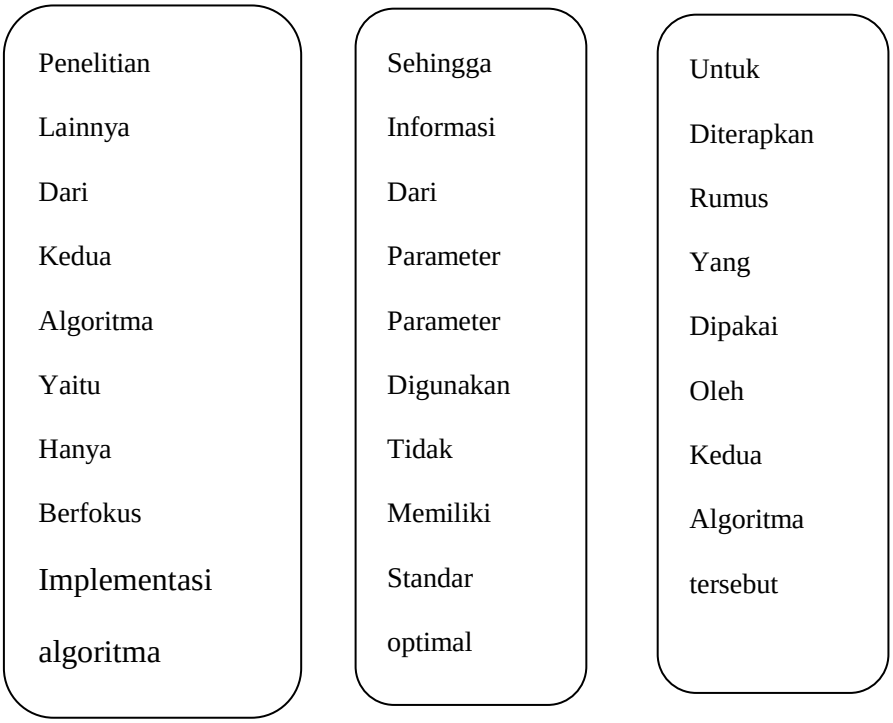
Proses filtering merupakan proses mengambil kata-kata penting yang terdapat dari hasil *tokenizing.* Untuk melakukan *filtering* bisa menggunakan *stoplist* atau *word list.* Data hasil tokenizing akan dibandingkan dengan

kamus. Jika tidak terdapat terdapat dalam kamus maka kata tersebut akan dihapus. Kata-kata yang tersisa merupakan kata yang dianggap penting. Untuk lebih jelasnya tahapan proses filtering adalah sebagai berikut:

a. Kata hasil proses tokenizing dibandingkan dengan tabel stopword.
b. Jika data hasil tokenizing sama dengan kata di tabel stopword maka akan dihapus.
c. Jika tidak sama dengan tabel stopword maka kata tersebut akan disimpan.

Berikut adalah contoh dari proses filtering:

Daftar *stopword* : yang, akan,pada.



Gambar 3.3 Proses Filtering

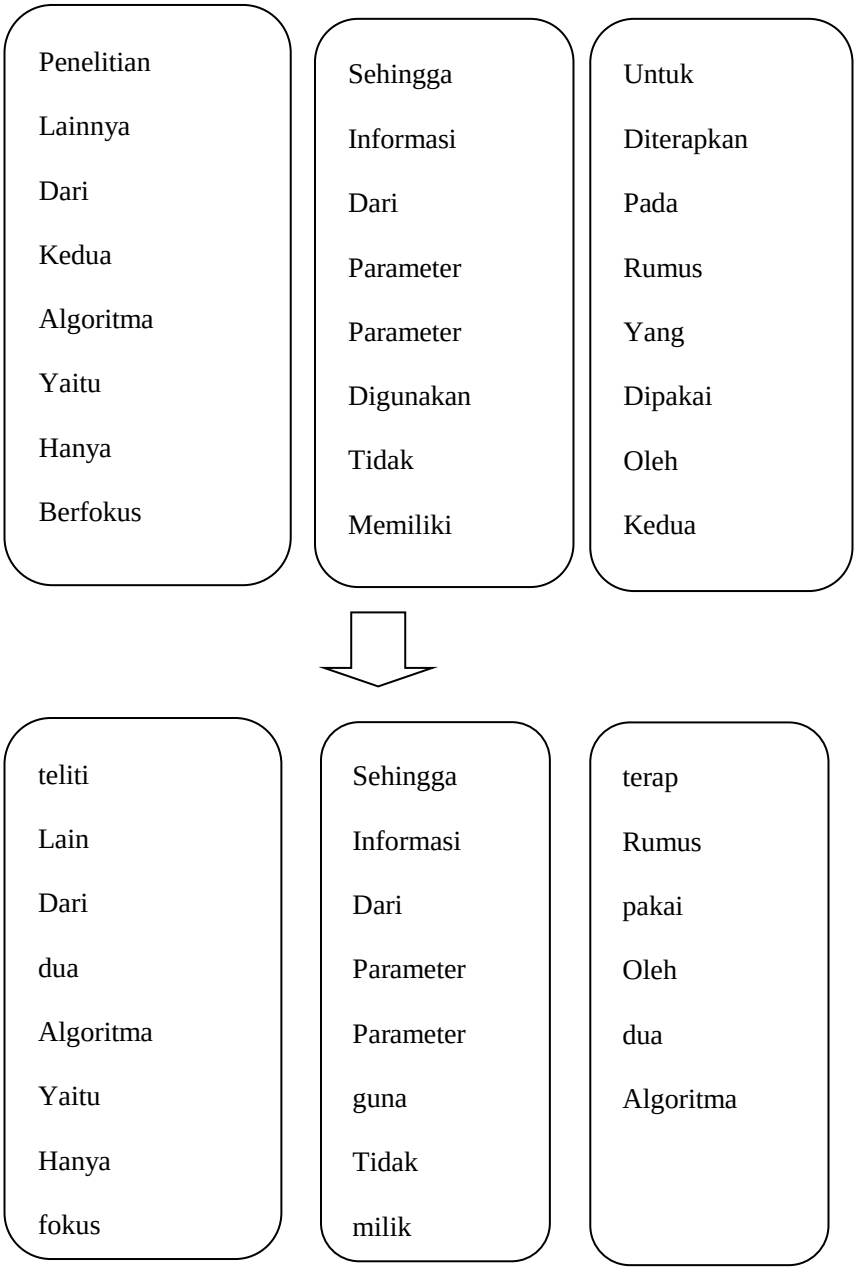
3. Proses Stemming

Proses stemming merupakan pengelolaan kata hasil filtering menjadi kata dasar yaitu dengan cara menghilangkan imbuhan yang diantaranya adalah ”di”, “ke”, “me”, “meng”, “-an”, “-kan” untuk lebih jelasnya tahapan dari proses *stemming* adalah sebagai berikut:

a. Data hasil filtering diperiksa apakah mengandung imbuhan atau tidak.Jika terdapat imbuhan maka akan dilakukan pembuangan imbuhan,terus diulang sampai kata tersebut tidak mengandung imbuhan.

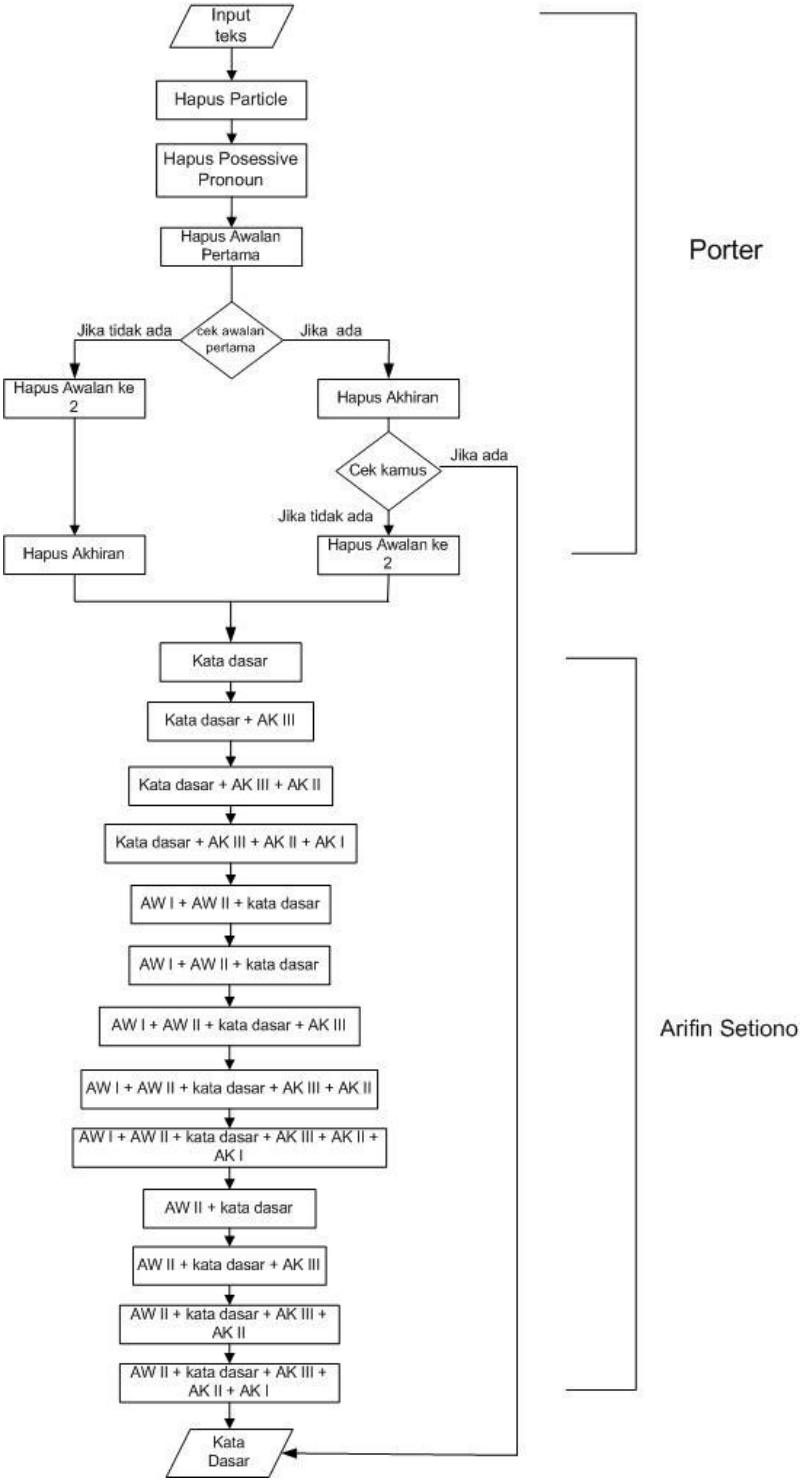
b. Jika sudah tidak mengandung imbuhan maka akan disimpan.

Contoh dari proses stemming adalah sebagai berikut :



Gambar 3.4 proses stemming

3.4 Analisis Penggabungan Algoritma Porter dan Arifin Setiono



Gambar 3.5 Langkah Penggabungan Algoritma

1. Langkah 1:

Contoh kalimat: Berdirinya Pusat Penelitian Informatika tidak dapat dipisahkan dengan sejarah terbentuknya Lembaga Elektroteknika Nasional. Lembaga Elektroteknika Nasional sebagai lembaga penelitian dalam bidang elektronika diresmikan pada tanggal 10 Juni 1965.

Langkah awal pada algoritma yaitu melakukan hapus *particle* pada setiap kata, Karakter selain huruf pada tahap ini akan dihilangkan. Maka kalimat diatas akan menjadi :

Berdirinya	Nasional
Pusat	Lembaga
Penelitian	Elektroteknika
Informatika	Nasional
Dapat	Lembaga
Dipisahkan	Penelitian
Sejarah	Bidan
Terbentuknya	Elektronika
Lembaga	Diresmikan
Elektroteknika	tanggal
	Juni

Gambar 3.6 langkah ke 1

2. Langkah 2:

Langkah berikutnya yaitu proses filetring dan melakukan penghapusan *Possesive pronoun*, maka teks akan menjadi :

Berdiri	Nasional
Pusat	Lembaga
Penelitian	Elektroteknika
Informatika	Nasional
Dapat	Lembaga

Dipisahkan Sejarah Terbentuk Lembaga Elektroteknika	Penelitian Bidang Elektronika Diresmikan tanggal Juni

Gambar 3.7 langkah ke 2

3. Langkah 3:

Langkah berikutnya yaitu proses penghapusan awalan pertama.maka kalimat tersebut menjadi:

Diri Pusat elitian Informatika Dapat pisahkan Sejarah bentuk Lembaga Elektroteknika Nasional Lembaga Elektroteknika Nasional Lembaga Penelitian Bidang Elektronika resmikan

tanggal
Juni

Gambar 3.8 langkah ke 3

4. Langkah 4:

   Langkah berikutnya yaitu proses hapus akhiran. Jika tidak ditemukan maka kata tersebut jika ada yang kurang sesuai dengan kamus maka akan ditambahkan huruf pengganti yang diasumsikan sebagai kata dasar.

diri
Pusat
elit
Informatika
Dapat
pisah
Sejarah
bentuk
Lembaga
Elektroteknika
Nasional
Lembaga
Elektroteknika
Nasional
Lembaga
elit
Bidang
Elektronika
resmi
Pada
tanggal

Juni

Gambar 3.9 langkah ke 4

5. Langkah 5:

   Dikarenakan hasil dari algoritma porter diatas ada beberapa kata yang belum sesuai dengan kamus maka dilakukan proses kombinasi dari algoritma Arifin Setiono. Pada langkah ini akan dilakukan kombinasi dikarenakan ada beberapa kata yang tidak sesuai dengan kamus ,yaitu dengan menggunakan kombinasi kata dasar + akhiran 3 maka kata akan menjadi :

diri	Lembaga
Pusat	Elektroteknika
elit	Nasional
Informatika	Lembaga
Dapat	eliti
pisah	Bidang
Sejarah	Elektronika
bentuk	Resmi
Lembaga	Pada
Elektroteknika	Tanggal
Nasional	juni

Gambar 3.10 langkah ke 5

6. Langkah 6:

   Pada langkah ini akan dilakukan kombinasi dari hasil langkah ke 6 yaitu kata dasar + akhiran 3 + akhiran 2 maka kata akan menjadi :

diri	Sejarah
Pusat	bentuk
eliti	Lembaga
Informatika	Elektroteknika
Dapat	Nasional
pisah	Lembaga

Elektroteknika	Elektronika
Nasional	resmi
Lembaga	Pada
eliti	tanggal
Bidang	Juni

Gambar 3.11 langkah ke 6

7. Langkah 7:

   Pada langkah ini akan dilakukan kombinasi dari hasil langkah ke 6 yaitu kata dasar + akhiran 3 + akhiran 2 + akhiran 1 maka kata akan menjadi: dirinya

Pusat	Elektroteknika
elitian	Nasional
Informatika	Lembaga
Dapat	elitian
pisah	Bidang
Sejarah	Elektronika
bentuk	resmi
Lembaga	Pada
Elektroteknika	tanggal
Nasional	Juni
Lembaga	

Gambar 3.12 langkah ke 7

8. Langkah 8:

   Pada langkah ini akan dilakukan kombinasi dari hasil langkah ke 6 yaitu awalan 1 + awalan 2 + kata dasar maka kata akan menjadi :

Berdiri	Dipisah
Pusat	Sejarah
Peneliti	Terbentuk
Informatika	Lembaga
Dapat	Elektroteknika

Nasional Lembaga Elektroteknika Nasional Lembaga Peneliti	Bidang Elektronika Diresmi tanggal Juni

Gambar 3.13 langkah ke 8

9. Langkah 9:

   Dikarenakan kata masih belum sesuai dengan kata dasar yang berada pada kamus maka langkah dilanjutkan dengan menggunakan kombinasi awalan 1 + awalan 2 + kata dasar + akhiran 3 maka menjadi:Berdiri

Pusat Peneliti Informatika Dapat Dipisah Sejarah Terbentuk Lembaga Elektroteknika Nasional	Lembaga Elektroteknika Nasional Lembaga Peneliti Bidang Elektronika Diresmi tanggal Juni

Gambar 3.14 langkah ke 9

10. Langkah 10 :

    Dikarenakan kata masih belum sesuai dengan kata dasar yang berada pada kamus maka langkah dilanjutkan dengan menggunakan kombinasi awalan 1 + awalan 2 + kata dasar + akhiran 3 + akhiran 2 maka menjadi:

Berdiri	Pusat

Peneliti Informatika Dapat Dipisah Sejarah Terbentuk Lembaga Elektroteknika Nasional Lembaga	Elektroteknika Nasional Lembaga Peneliti Bidang Elektronika Diresmi tanggal Juni

Gambar 3.15 langkah ke 10

11. Langkah 11:

    Dikarenakan kata masih belum sesuai dengan kata dasar yang berada pada kamus maka langkah dilanjutkan dengan menggunakan kombinasi awalan 1 + awalan 2 + kata dasar + akhiran 3 + akhiran 2 + akhiran 1 maka menjadi:

Berdirinya Pusat Penelitian Informatika Dapat Dipisah Sejarah Terbentuk Lembaga Elektroteknika Nasional	Lembaga Elektroteknika Nasional Lembaga Penelitian Bidang Elektronika Diresmikan tanggal Juni

Gambar 3.16 langkah ke 11

12. Langkah 12:

Dikarenakan kata masih belum sesuai dengan kata dasar yang berada pada kamus maka langkah dilanjutkan dengan menggunakan kombinasi awalan 2 + kata dasar maka menjadi:

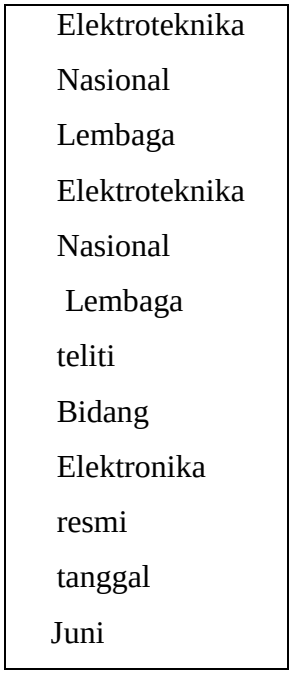
Diri	Lembaga
Pusat	Elektroteknika
Peneliti	Nasional
Informatika	Lembaga
Dapat	Peneliti
pisah	Bidang
Sejarah	Elektronika
bentuk	resmi
Lembaga	tanggal
Elektroteknika	Juni
Nasional	

Gambar 3.17 langkah ke 12

13. Langkah 13:

Setelah ditemukan kata yang sesuai dengan kamus maka proses kombinasi berhenti dan beberapa kata agar lebih sesuai dengan kamus digunakan kata pengganti dan menjadi kata dasar yang sesuai.

diri
Pusat
teliti
Informatika
Dapat
pisah
Sejarah
bentuk
Lembaga



Gambar 3.18 langkah ke 13

3.5 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

3.5.1 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*)

3.5.1.1. Analisis Perangkat Keras

Spesifikasi perangkat keras minimum yang butuhkan untuk menjalankan sistem Deteksi similaritas dokumen dapat dilihat pada tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

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

3.5.1.2. Analisis Perangkat Lunak

Komponen perangkat lunak pendukung yang dibutuhkan untuk menjalankan sistem Deteksi Similaritas Dokumen ini, dapat dilihat pada table 3.2

Tabel 3.2 Spesifikasi Perangkat Lunak

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

3.5.1.3 Analisis Perangkat Pikir

Analisis kebutuhan perangkat pikir merupakan uraian mengenai penguji 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.

3.5.2 Analisis Kebutuhan Fungsional

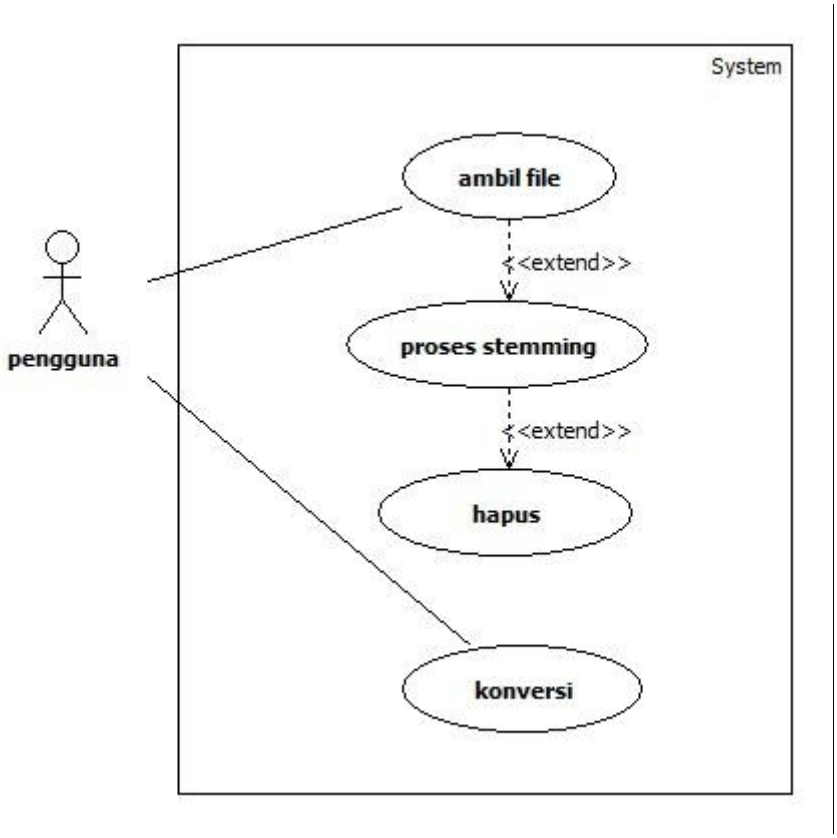
Spesifikasi kebutuhan fungsional adalah spesifikasi tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi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

3.5.2.1 Usecase

Use case merupakan gambaran umum dari perancangan sistem yang akan dibangun. Terdapat tiga buah point yang merincikan use case diagram ini, yaitu :

1. Identifikasi Use Case
2. Identifikasi Aktor
3. Skenario Use Case

Adapun rincian dari use case diagram sistem stemming:



Gambar 3.19 Usecase

3.5.2.1.1 Identifikasi Aktor

Bagian ini diisi dengan daftar actor dan deskripsi role untuk actor tersebut. Deskripsi role harus menjelaskan wewenang pada role tersebut dalam perangkat lunak. Bisa dibuat dalam bentuk tabel berikut:

Tabel 3.3 Identifikasi Aktor

No	Actor	Deskripsi
1	User	User menginputkan document, user input text manual, user merefresh document

3.5.2.1.2 Use Case Skenario Ambil File

Tabel 3.4 Usecase Skenario Ambil File

Use Case Name	Ambil file	
Related Requirements	-	
Goals In Context	mengambil file berekstensi .txt	
Preconditions	Menampilkan tombol Ambil file	
Succesfull End Conditions	Text Terinput ke dalam Form Input	
Failed And Condition	Text Tidak Berhasil diinput	
Primary Actors	User	
Secondary Actors	-	
Trigger	Aplikasi aktif	
Included Case	-	
Main Flow	Step	Action

	1	Pengguna Masuk ke form
	2	Sistem menampilkan form
	3	Pengguna menekan tombol ambil
	4	mengambil document/file berektensi .txt ke directori komputer
	5	Sistem menampilkan hasil inputan

3.5.2.1.3 Use Case Skenario Proses

Tabel 3.5 Usecase Skenario Proses

Use Case Name	Proses	
Related Requirements	-	
Goals In Context	Melakukan proses kata dari form inputan document/file	
Preconditions	Dokument berhasil di input ke form input	
Succesfull End Conditions	Proses kata berhasil dilakukan dan menampilkan hasil	
Failed And Condition	Proses kata gagal dilakukan	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pemilihan tombol proses	
Included Case	-	
Main Flow	Step	Action
	1	Pengguna mengambil document/file berektensi

		.txt ke directori komputer
	2	Pengguna menekan tombol proses
	3	Sistem melakukan proses preprocessing, tokenizing,stemming kemudian melakukan perhitungan kemunculan
	4	Sistem menampilkan hasil

3.5.2.1.4 Use Case Skenario Hapus

Tabel 3.6 Usecase Skenario Hapus

Use Case Name	Hapus	
Related Requirements	-	
Goals In Context	Melakukan Proses hapus dari document inputan	
Preconditions	Proses hapus berhasil dilakukan	
Succesfull End Conditions	Proses hapus berhasil dilakukan	
Failed And Condition	Proses hapus gagal dilakukan	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pemilihan tombol hapus	
Included Case	-	
Main Flow	Step	Action
	1	Pengguna pilih tombol hapus
	2	Sistem menghapus data masukan

3.5.2.1.5 Use Case Skenario Konversi

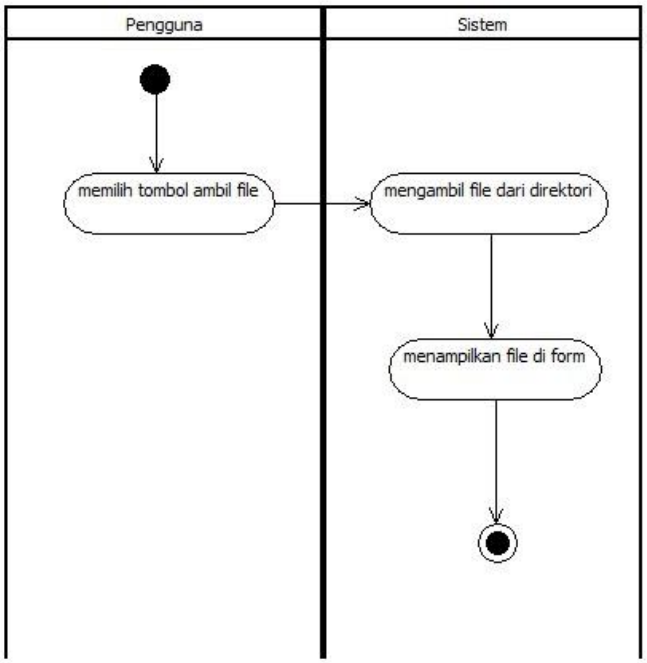
Tabel 3.7 Usecase Skenario Konversi

Use Case Name	Konvert	
Related Requirements	-	
Goals In Context	Melakukan konvert	
Preconditions	Dokument berhasil di konvert	
Succesfull End Conditions	Proses konvert berhasil dilakukan dan menampilkan hasil	
Failed And Condition	Proses konvert gagal dilakukan	
Primary Actors	Pengguna	
Secondary Actors	-	
Trigger	Pemilihan tombol konvert	
Included Case	-	
Main Flow	Step	Action
	1	Pengguna mengambil document/file berektensi .pdf ke directori komputer
	2	Pengguna menekan tombol konvert
	3	Sistem melakukan proses konvert menjadi .txt
	4	Sistem menampilkan hasil

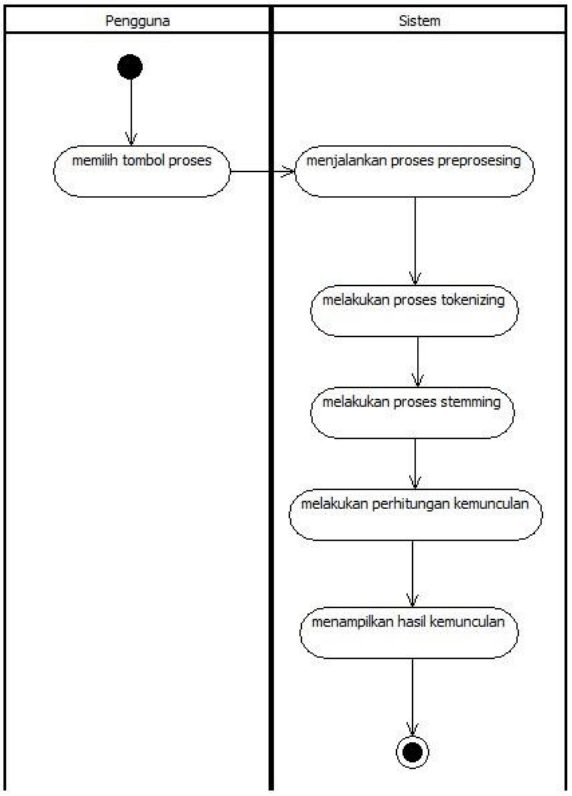
3.5.2.2 Activity Diagram

Activity diagrams menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat

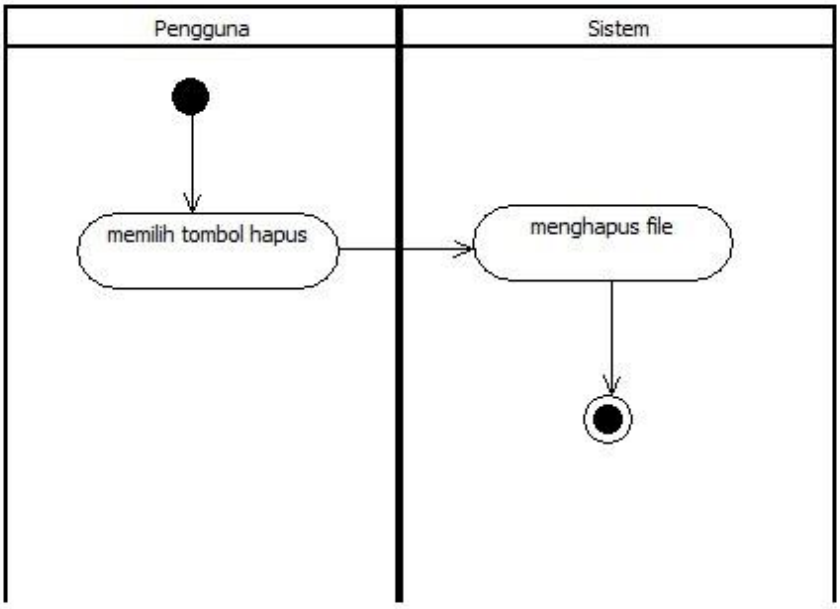
menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum



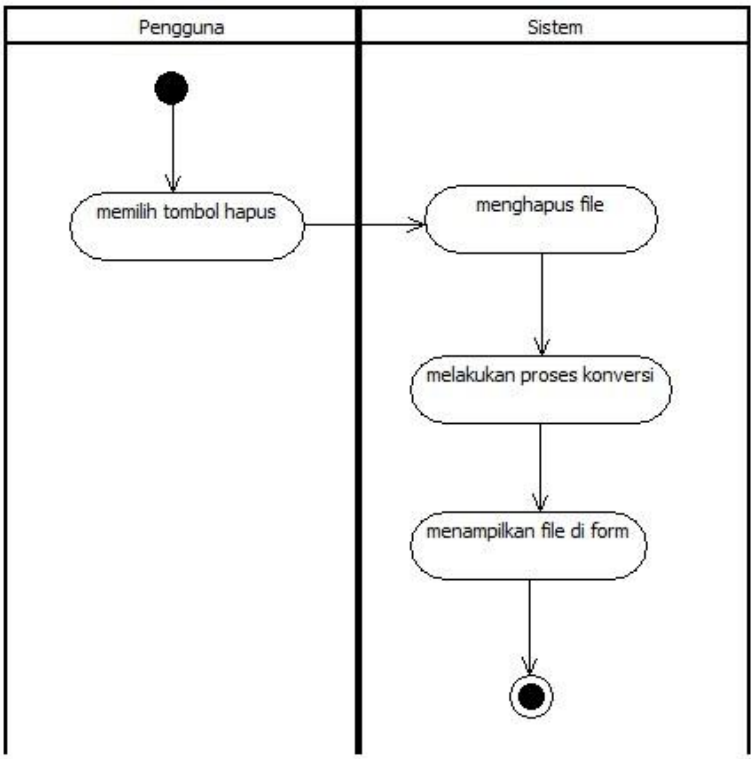
Gambar 3.20 Activity Diagram ambil file



Gambar 3.21 Activity Diagram Proses Stemming



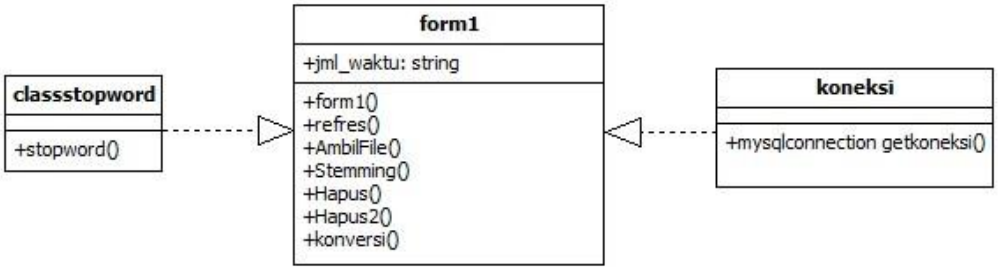
Gambar 3.22 Activity Diagram Hapus



Gambar 3.23 Activity diagram konversi

3.5.2.3 Class Diagram

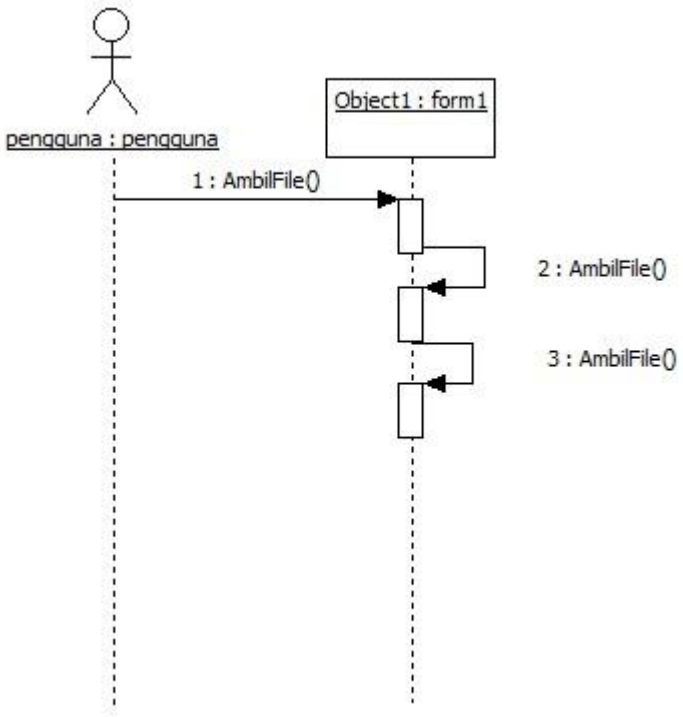
Class diagram adalah diagam yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita gunakan.Class diagram memberi kita gambaran (diagram statis) tentang sistem/perangkat lunak dan relas-relasi yang ada didalamnya. Berikut adalah class diagram dari sistem yang dibangun:



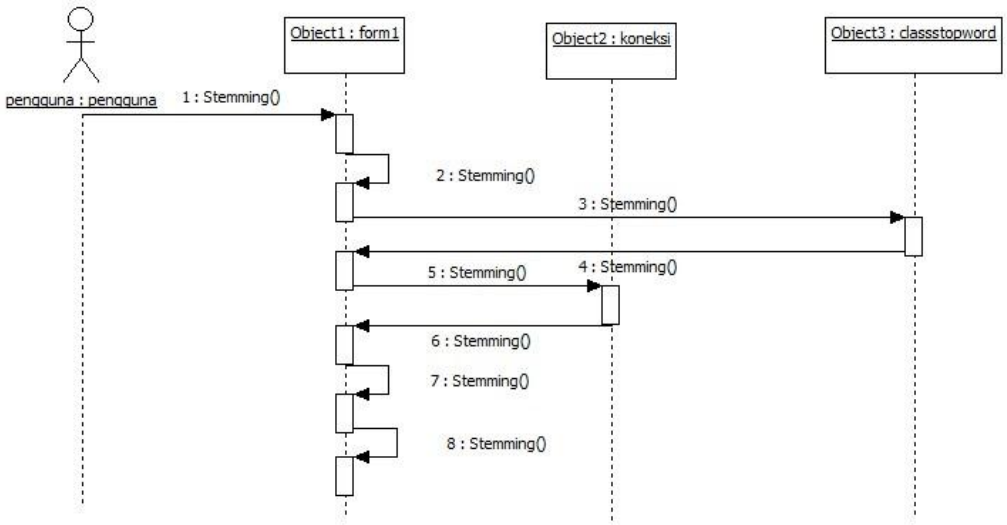
Gambar 3.24 Class Diagram

3.5.2.4 Sequence Diagram

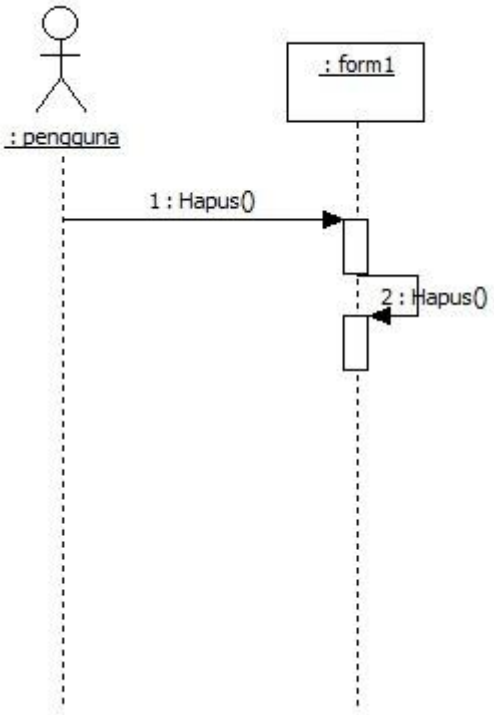
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.



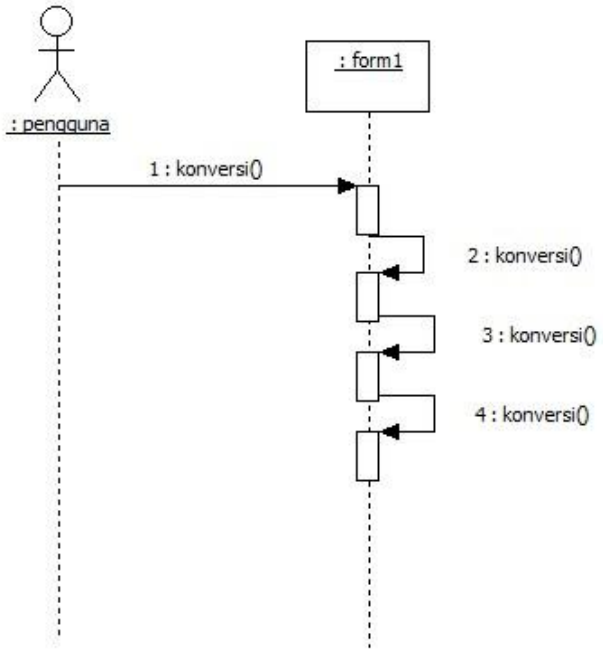
Gambar 3.25 Sequence Diagram Ambil File



Gambar 3.26 Sequence Diagram Proses Stemming



Gambar 3.27 Sequence Diagram hapus



Gambar 3.28 Sequence Diagram konversi

3.5.3 Perancangan sistem

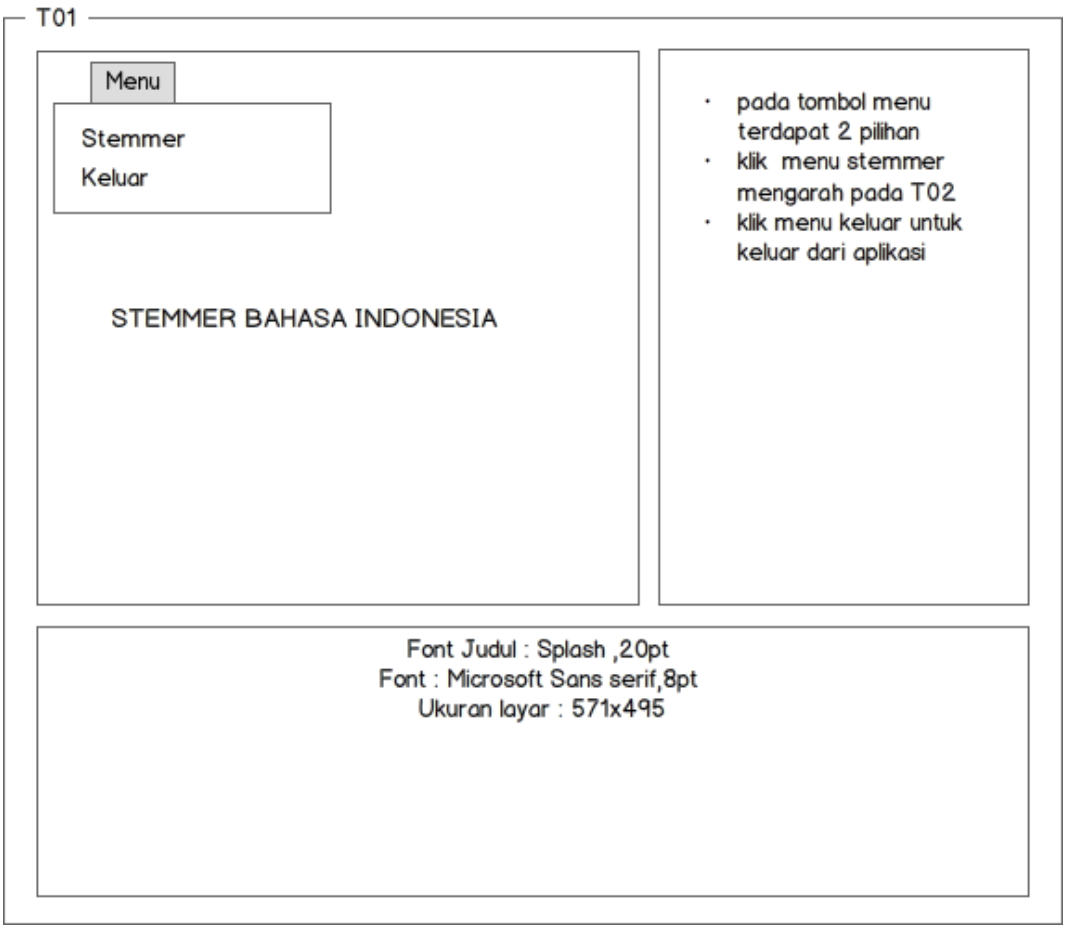
Pada tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

3.5.3.1 Perancangan antarmuka

Perancangan antar muka Sistem Stemming Dokumen teks berbahasa Indonesia dilakukan agar membantu dalam proses implementasi antar muka sistem dari hasil analisis yang telah dilakukan.

3.5.3.1.1 Perancangan Antar Muka T01

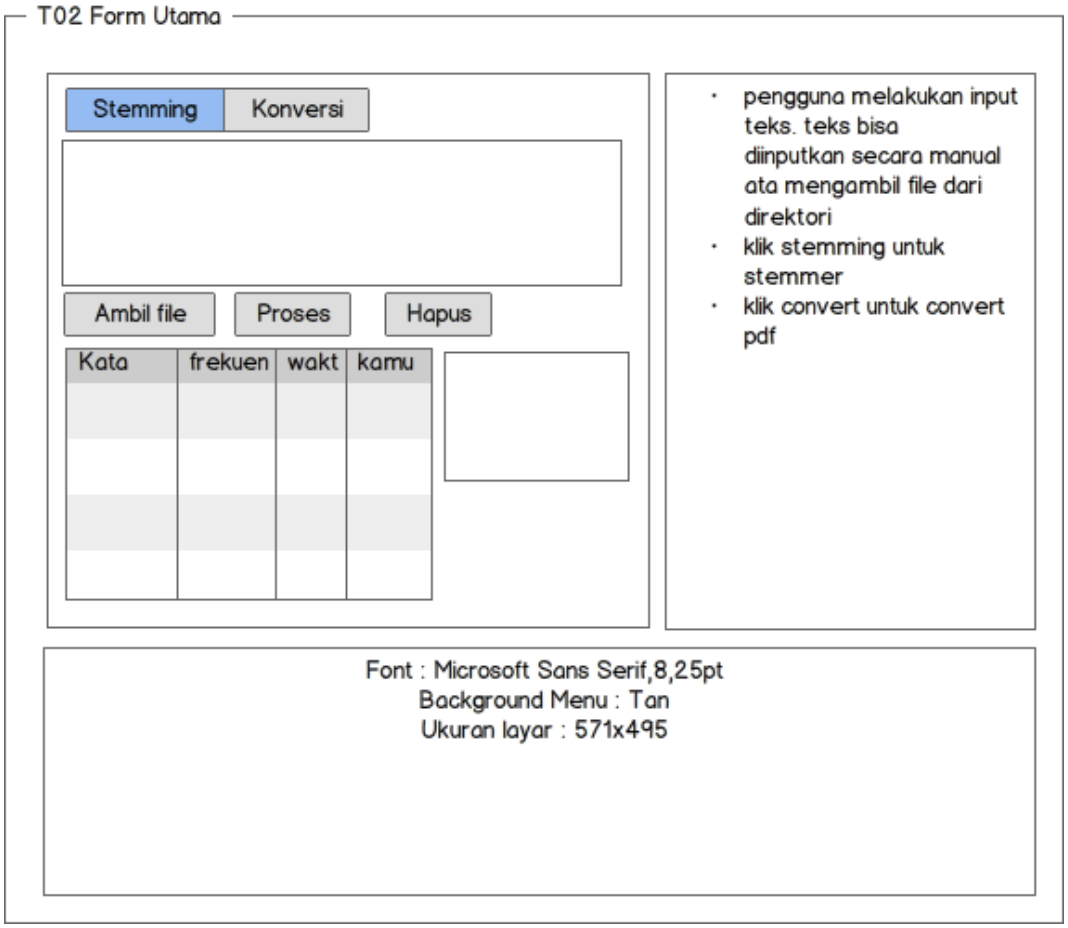
Antar muka halaman pertama yaitu berisi menu-menu pilihan yang dapat dilihat pada gambar 3.29.



Gambar 3.29 Antar Muka Menu Utama

3.5.3.1.2 Perancangan Antar Muka T02

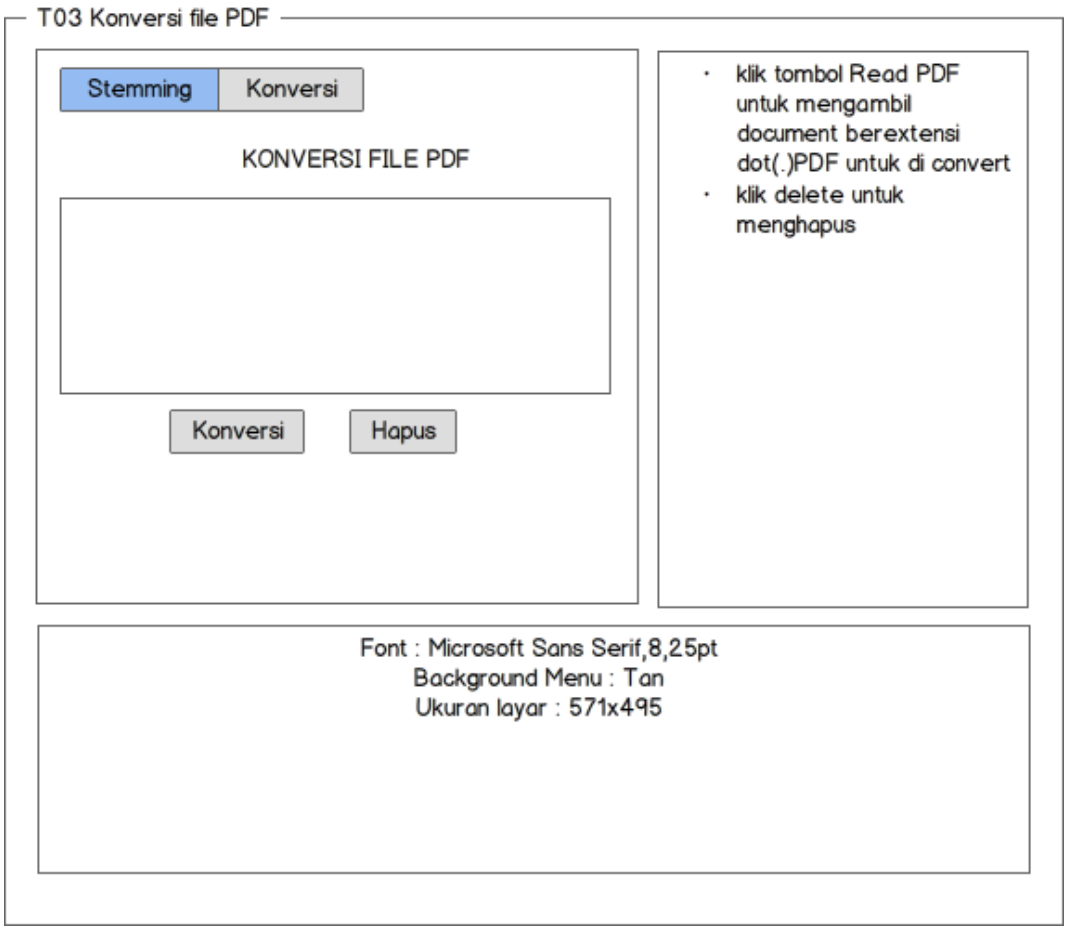
Antar Muka halaman kedua yaitu berisi form inputan teks, button Ambil, button proses, button hapus yang diperlukan untuk perhitungan waktu proses. Antar muka T02 dapat dilihat pada gambar 3.30.



Gambar 3.30 Antar Muka Form Utama

3.5.3.1.3 Perancangan antar muka T03

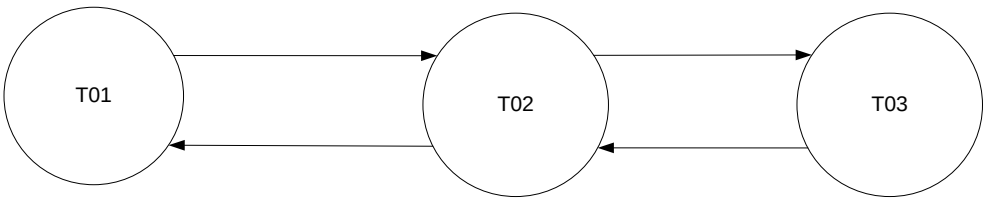
Antar Muka halaman ketiga yaitu berisi form untuk *Convert* dari PDF ke text yang diperlukan untuk masukan data untuk diproses. Antar muka T03 dapat dilihat pada gambar 3.31.



Gambar 3.31 Antarmuka konversi dari *PDF*

3.6 Jaringan Semantik

Jaringan semantik menggambarkan keterkaitan antar halaman yang dibangun. Jaringan semantik sistem dapat dilihat pada gambar 3.28



Gambar 3.32 Jaringan Semantik