BAB 3
ANALISIS DAN PERANCANGAN

3.1. Analisis

Analisis merupakan suatu proses untuk menerjemahkan konsep ke dalam bagian-bagian yang lebih sederhana, sehingga struktur logikanya dapat dibaca dengan mudah dan menjadi lebih jelas. Analisis merupakan metode untuk menguji, menilai, dan memahami sistem pemikiran yang rumit dengan memecahnya kedalam unsur-unsur yang lebih sederhana sehingga hubungan antar unsur-unsur itu menjadi jelas.

3.1.1. Analisis Masalah

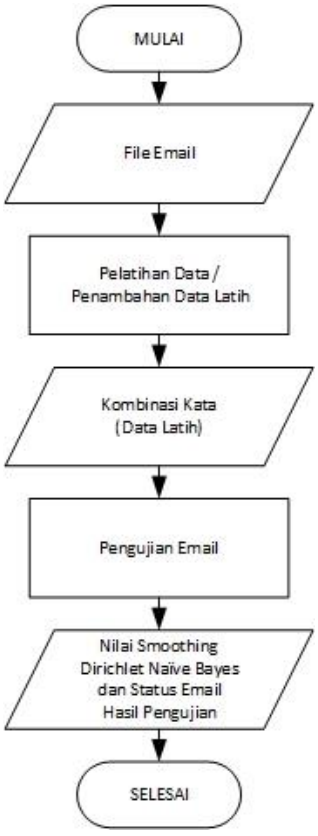
Mengetahui *email spam* yang masuk ke dalam *inbox email* merupakan hal yang sebaiknya dapat dikenali. *Spam* yang terkirim kepada pengguna layanan *email* dapat menimbulkan masalah berupa dapat menghabiskan kapasitas penyimpanan dan menghabiskan waktu pengguna untuk menghapus *spam* tersebut. Oleh karena itu, mulai muncul dan terus dikembangkan *spam filtering* agar *email spam* tidak masuk ke *inbox email*. Metode yang paling banyak digunakan dalam *spam filtering* yaitu Metode Naïve Bayes. Metode Naïve Bayes banyak digunakan karena kesederhanaan algoritmanya dan mudah dalam implementasi. Kelemahan dari metode Naïve bayes yaitu saat memprediksi peluang *email* tersebut dikatakan *spam* hanya melihat peluang perkata dari *subject email*-nya, sedangkan *subject* dapat memiliki kata yang lebih dari satu dan saling berkaitan. Kelemahan lainnya tidak menutup kemungkinan munculnya peluang nilai satu.

Metode untuk mengatasi kelemahan dari metode Naïve Bayes digunakan metode Teori Kombinasi agar dapat melihat keterkaitan antar kata pada *subject*-nya sehingga dapat mengenal pola dari *subject email* tersebut. Dibutuhkan juga cara agar metode Naïve Bayes tidak menghitung peluang nilai 1 yang muncul dengan menggunakan Metode *Smoothing Dirichlet*. Dalam sistem yang akan dibangun urutan proses pengujian yaitu Proses Kombinasi dan Proses *Smoothing Dirichlet*

Naïve Bayes, sedangkan untuk proses penambahan data latih hanya menggunakan Proses Kombinasi saja. Proses Kombinasi akan menghasilkan suatu himpunan kombinasi-kombinasi kata yang terbentuk dari *subject email*, untuk proses penambahan data latih status *spam* atau ham sudah ditentukan saat akan menambahkan data. Proses selanjutnya yaitu Proses *Smoothing Dirichlet* Naïve Bayes yang akan menghitung peluang setiap kombinasi kata yang terbentuk di Proses Kombinasi dengan menghitung nilai status *spam* dan ham. Menggunakan Teori Kombinasi dan Metode Naïve Bayes *Smoothing Dirichlet*, maka pengklarifikasian *email* ini diharapkan akan menjadi lebih optimal. Maka dari itu, perlu dilakukan analisis akurasi pengklarifikasian *email spam* dengan metode Teori Kombinasi dan Naïve Bayes *Smoothing Dirichlet*.

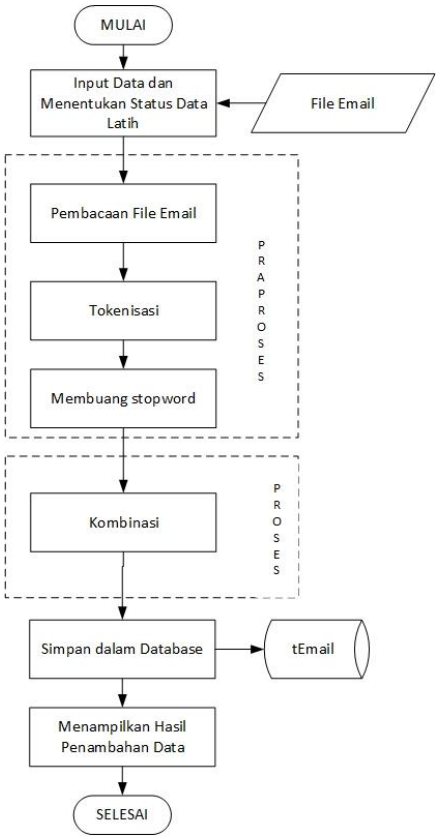
3.1.2. Analisis Sistem yang akan Dibangun

Tahapan yang akan dilakukan Sistem Klasifikasi *Email Spam* yaitu pada Gambar 3.1.



Gambar 3. 1 *Flowchart* Sistem Klasifikasi *Email Spam*

Dalam membangun Sistem Klasifikasi *Email Spam*, dilakukan beberapa tahapan. Tahapan penambahan data latih pada Sistem Klasifikasi *Email Spam* yaitu pada Gambar 3.2.



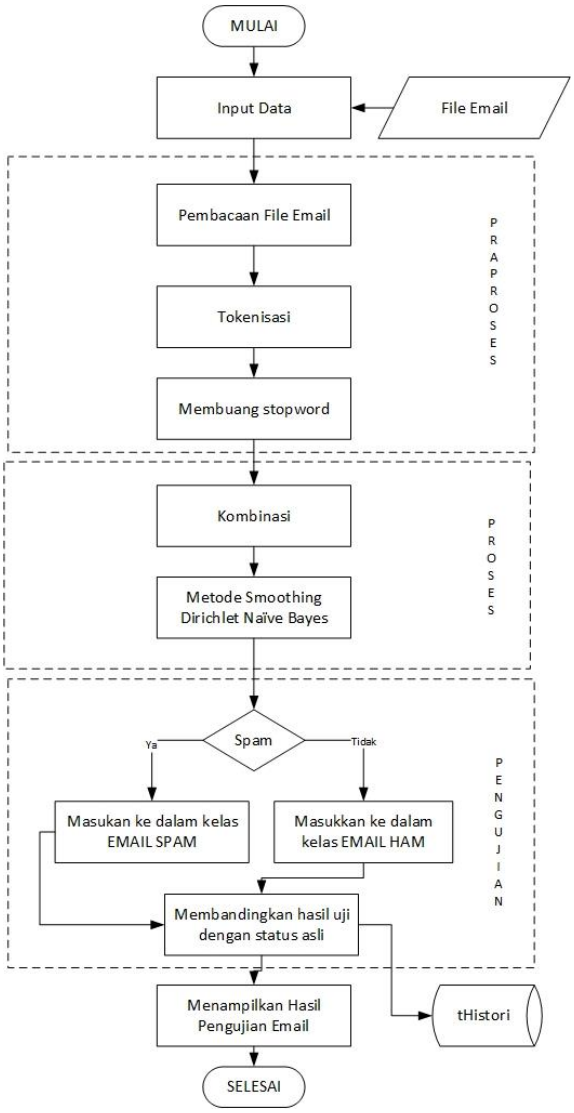
Gambar 3. 2 *Flowchart* Penambahan Data Latih pada Sistem

Berdasarkan Gambar 3.2, penambahan data latih pada sistem yang akan dibangun terbagi menjadi tahapan praproses dan proses. Proses pertama yaitu user memasukkan *file email* dengan cara *browse* data *email* dan menentukan status *email* yang akan ditambahkan sebagai data latih. Status *Email* yang dimiliki ada dua yaitu *email spam* dan *email ham* (*email* bukan *spam*).

Tahap selanjutnya yaitu tahap praproses, proses-proses ini menyiapkan data hingga data siap diolah pada proses utama. Proses-proses yang ada pada tahap ini yaitu: Pembacaan *file email*, Tokenisasi dan Membuang *Stopword*.

Proses selanjutnya adalah adalah Teori Kombinasi. Proses terakhir adalah proses penyimpanan hasil proses Teori Kombinasi ke dalam *database*.

Tahapan untuk pengujian *email spam* pada sistem klasifikasi *email spam* terdapat pada Gambar 3.3.



Gambar 3. 3 *Flowchart* Pengujian *Email* dalam Sistem

Berdasarkan Gambar 3.3, pengujian *email* pada sistem yang akan dibangun terbagi menjadi 3 tahapan yaitu praproses, proses dan pengujian. Proses pertama yaitu *user* memasukkan *file email* dengan cara *browse* data *email*. Sistem akan menyimpan direktori atau lokasi dari *file email* yang dimasukkan.

Tahap pertama yaitu tahap praproses, proses-proses ini menyiapkan data hingga data siap diolah pada proses utama. Proses-proses yang ada pada tahap ini yaitu: Pembacaan *file email*, Tokenisasi dan Membuang *Stopword*.

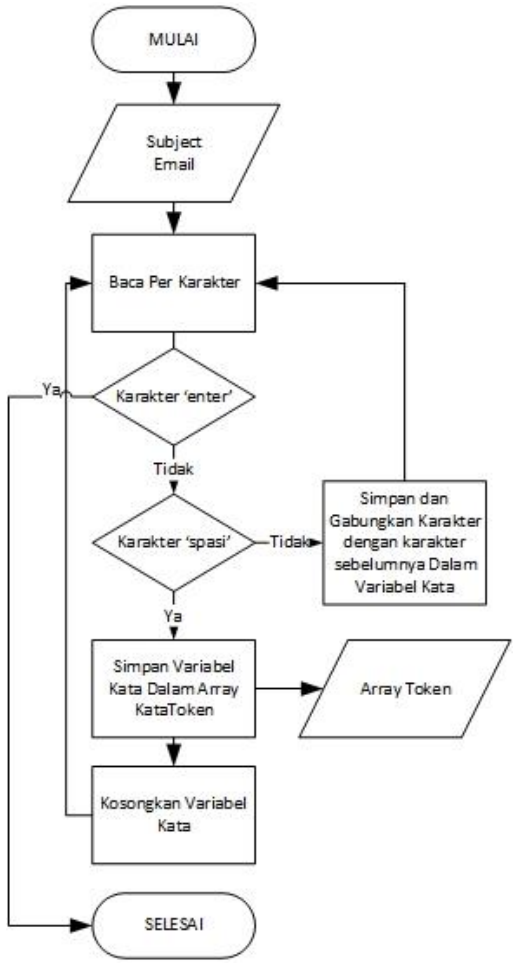
Tahap Proses selanjutnya adalah adalah proses utama dalam pengklasifikasian. Proses-proses yang ada pada tahap ini yaitu: Kombinasi kata dan Metode *Smoothing Dirichlet* Naïve Bayes.

Tahap Proses terakhir adalah proses pengujian. Pada tahap ini *email* yang diujikan sudah memiliki status *email spam* atau *email ham*. Proses selanjutnya yaitu sistem membandingkan status hasil pengujian dengan direktori atau lokasi *file email* berada, jika misalkan lokasi *email* adalah `C:\data*Email*\*spam*` dan hasil pengujian status *email* adalah *spam* maka pengujian status benar. Lalu menyimpan proses pengujian sesuai hasil status *Email* yang didapatkan pada proses sebelumnya.

Penjelasan untuk setiap tahap praproses yang dilakukan Sistem Klasifikasi *Email Spam* yaitu:

a. Pembacaan *file email* yaitu proses pembacaan *file email* yang sudah tersedia berupa teks. Sistem akan mencari semua elemen header dari *file email* yaitu *Mime-version, Form, Received, Date, Delivered-To, Message-Id, Subject, To* dan *Return-Path* sebagai informasi yang akan ditampilkan pada *user*. Data yang digunakan untuk tahap selanjutnya hanya *subject email* saja.
b. Tokenisasi yaitu proses pembentukan token per kata dari *subject email* yang akan dijadikan data latih. Proses tokenisasi melakukan pemotongan kata ketika menemukan spasi.

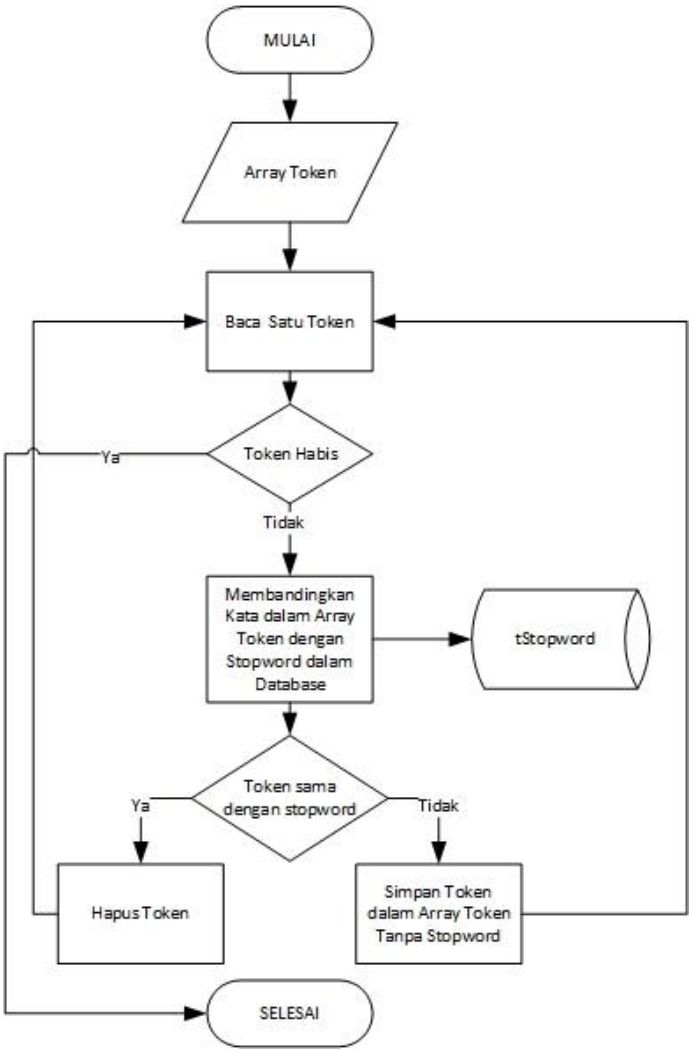
   Flowchart Tokenisasi terdapat pada Gambar 3.4.



Gambar 3. 4 *Flowchart* Tokenisasi

c. Membuang *Stopword* yaitu proses pembuangan karakter-karakter tertentu untuk mengoptimasikan dalam memunculkan token dari pembentuk *subject email* utamanya. Proses membuang *stopword* yaitu membandingkan token dengan daftar *stopword* yang ada pada *database.* Daftar stopword yang digunakan dalam penelitian ini didapat dari: http://jmlr.org/papers/volume5/lewis04a/a11-smart-stoplist/english.stop

Flowchart Membuang *Stopword* terdapat pada Gambar 3.5.



Gambar 3. 5 *Flowchart* Membuang *Stopword*

Penjelasan untuk setiap tahap proses utama yang dilakukan sistem klasifikasi *email spam* yaitu:

a. Teori Kombinasi yaitu proses pembentukan kombinasi kata dari *subject email*. Proses ini akan digunakan saat akan menambahkan data latih dan saat akan mengujikan suatu *email*.
b. Metode *Smoothing Dirichlet* Naïve Bayes yaitu proses menghitung kemungkinan *email* yang diujikan berupa *email spam* atau *ham* dari kombinasi kata yang sudah didapatkan pada proses Kombinasi.

3.1.3. Analisis Data Masukan

Data masukkan yang digunakan dalam proses pembelajaran dan pengujian agar sistem ini dapat berjalan yaitu:

a. Data yang digunakan adalah korpus *email* publik yang tersedia pada halaman *web Spam*assassin dengan kode prefix "20030228" yang berisi *folder*:
   1) easy_ham: berupa *email* ham
   2) easy_ham2: berupa *email* ham
   3) hard_ham: berupa *email* ham
   4) *spam*: berupa *email spam*
   5) *spam*_2: berupa *email spam*

b. Korpus yang digunakan adalah dokumen *email* dengan standar MIME dalam format *raw* sesuai dengan Tabel 3.1.

Tabel 3. 1 Tabel Komponen *Email* Standar MIME

Jenis Komponen	Nama Sintaks	Definisi Sintaks
Header	*MIME- version*	Menunjukan versi MIME yang digunakan
	Form	Nama dan alamat pengirim pesan
	Received	Daftar semua server / komputer dimana pesan dapat sampai ke penerimanya
	Date	Menunjukan tanggal dan waktu pesan*Email* dibuat
	Delivered-To	Alamat penerima *email*
	Message-ID	Sebuah string unik yang diberikan oleh sistem mail saat pesan tersebut pertama kali dibuat
	Subject	Subjek dari pesan
	To	Alamat yang digunakan untuk mengirim pesan
	X-Mailer	Aplikasi yang digunakan untuk mengirimkan pesan
	Return-Path	Alamat pengembalian pesan jika alamat penerima tidak ditemukan
Body	*Plain text*	Isi pesan dengan format penulisan dalam teks ASCII biasa
	HTML text	Isi pesan yang mengandung tag HTML
	Attachment	Informasi yang memberikan lampiran dari sebuah pesan.

3.1.4. Analisis Metode

Penelitian ini diadakan untuk membangun sebuah sistem analisis yang dapat mengklasifikasi *email spam* atau ham, dengan menggunakan Teori Kombinasi dan Metode *Smoothing Dirichlet* Naïve Bayes.

Data yang digunakan berasal dari korpus *email* Spamassassin dan akan digunakan sebagai data latih untuk menganalisis metode dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Data *Email* yang dijadikan data latih dalam Analisis

No	Messsage ID	*Subject*	Status Asli	Asal Folder
1	<4678c1042001882MM_CD@dd_it6>	Life with Insurance Pay	*Email* Spam	*/spam*
2	<008e45b47bde$5627d8c0$0ee03ad6@vjsxnm>	Life Pay for Insurance	*Email* Ham	*/hard_ham*
3	<200208262200.XAA28140@webnote.net>	[Ilug] No Subject	*Email* Spam	*/spam_2*
4	<200208270447.FAA29576@webnote.net>	New Sequence Windows	*Email* Ham	*/easy_ham*
5	<200208270036.g7R0aJp24458@l4.newnamedns.com>	List MacOSX Mailing?	*Email* Ham	*/easy_ham2*

Hasil dari proses penambahan data latih sesuai pada tahapan Gambar 3.3 dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Kombinasi Data Latih dalam Analisis

No	Kombinasi	Status
1	Life	*Spam*
2	Life	*Ham*
3	Insurance	*Spam*
4	Insurance	*Ham*
5	Pay	*Spam*
6	Pay	*Ham*
7	Life-Pay	Ham
8	Life-Pay	*Spam*
9	Life-Insurance	Ham
10	Life-Insurance	*Spam*
11	Insurance-Pay	*Spam*
12	Insurance-Pay	Ham
13	Life-Insurance-Pay	Ham
14	Life-Insurance-Pay	Spam
15	Ilug	*Spam*
16	No	*Spam*
17	Subject	*Spam*
18	Ilug-No	*Spam*
19	No-Subject	*Spam*
20	Ilug-Subject	*Spam*
21	New	Ham
22	Sequence	Ham

No	Kombinasi	Status
23	Window	Ham
24	New-Sequence	Ham
25	New-Window	Ham
26	Sequence-Window	Ham
27	MacOSX	Ham
28	Mailing	Ham
29	List	Ham
30	MacOSX-Mailing	Ham
31	List-MacOSX	Ham
32	List-Mailing	Ham
33	List-MacOSX-Mailing	Ham

Data uji yang akan dikaji pada sistem yang akan dibangun berasal dari korpus *email* Spamassassin dengan direktori atau lokasi *file email* `C:\data`*`Email`*`\`*`spam`* yaitu pada Tabel 3.4.

Tabel 3. 4 Data Yang Akan Diujikan dalam Analisis

No	Messsage ID	*Subject*	Status Asli
1	<0103c1042001882DD_IT7@dd_it7>	Life Insurance Why Pay More?	Email Spam

Tahap pengujian yang dilakukan sesuai dengan tahapan pada Gambar 3.4. Tahap pertama yang akan dilakukan adalah tahap praproses. Tahap Praproses yang dilakukan adalah sebagai berikut:

a. Pembacaan *File Email*

   Pembacaan *file email* untuk mendapatkan bagian *email* yang akan dimasukkan dalm proses tokenisasi. Isi dari sebuah dokumen *email* terdiri dari *header* dan diikuti dengan *body* (opsional). Pada Tabel 3.1 menampilkan komponen *header* dan *body* menurut *Multipurpose Internet Mail Extensions* (MIME), sedangkan komponen *header* yang digunakan untuk mendapatkan ekstraksi dokumen *email* dalam penelitian adalah *subject* saja.

b. Tokenisasi

   Tokenisasi merupakan tahap praproses yang membuat kata penyusun *subject email* dibuat menjadi token-token kata per kata. Simbol-simbol yang ada pada *subject* akan diabaikan untuk sistem ini. Dari data yang akan diujikan maka token-token yang muncul adalah pada Tabel 3.5.

Tabel 3. 5 *List* Token Hasil Proses Tokenisasi

No	Token
1	Life
2	Insurance
3	Why
4	Pay
5	More

c. Membuang *Stopword*

Tahap praproses selanjutnya adalah membuang *stopword*. *Stopword* yang sudah terdaftar dapat dilihat di lampiran E. Maka token dengan kata '*Why'* dan '*More'* akan dihapuskan sehingga tersisa token-token pada Tabel 3.6.

Tabel 3. 6 *List* Token Setelah Membuang *Stopword*

No	Token
1	Life
2	Insurance
3	Pay

Tahap kedua yang dilakukan adalah tahap proses. Tahap proses akan membaca dokumen data latih terlebih dahulu lalu kembali ke proses pengujian. Tahap proses yaitu sebagai berikut:

a. Teori Kombinasi

Pada proses ini token yang sudah didapatkan pada tahap praproses dibentuk menjadi kombinasi-kombinasi kata. Pembentukan kombinasi dimulai dari jumlah anggota sama dengan satu.

Iterasi pengombinasian berhenti saat jumlah anggota kombinasi sama dengan jumlah token dari *subject* yang diujikan. Dalam pengujian ini pengombinasian berhenti setelah iterasi sama dengan tiga karena jumlah token yang didapatkan dari tahap praproses berjumlah tiga.

Kombinasi dengan jumlah anggota sama dengan satu (C_1) diproses karena satu kurang dari sama dengan tiga, sehingga hasilnya yaitu

$$C_1 = \{\{Life\}, \{Insurance\}, \{Pay\}\}$$

Kombinasi dengan jumlah anggota dua (C_2) diproses karena dua kurang dari sama dengan tiga, sehingga hasilnya didapat dengan memasangkan tiap token ke token lainnya sehingga didapatkan kombinasi

$$C_2 = \{\{Insurance, Life\}, \{Life, Pay\}, \{Insurance, Pay\}\}$$

Kombinasi dengan jumlah anggota tiga (C_3) diproses karena tiga kurang dari sama dengan tiga, sehingga hasilnya didapat dengan memasangkan tiap dua token ke token lainnya sehingga didapatkan kombinasi

$$C_3 = \{\{Insurance, Life, Pay\}\}$$

Hasil dari Proses Kombinasi adalah sebagai Himpunan Kombinasi Kata sebagai berikut:

$$C = \{\{Life\}, \{Insurance\}, \{Pay\}, \{Insurance, Life\}, \{Life, Pay\}, \{Insurance, Pay\}, \{Insurance, Life, Pay\}\}$$

b. Metode *Smoothing Dirichlet* Naïve Bayes

Perhitungan peluang setiap menggunakan Metode *Smoothing* masih menggunakan kaidah Naïve Bayes, namun berbeda pada persamaan penugaan parameter $\acute{P}(t|c)$. Yang akan digunakan pada Metode *Smoothing Dirichlet* yaitu Rumus 2.3 dan Rumus 2.4.

Maka dari kombinasi kata yang didapatkan dari data uji, maka perhitungan untuk perkiraan *email spam* (c = *spam*) dilihat dari data latih pada Tabel 3.3 untuk menghitung nilai peluang *Smoothing Dirichlet* Naïve Bayes. Nilai μ yang digunakan dalam penelitian ini adalah nilai μ yang memiliki nilai akurasi tertinggi dalam penelitian Mutia Hafilizara, yaitu dengan $\mu = 0{,}1$. Hasil perhitungan pada Tabel 3.7.

Tabel 3. 7 Proses Perhitungan *Smoothing* Naive Bayes untuk *Email spam*

No	Kombinasi	$P(t\|C)$	$P\mu(t\|c)$
1	Life	$1/_{13} = 0{,}08$	(1+0,1(0,08))/(2+0,1) = 0,48
2	Insurance	$1/_{13} = 0{,}08$	(1+0,1(0,08))/(2+0,1) = 0,48
3	Pay	$1/_{13} = 0{,}08$	(1+0,1(0,08))/(2+0,1) = 0,48
4	Insurance –Life	$1/_{13} = 0{,}08$	(1+0,1(0,08))/(2+0,1) = 0,48

No	Kombinasi	$P(t\|C)$	$P\mu(t\|c)$
5	Life-Pay	$1/13 = 0{,}08$	(1+0,1(0,08))/(2+0,1) = 0,48
6	Insurance-Pay	$1/13 = 0{,}08$	(1+0,1(0,08))/(2+0,1) = 0,48
7	Life-Insurance-Pay	$1/13 = 0{,}08$	(1+0,1(0,08))/(2+0,1) = 0,48
		$\sum P\mu(t\|c)$	3,36

Dari Tabel 3.7 dapat diperkirakan untuk *email* yang diujikan sebagai *spam* memiliki nilai rata-rata

$$Nilai\ Spam = \frac{\sum P\mu(t|c)}{Jumlah\ rule\ email\ yang\ diujikan} = \frac{3{,}36}{7} = 0{,}48$$

Perkiraan *email ham* (c = *ham*) dilihat dari data latih pada Tabel 3.3 untuk menghitung nilai peluang *Smoothing Dirichlet* Naïve Bayes. Nilai μ yang digunakan dalam penelitian ini adalah nilai μ yang memiliki nilai akurasi tertinggi dalam penelitian Mutia Hafilizara, yaitu dengan $\mu = 0{,}1$. Hasil perhitungan pada Tabel 3.8.

Tabel 3. 8 Proses Perhitungan Smoothing Naive Bayes untuk *Email* ham

No	Kombinasi	$P(t\|C)$	$P\mu(t\|c)$
1	Life	$1/20 = 0{,}05$	(1+0,1(0,05))/(2+0,1) = 0,478
2	Insurance	$1/20 = 0{,}05$	(1+0,1(0,05))/(2+0,1) = 0,478
3	Pay	$1/20 = 0{,}05$	(1+0,1(0,05))/(2+0,1) = 0,478
4	Insurance –Life	$1/20 = 0{,}05$	(1+0,1(0,05))/(2+0,1) = 0,478
5	Life-Pay	$1/20 = 0{,}05$	(1+0,1(0,05))/(2+0,1) = 0,478
6	Insurance-Pay	$1/20 = 0{,}05$	(1+0,1(0,05))/(2+0,1) = 0,478
7	Life-Insurance-Pay	$1/20 = 0{,}05$	(1+0,1(0,05))/(1+0,1) = 0,478
		$\sum P\mu(t\|c)$	3,346

Menurut Tabel 3.8 dapat diperkirakan untuk *email* yang diujikan sebagai *ham* memiliki nilai rata-rata

$$Nilai\ Ham = \frac{\sum P\mu(t|c)}{Jumlah\ rule\ email\ yang\ diujikan} = \frac{3{,}346}{7} = 0{,}478$$

Tahap Ketiga dari sistem yaitu proses pengujian. Pengujian akan membaca nilai *spam* dan nilai ham pada data yang diujikan. Jika nilai *spam* lebih besar dari

nilai ham maka *email* diklasifikasikan sebagai *email spam* dan jika nilai ham lebih besar dari nilai *spam* maka *email* diklasifikasikan sebagai *email* ham.

Dari data yang diujikan, nilai *spam* memiliki nilai 0,48 dan nilai ham memiliki nilai 0,478. Hasil pengujian yang diambil adalah yang nilai paling besar yaitu nilai untuk ham. Dengan kesimpulan data *email* yang diujikan adalah *email spam*. Proses selanjutnya adalah membandingkan hasil dengan direktori atau lokasi *file email*. Hasil dari pengujian sebelumnya adalah *email spam* dan direktorinya adalah `C:\data`*`Email\spam`*, maka hasil pengujian dinyatakan benar karena *email* yang diujikan adalah *spam*.

3.1.5. Analisis Kebutuhan Non Fungsional

Pada tahap ini meliputi elemen-elemen yang diperlukan dalam membuat Sistem Klasifikasi *Email Spam*. Kebutuhan non fungsional terbagi menjadi tiga yaitu analisis perangkat keras, analisis perangkat lunak dan analisis pengguna.

3.1.5.1. Analisis Kebutuhan Perangkat Keras

Perangkat keras atau *hardware* salah satu hal yang penting karena tanpa *hardware* yang memenuhi syarat, program yang akan dibuat tidak akan dapat berjalan. Berikut spesifikasi minimal perangkat keras yang dipergunakan untuk membangun Sistem Klasifikai *Email Spam* ini dan spesifikasi perangkat keras yang dapat dipergunakan untuk menjalankan sistem ini yaitu pada Tabel 3.9.

Tabel 3. 9 Tabel Kebutuhan Minimal Perangkat Keras

No.	Nama Perangkat Keras	Spesifikasi
1.	Processor	2.10GHz
2.	RAM	2 GB
3.	VGA	1 GB
4.	Hardisk Drive	50 GB

3.1.5.2. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak atau *software* salah satu hal yang terpenting dalam mendukung kinerja sebuah sistem. Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi diantara keduanya. Sedangkan perangkat lunak yang dapat

dibutuhkan untuk menjalankan Sistem Klasifikasi *Email Spam* ini yaitu pada Tabel 3.10.

Tabel 3. 10 Tabel Kebutuhan Minimal Perangkat Lunak

No.	Nama Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows XP, Windows 7
2.	*Browser*	*Mozilla Firefox, Google Chrome*
3.	*Compiler*	*Microsoft Visual Studio 2012*
4	Bahasa Pemerograman	*C#*
5	*Web Server*	*Apache(XAMPP)*
6	*Database*	*MySQL*

3.1.5.3. Analisis Pengguna

Pengguna atau *brainware* salah satu hal yang terpenting dalam mendukung kinerja sebuah sistem. *Brainware* digunakan sebagai orang yang menggunakan sistem yang dibangun. Spesifikasi pengguna sistem terdapat pada Tabel 3.11.

Tabel 3. 11 Tabel Analisis Pengguna

Tingkat Pengalaman Pengguna	
Kecakapan Menggunakan Komputer	Pengalaman menggunakan Komputer/*notebook*
Pengalaman Menggunakan Aplikasi	Pernah meggunakan internet dan aplikasi *browser*
Edukasi	User mengetahui istilah-istilah dalam *service*, komponen perangkat keras
Tingkat Baca	Teknisi
Kemampuan Mengetik	*Novice*
Bahasa yang digunakan	Untuk sistem menggunakan Bahasa Indonesia dan *email* yang digunakan Bahasa Inggris
Handedness	Kanan dan kiri

3.1.6. Analisis Kebutuhan Fungsional

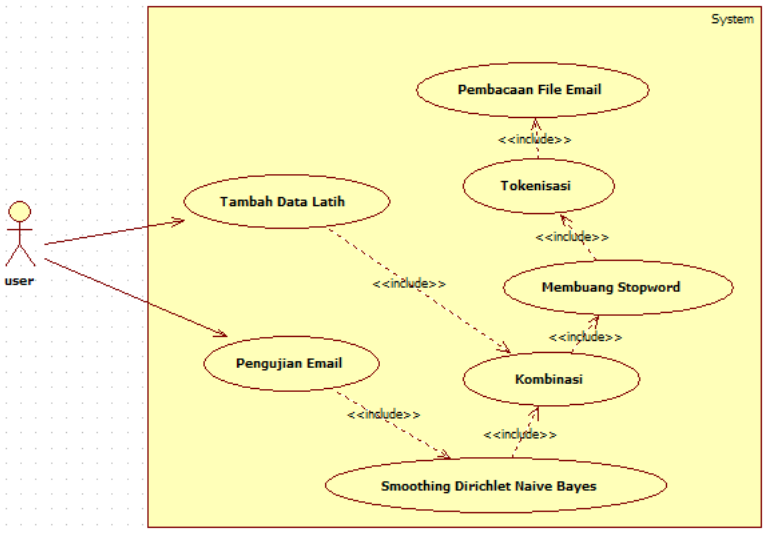
Identifikasi aktor dapat dilakukan dalam analisis berorientasi objek dengan menggunakan UML yaitu menentukan aktor atau pengguna sistem. Aktor dalam konteks UML menampilkan peran pengguna atau sesuatu diluar sistem yang dikembangkan dapat berupa perangkat keras, *end user*, sistem yang lain dan sebagainya.

3.1.6.1. Indentifikasi Aktor

Aktor yang teridentifikasi dan terlibat dalam sistem hanya ada satu, yaitu *user*.

3.1.6.2. *Use Case Diagram*

Use Case adalah interaksi antara sistem dan aktor yang terlibat, didalamnya termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem. *Use Case* dari sistem yang akan dibangun terdapat pada Gambar 3.6.



Gambar 3. 6 *Use Case Diagram* Sistem Klasifikasi *Email Spam*

3.1.6.3. *Scenario Diagram*

Scenario Diagram menjelaskan skenario dari setiap proses yang terdapat pada *use case diagram*. Berikut ini *scenario diagram* dari *use case* berdasarkan *use case* yang akan dibangun:

Tabel 3. 12 Tabel *Scenario Diagram* Pembacaan *File Email*

Use Case	*Pembacaan File Email*
Actor	User
Goals	Membaca seluruh elemen header dari *Email*
Pre-conditions	User sudah memilih file *Email* yang akan ditambahkan menjadi data latih atau file *Email* yang akan diujikan
Post-conditions	Sistem menyimpan elemen *header* dari *Email*
Trigger	Adanya inputan *File Email* yang sudah dipilih dan dikonfirmasi user
Description	Pembacaan *File Email* ini bertujuan untuk membaca dan mendapatkan informasi *Email* yang akan ditambahkan sebagai data latih atau *Email* yang akan diujikan

Main Flow	**Step**	***Action***
	1	User sudah memilih data *Email*
	2	Sistem membaca dan menyimpan file elemen header dari *Email*
	3	Sistem memilih *subject Email* untuk diproses ke tahap selanjutnya

Tabel 3. 13 Tabel *Scenario Diagram* Tokenisasi

Use Case	*Tokenisasi*	
Actor	User	
Goals	Kata dari *subject Email* menjadi token-token	
Pre-conditions	Mendapatkan isi *subject Email* dari proses sebelumnya	
Post-conditions	*Subject Email* dijadikan token-token kata	
Trigger	Proses Pembacaan File *Email* telah selesai dilakukan	
Description	Proses Tokenisasi adalah proses menjadikan token-token perkata dari *subject Email*	
Main Flow	**Step**	***Action***
	1	Sistem membaca *subject Email* dari hasil proses pembacaan *Email*
	2	Mengubah *subject Email* menjadi token-token perkata
	3	Menyimpan token-token perkata

Tabel 3. 14 Tabel *Scenario Diagram* Membuang *Stopword*

Use Case	*Membuang Stopword*	
Actor	User	
Goals	Membuang semua *stopword* yang terdapat dalam token-token	
Pre-conditions	Sistem sudah memiliki token-token kata *subject Email* dari Proses Tokenisasi	
Post-conditions	Memiliki token-token yang sudah tidak ada *stopword*-nya	
Trigger	Token-token dari *subject Email* telah ditentukan	
Description	Proses Membuang *Stopword* merupakan proses membuang token kata yang termasuk stopword	
Main Flow	**Step**	***Action***
	1	Sistem memiliki token-token kata dari *subject Email*
	2	Menghapus token yang sesuai dengan *stopword* yang terdapat dalam *database*
	3	Mendapatkan token yang sudah tidak ada *stopword*-nya

Tabel 3. 15 Tabel *Scenario Diagram* Kombinasi

Use Case	*Kombinasi*
Actor	User
Goals	Mempunyai daftar kombinasi yang dibentuk dari *subject Email* yang dimasukkan
Pre-conditions	User memilih data *Email* yang akan diujikan atau data yang akan dijadikan data latih
Post-conditions	Sistem menyimpan himpunan kombinasi kata dari *subject Email*
Trigger	Proses pembuangan *stopword* telah dilakukan
Description	Proses Teori Kombinasi merupakan proses pembentukan kombinasi-kombinasi kata dari *subject Email* yang selanjutnya akan dihitung dalam proses *Smoothing Dirichlet* Naïve Bayes

Main Flow	**Step**	***Action***
	1	Sistem memasukkan token/kata hasil Proses Membuang *Stopword* ke Proses Teori Kombinasi
	2	Mengkombinasikan setiap token/kata
	3	Menghasilkan himpunan kombinasi kata

Tabel 3. 16 Tabel *Scenario Diagram* Tambah Data Latih

Use Case	*Tambah Data Latih*	
Actor	User	
Goals	User menambahkan data latih untuk pengklarifikasian *Email*	
Pre-conditions	User memilih data *Email* yang akan ditambah jadi data latih	
Post-conditions	Sistem menambahkan data latih sesuai masukkan usir	
Trigger	User melakukan *request* penambahan data latih	
Description	Penambahan data latih ini digunakan agar sistem dapat mendeteksi *Email spam* sesuai kriteria yang diinginkan user	
Main Flow	**Step**	***Action***
	1	Sistem menampilkan menu pilihan
	2	User memilih menu tambah data latih
	3	Sistem menampilkan menu tambah data latih
	4	User memilih data yang akan ditambahkan sebagai data latih dan menentukan status *Email*.
	5	Sistem melakukan praproses (Pembacaan *File Email*, Tokenisasi dan Membuang *Stopword*)
	6	Sistem menampilkan preview *Email* yang dipilih
	7	User mengkonfirmasi *Email* yang dipilih sebagai data latih
	8	Sistem melakukan proses Kombinasi dari *Subject Email* yang akan ditambahkan sebagai data latih
	9	Sistem menyimpan list kombinasi di database

Tabel 3. 17 Tabel *Scenario Diagram Smoothing Dirichlet* Naïve Bayes

Use Case	*Smoothing Dirichlet Naïve Bayes*	
Actor	User	
Goals	Memberikan nilai *spam* dan ham ke setiap kombinasi	
Pre-conditions	Sudah memiliki list kombinasi kata yang akan dihitung peluangnya	
Post-conditions	Sistem memiliki nilai *spam* dan ham untuk dibandingkan	
Trigger	User melakukan request pengujian *Email* dan sistem telah selesai melakukan proses Kombinasi	
Description	*Smoothing Dirichlet* Naïve Bayes merupakan proses pemberian nilai *spam* dan ham menggunakan rumus Naïve Bayes *Smoothing Dirichlet*.	
Main Flow	**Step**	***Action***
	1	Sistem memasukan list himpunan kombinasi kata ke proses *Smoothing Dirichlet* Naïve Bayes
	2	Menghitung nilai *spam* dan ham untuk setiap kombinasi kata
	3	Membandingkan nilai terbesar antara nilai *spam* dan ham
	4	Mengambil hasil dari nilai terbesar

Tabel 3. 18 Tabel *Scenario Diagram* Pengujian *Email*

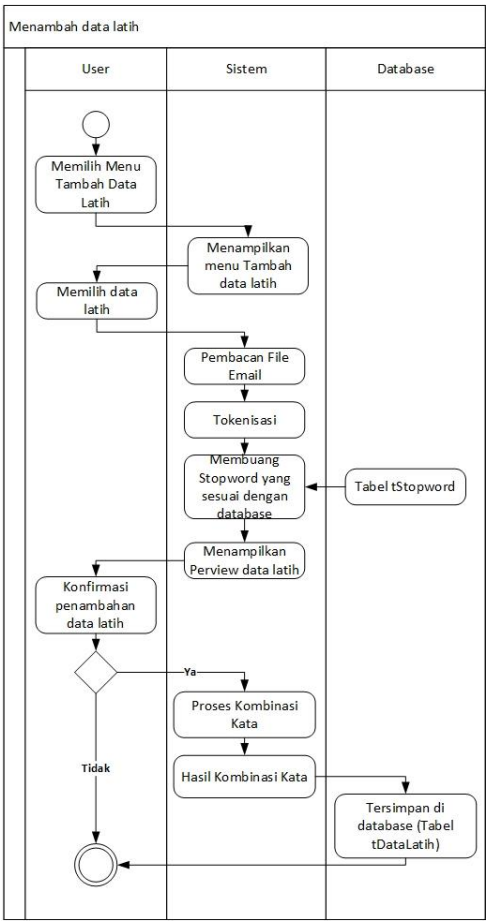
Use Case	*Pengujian Email*	
Actor	User	
Goals	Dapat mengetahui data *Email* yang dipilih merupakan *spam* atau ham	
Pre-conditions	User ingin menguji *Email* termasuk *spam* atau ham	
Post-conditions	Sistem menampilkan hasil proses *Email* yang dipilih merupakan *Email spam* atau *Email* ham	
Trigger	User melakukan *request* pengujian data *Email*	
Description	Menu ini digunakan untuk menguji atau mengetahui apakah *Email* yang diujikan merupakan *spam* atau ham	
Main Flow	**Step**	***Action***
	1	Sistem menampilkan menu pilihan
	2	User memilih menu pengujian *Email*
	3	Sistem menampilkan menu pengujian *Email*
	4	User memilih den memasukkan *Email* yang akan diuji
	5	Sistem melakukan praproses pengujian (Pembacaan *File Email*, Tokenisasi dan Membuang *Stopword*)
	6	Sistem menampilkan preview *Email* yang dipilih
	7	User mengkonfirmasi pengecekan *Email*
	8	Sistem mengolah *Email* (Kombinasi Kata dan Naïve Bayes)
	9	Sistem menampilkan hasil dari proses pengolahan dan klasifikasi *Email* yang diujikan

3.1.6.4. *Activity Diagram*

Activity diagram merupakan gambaran berbagai aliran aktivitas yang terjadi dalam apliaksi yang sedang dirancang. Berikut ini beberapa activity diagram yang terdapat pada Sistem Klasifikasi *Email Spam* yaitu :

a. *Activity Diagram* Menambahkan Data Latih

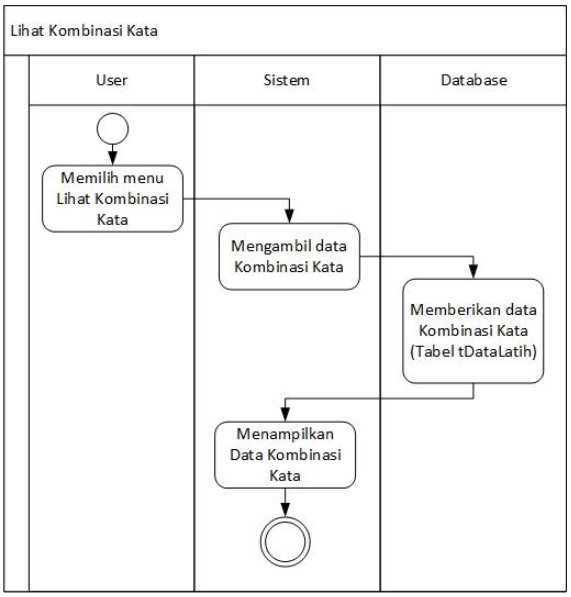
   Aliran Aktivitas yang terjadi saat user menambahkan data latih adalah pada Gambar 3.7.



Gambar 3. 7 *Activity Diagram* Tambah Data Latih

b. *Activity Diagram* Melihat Kombinasi Kata

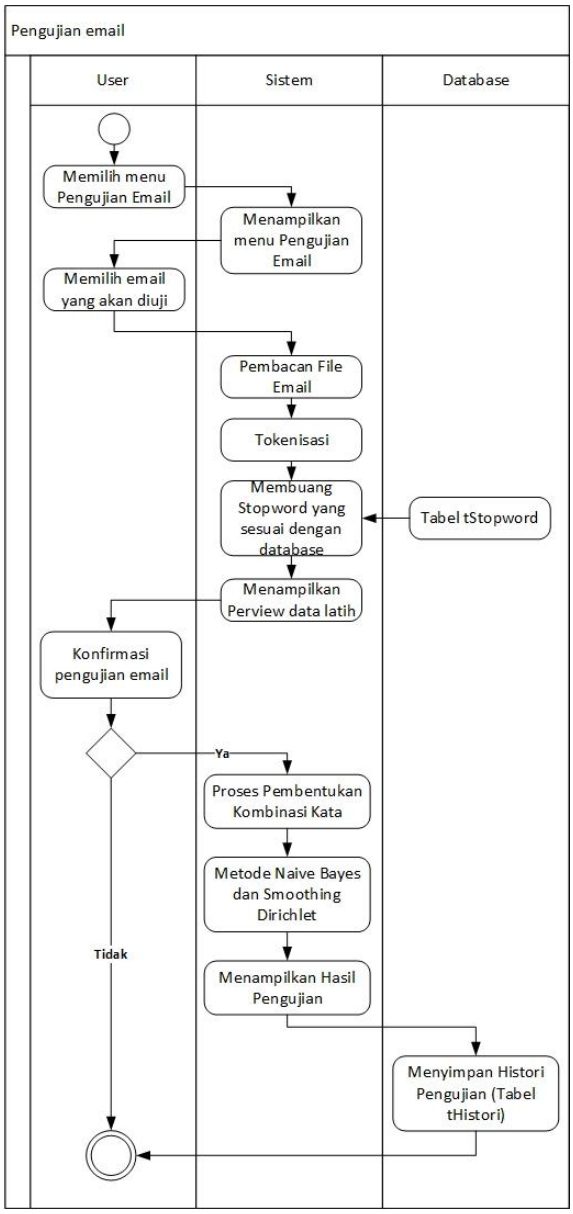
Aliran Aktivitas yang terjadi saat user melihat kombinasi kata adalah pada Gambar 3.8.



Gambar 3. 8 *Activity Diagram* Lihat Kombinasi Kata

c. *Activity Diagram* Pengujian *Email*

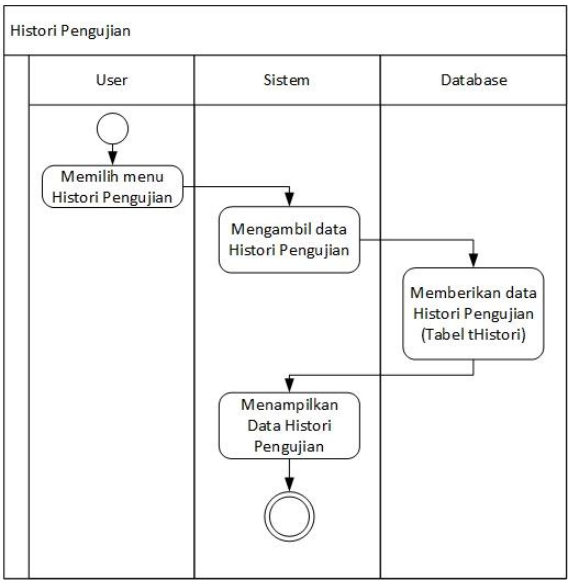
Aliran Aktivitas yang terjadi saat user pengujian *email* adalah pada Gambar 3.9.



Gambar 3. 9 *Activity Diagram* Pengujian *Email*

d. Diagram *Activity* Histori Pengujian

Aliran Aktivitas yang terjadi saat user memilih menu histori pengujian adalah pada Gambar 3.10.

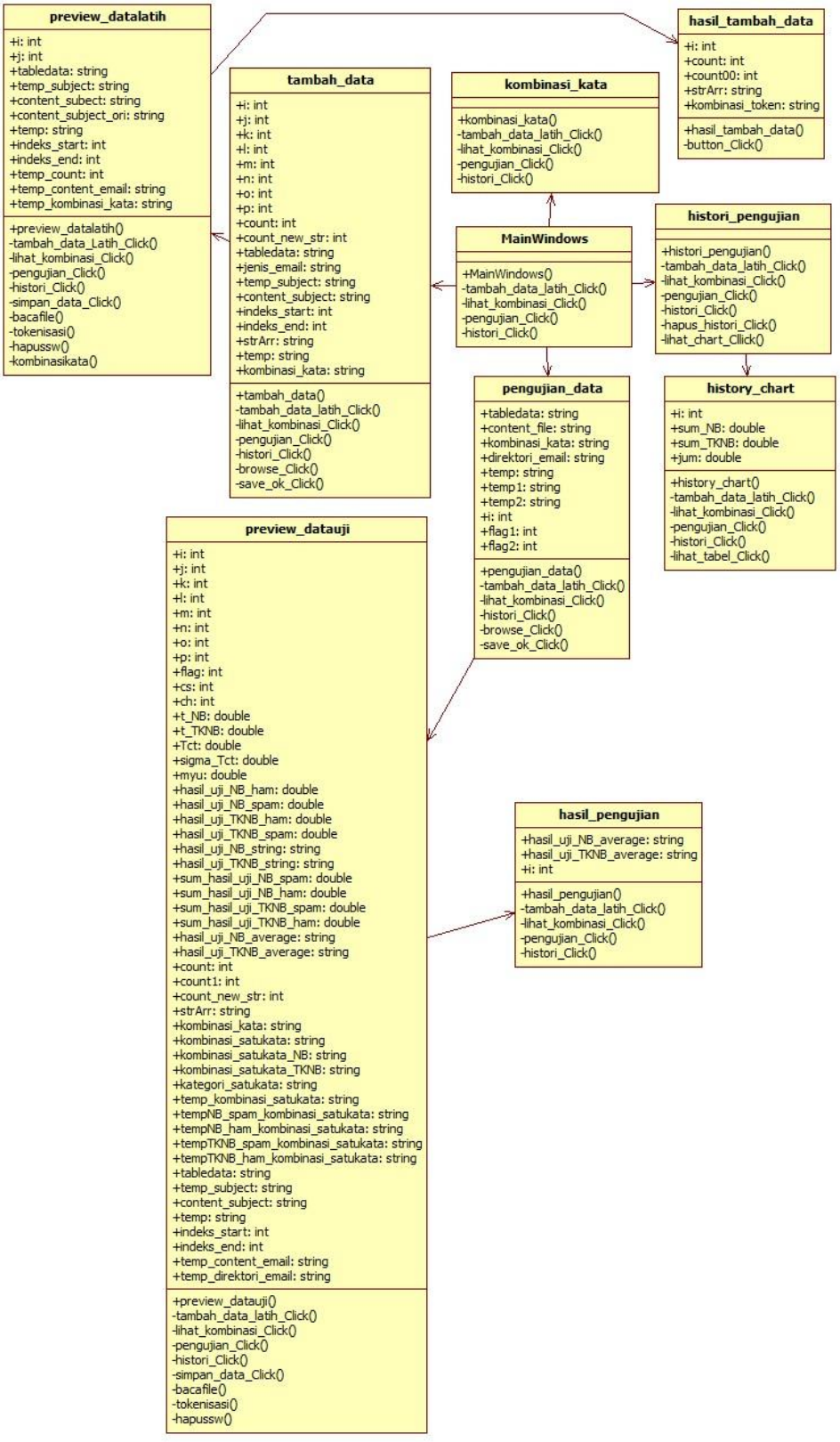


Gambar 3. 10 *Activity Diagram* Histori Pengujian

3.1.6.5. *Class Diagram*

Class Diagram adalah gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukan class-class objek yang menyusun sebuah sistem dan juga hubungan antara class objek tersebut.

Class Diagram dari sistem yang dibangun terdapat pada Gambar 3.11.



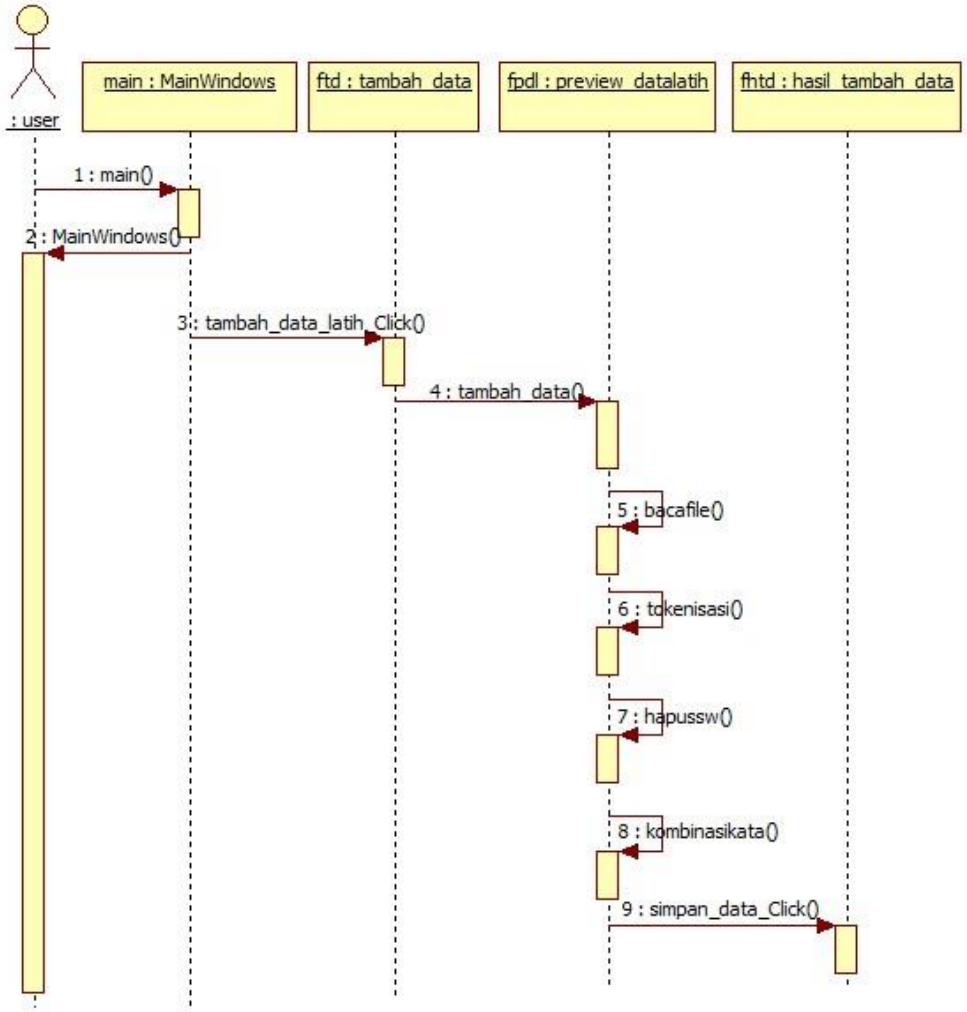
Gambar 3. 11 *Class Diagram* Sistem Klasifikasi *Email Spam*

3.1.6.6. *Sequence Diagram*

Sequence Diagram adalah diagram yang menggambarkan kelakuan atau aliran kerja objek pada use case dengan mendiskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

3.1.6.6.1. *Sequence Diagram* Tambah Data Latih

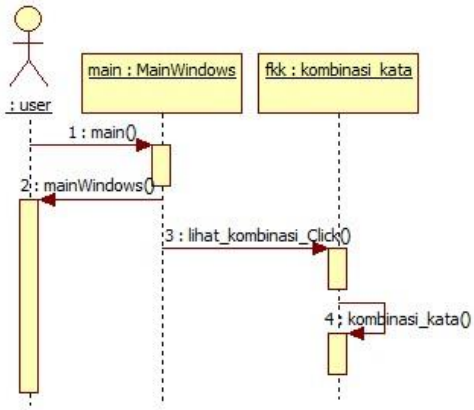
Berikut adalah *sequence diagram* untuk Tambah Data Latih pada Gambar 3.12.



Gambar 3. 12 *Sequence Diagram* Tambah Data Latih

3.1.6.6.3. *Sequence Diagram* Lihat Kombinasi Kata

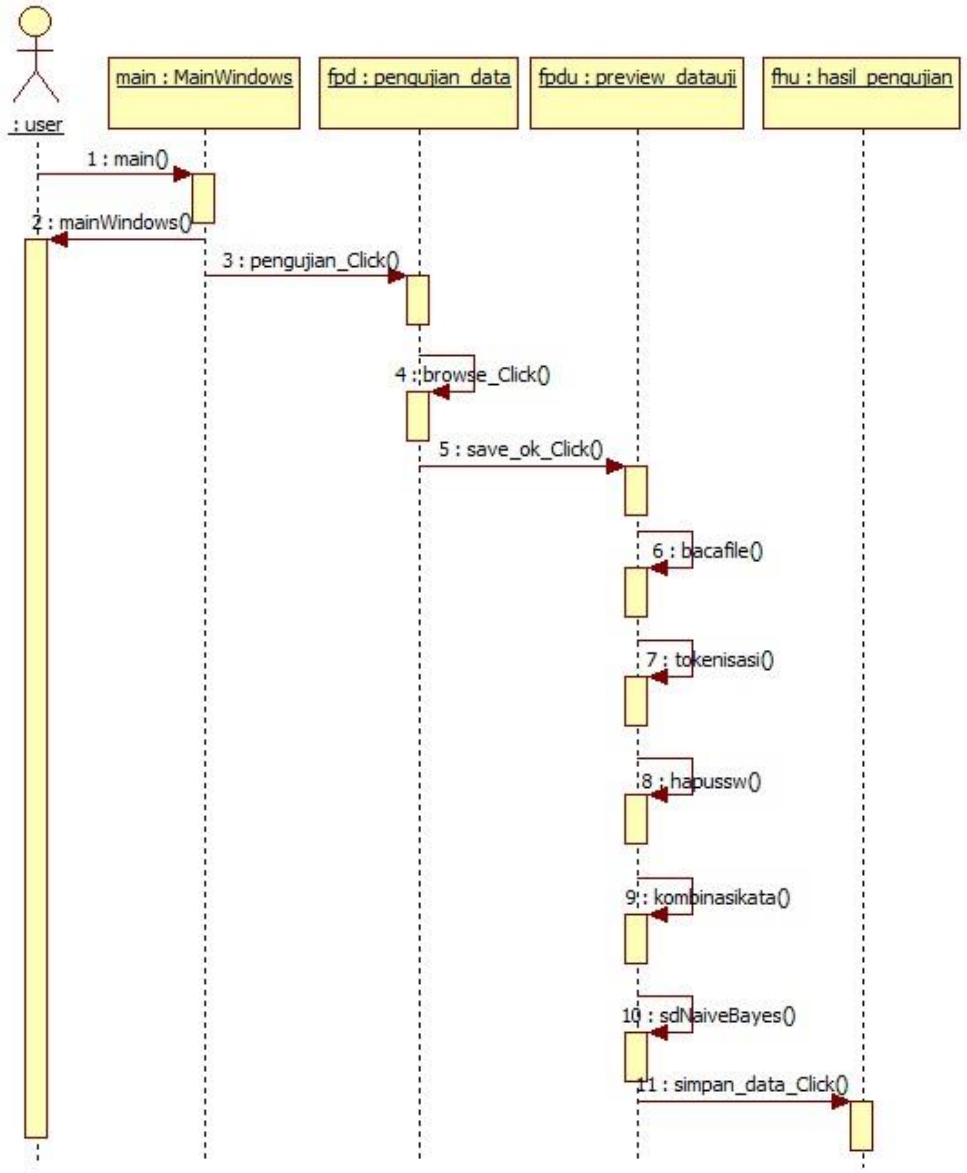
Sequence Diagram untuk Lihat Kombinasi Kata dapat dilihat pada Gambar 3.13.



Gambar 3. 13 *Sequence Diagram* Lihat Kombinasi Kata

3.1.6.6.4. *Sequence Diagram* Pengujian *Email*

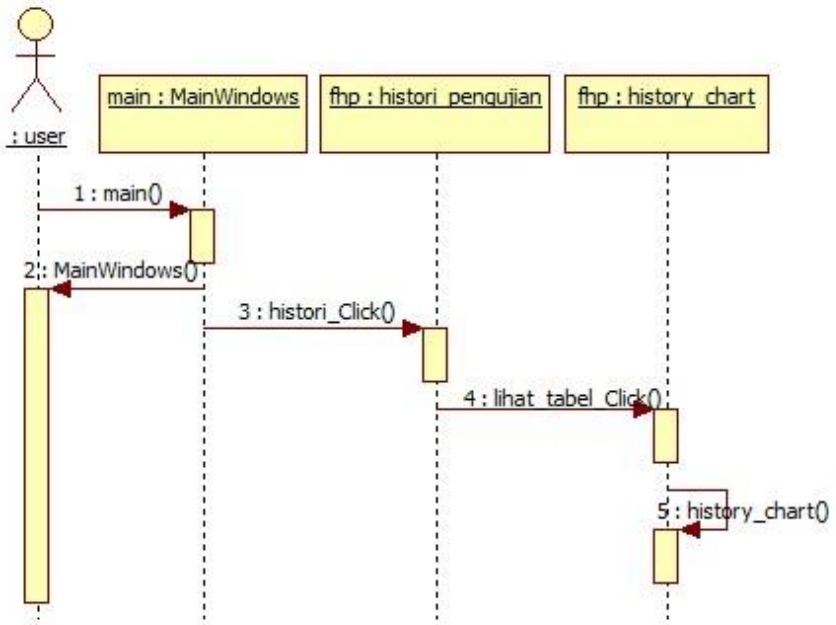
Sequence Diagram untuk Pengujian *Email* dapat dilihat pada Gambar 3.14.



Gambar 3. 14 *Sequence Diagram* Pengujian *Email*

3.1.6.6.5. *Sequence Diagram* Histori Pengujian

S*equence Diagram* untuk Histori Pengujian dapat dilihat pada Gambar 3.15.



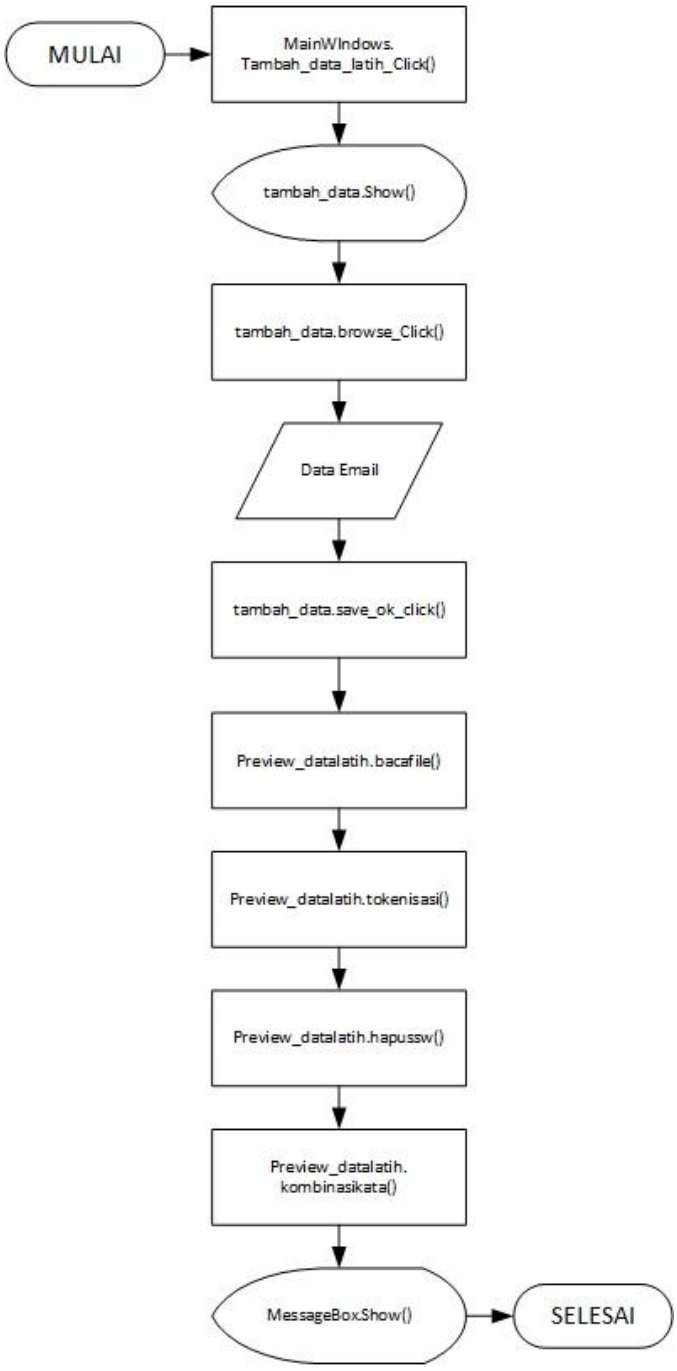
Gambar 3. 15 *Sequence Diagram* Histori Pengujian

3.1.6.7. Perancangan Metode

Merupakan kumpulan metode yang ada pada sistem yang dirancang dan berfungsi untuk menjalankan sistem. Perancangan method yang ada dalam sistem :

3.1.6.7.1. Perancangan Metode Menambah Data Latih

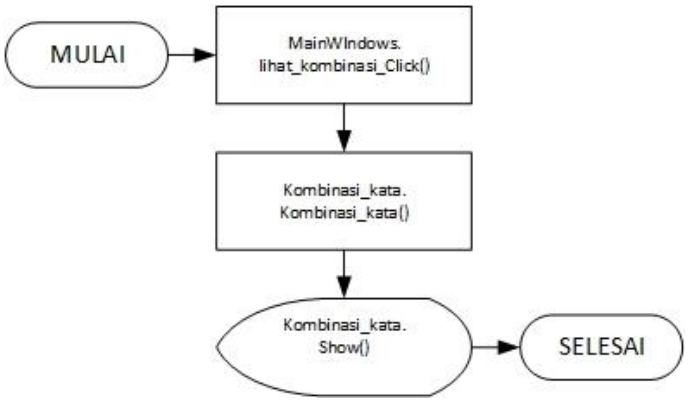
Perancangan Metode Menambah Data Latih dapat dilihat pada Gambar 3.16.



Gambar 3. 16 Perancangan Metode Tambah Data Latih

3.1.6.7.2. Perancangan Metode Melihat Data Latih

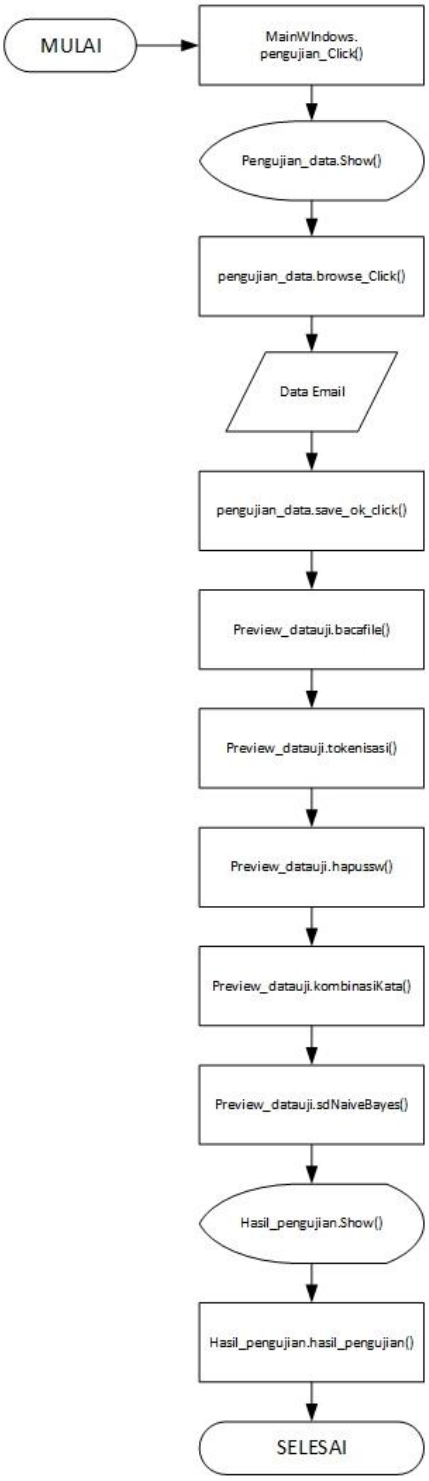
Perancangan Metode Melihat Data Latih dapat dilihat pada Gambar 3.17.



Gambar 3. 17 Perancangan Metode Lihat Data Latih

3.1.6.7.3. Perancangan Metode Pengujian *Email*

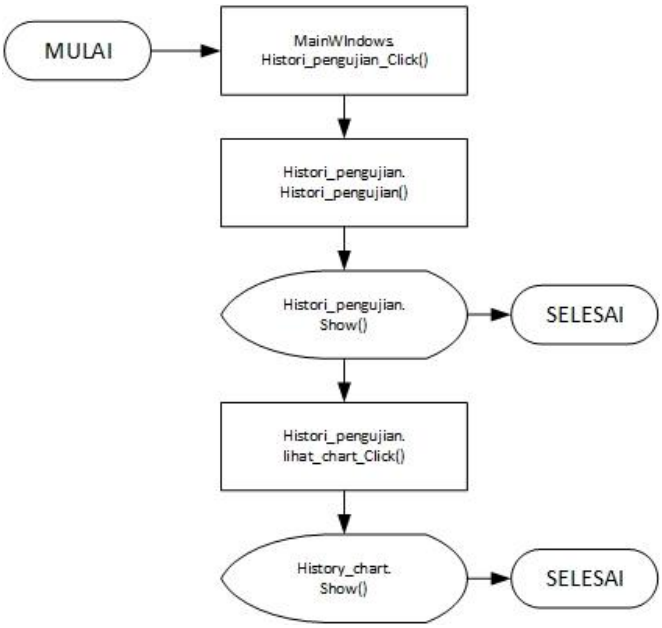
Perancangan Metode Pengujian *Email* dapat dilihat pada Gambar 3.18.



Gambar 3. 18 Perancangan Metode Pengujian *Email*

3.1.6.7.4. Perancangan Metode Melihat Histori Pengujian

Perancangan Metode Melihat Histori Pengujian dapat dilihat pada Gambar 3.19.



Gambar 3. 19 Perancangan Metode Lihat Histori Pengujian

3.1.7. Analisis Basis Data

Analisis Basis Data merupakan pendefinisan dengan lengkap data yang ada dalam sistem, baik itu masukkan, proses atau hasil dari suatu proses.

3.1.7.1. Kamus Data

Kamus Data dapat mendefinisikan dengan lengkap data yang mengalir diantara proses, penyimpanan data dan entitas luar pada sistem. Data yang mengalir tersebut dapat berupa masukan untuk sistem atau hasil dari proses sistem. Kamus Data tersebut terdapat pada Tabel 3.19 :

Tabel 3. 19 Tabel Kamus Data

1.	**Nama**	***tEmail***		
	Deskripsi :	t*Email* id_datalatih kombinasi status	= = = =	id_datalatih + kombinasi + status *int [0..9]* varchar [a..z] [A..Z][-] varchar

2	**Nama**	**tHistory**		
	Deskripsi :	tHistory	=	id_history + tgl_uji + id_massage + status
		id_history	=	*int [0..9]*
		tgl_uji	=	Timestamp
		id_massage	=	varchar [0..9] [a..z] [A..Z] [@]
		status_uji	=	varchar
		status_asli	=	varchar
3	**Nama**	**tStopword**		
	Deskripsi :	tStopword	=	id_sw + kataSw
		id_sw	=	*int [0..9]*
		kataSw	=	Varchar[a..z] [A..Z][-]

3.1.7.2. Struktur Tabel

Struktur Tabel merupakan urutan isi ataupun data yang berada dalam suatu *record*. Struktur tabel dalam Sistem Klasifikasi *Email Spam* dapat dilihat pada Tabel 3.20, Tabel 3.21 dan Tabel 3.22 :

Nama Tabel : t*Email*

Media Penyimpanan : Harddisk

Keterangan : Penyimpanan Data Latih

Tabel 3. 20 Struktur Tabel t*Email*

Field	Type	Size	Keterangan
id_dataLatih	int	20	Nomor id penyimpanan data latih
kombinasi	varchar	150	Kata-kata kombinasi
Status_uji	Varchar	20	Status field (*spam* atau ham)
Status_asli	Varchar	20	Status field (*spam* atau ham)

Nama Tabel : tHistory

Media Penyimpanan : Harddisk

Keterangan : Penyimpanan Data Latih

Tabel 3. 21 Struktur Tabel tHistory

Field	Type	Size	Keterangan
id_history	int	20	Nomor id penyimpanan history
tgl_uji	timestamp		Tanggal diujikannya *Email*
id_message	Varchar	40	Id dari *Email* yang telah diujikan
status	Varchar	20	Status field (*spam* atau ham)

Nama Tabel : tStopword

Media Penyimpanan : Harddisk

Keterangan : Penyimpanan Data Latih

Tabel 3. 22 Struktur Tabel tStopword

Field	Type	Size	Keterangan
id_sw	int	20	Nomor id penyimpanan stopword
kataSw	varchar	20	Berisi kata-kata stopword

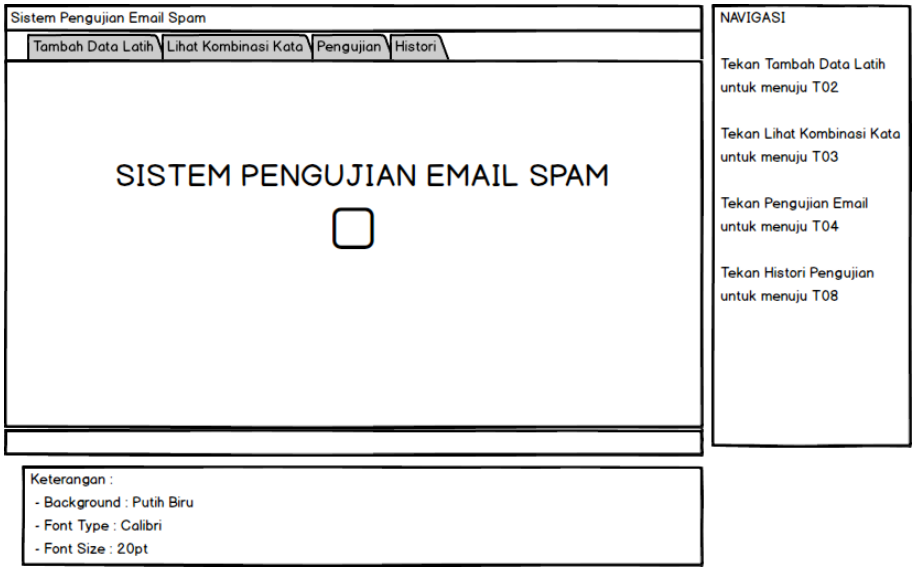
3.2. Perancangan Sistem

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi.

3.2.1. Perancangan Antarmuka

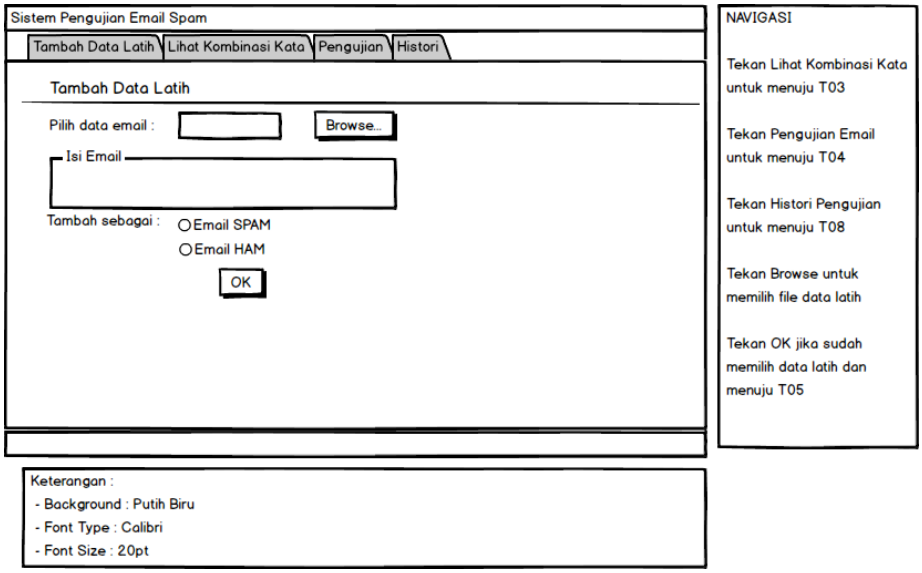
Perancangan antarmuka bertujuan untuk memberikan gambaran tentang sistem yang akan dibangun sehingga akan mempermudah dalam mengimplementasikan serta akan memudahkan dalam pembuatan sistem.

a. Perancangan Antarmuka Menu Utama : T01



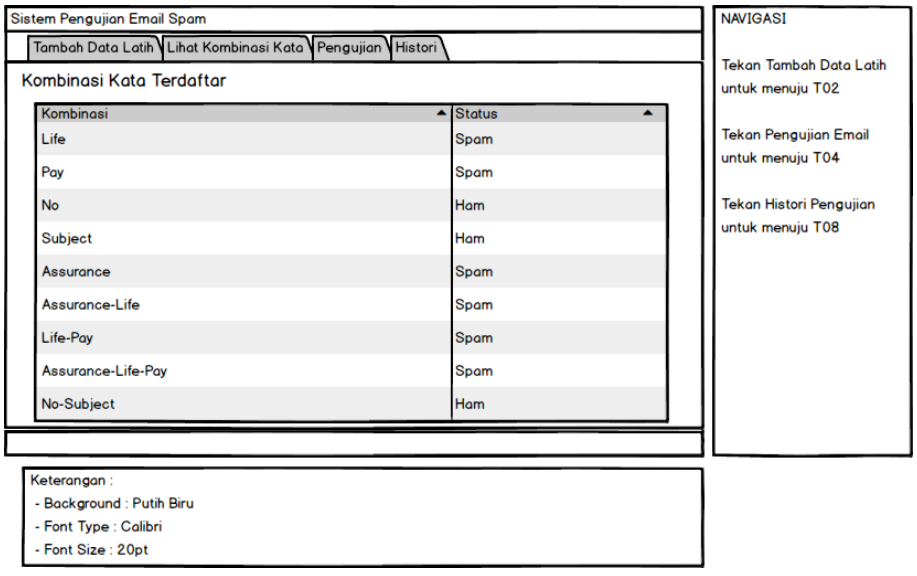
Gambar 3. 20 Perancangan Antarmuka Menu Utama

b. Perancangan Antarmuka Tambah Data Latih : T02



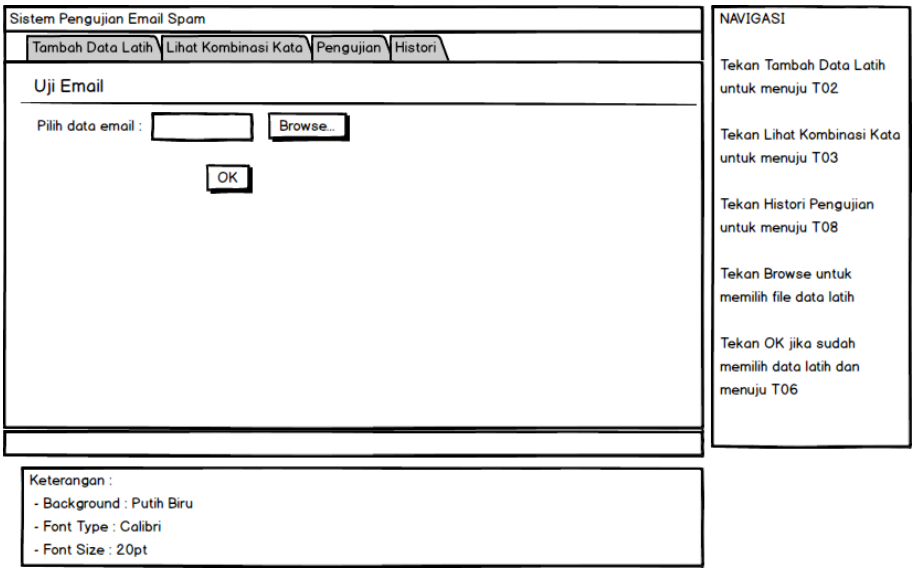
Gambar 3. 21 Perancangan Antarmuka Tambah Data Latih

c. Perancangan Antarmuka Lihat Kombinasi Kata : T03



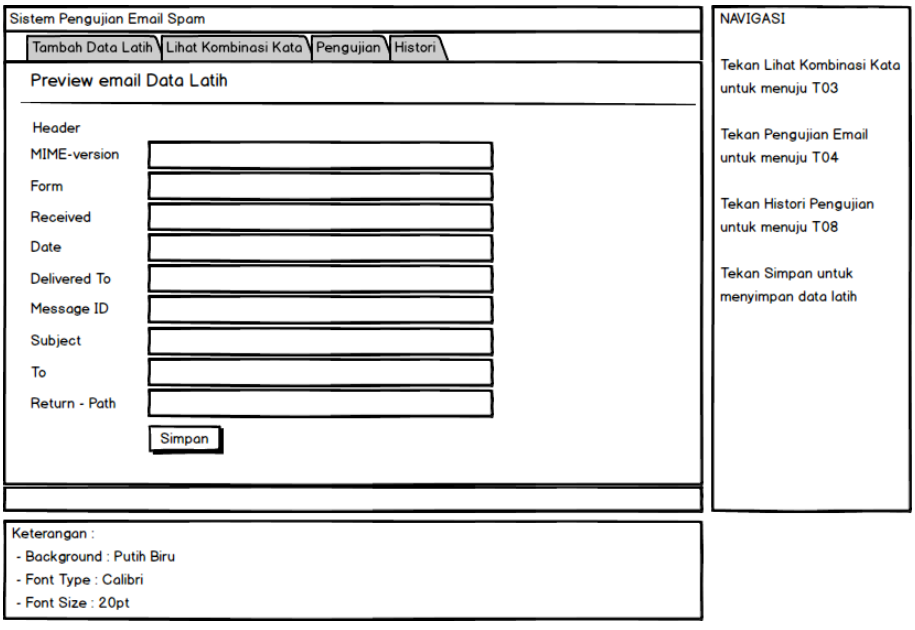
Gambar 3. 22 Perancangan Antarmuka Lihat kombinasi Kata

d. Perancangan Antarmuka Pengujian *Email* : T04



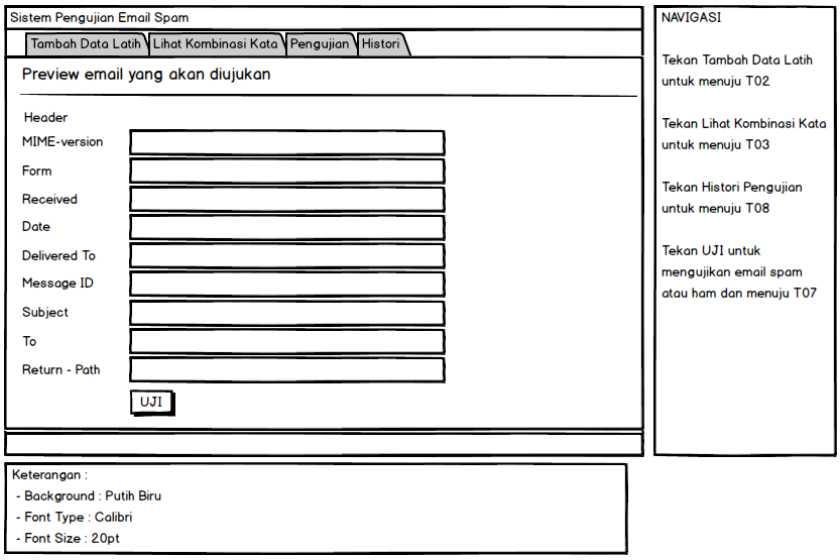
Gambar 3. 23 Perancangan Antarmuka Pengujian *Email*

e. Perancangan Antarmuka Preview *Email* data latih : T05



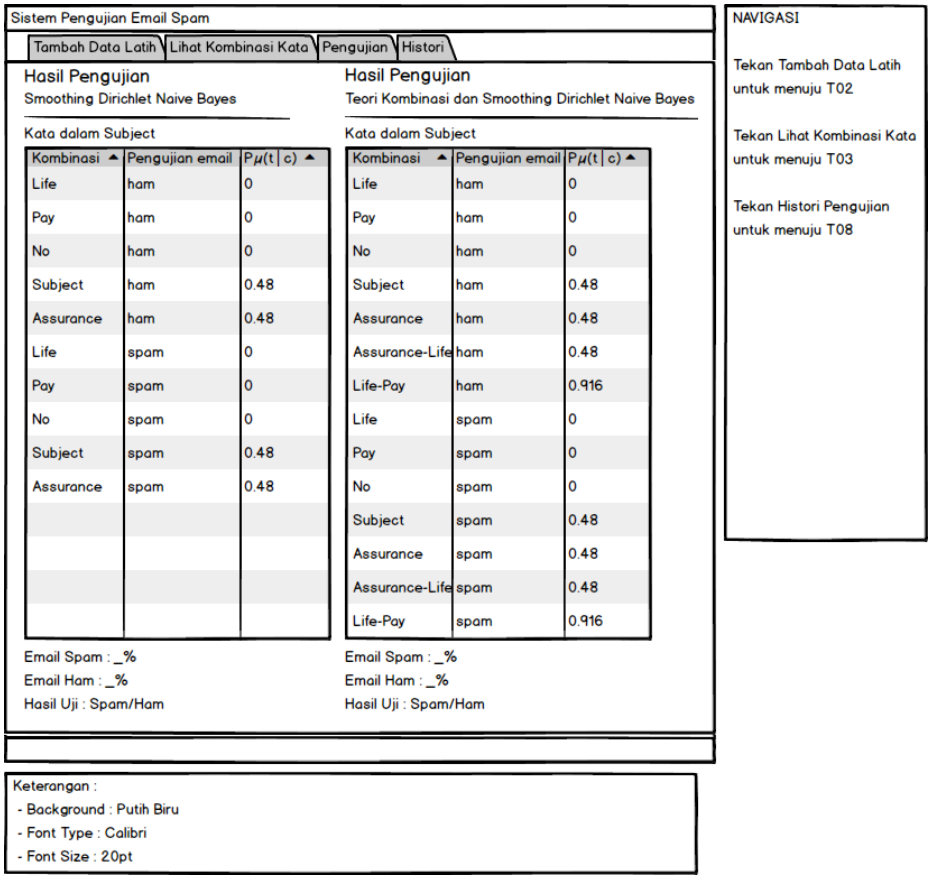
Gambar 3. 24 Perancangan Antarmuka Preview *Email* Data Latih

f. Perancangan Antarmuka Preview *Email* yang akan diujikan : T06



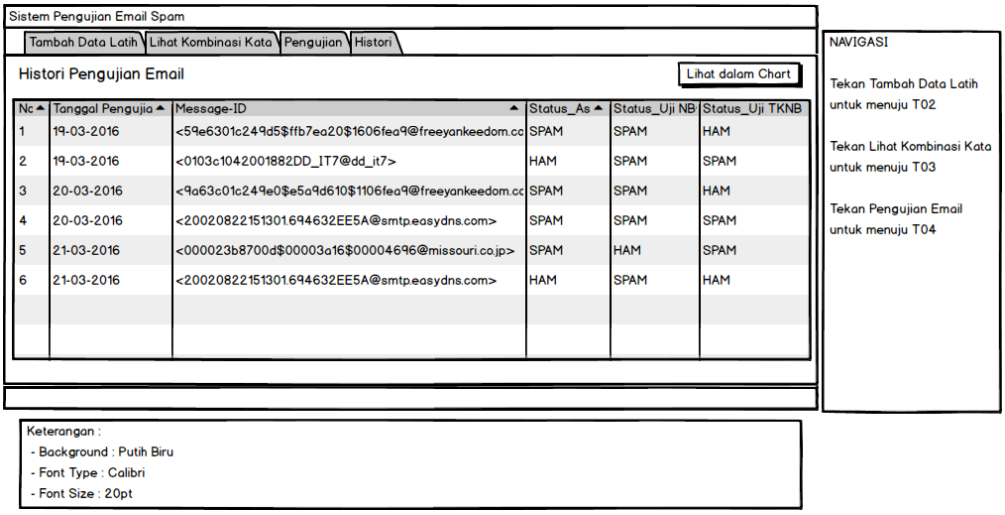
Gambar 3. 25 Perancangan Antarmuka Preview *Email* yang diujikan

g. Perancangan Antarmuka Hasil Pengujian *Email* : T07



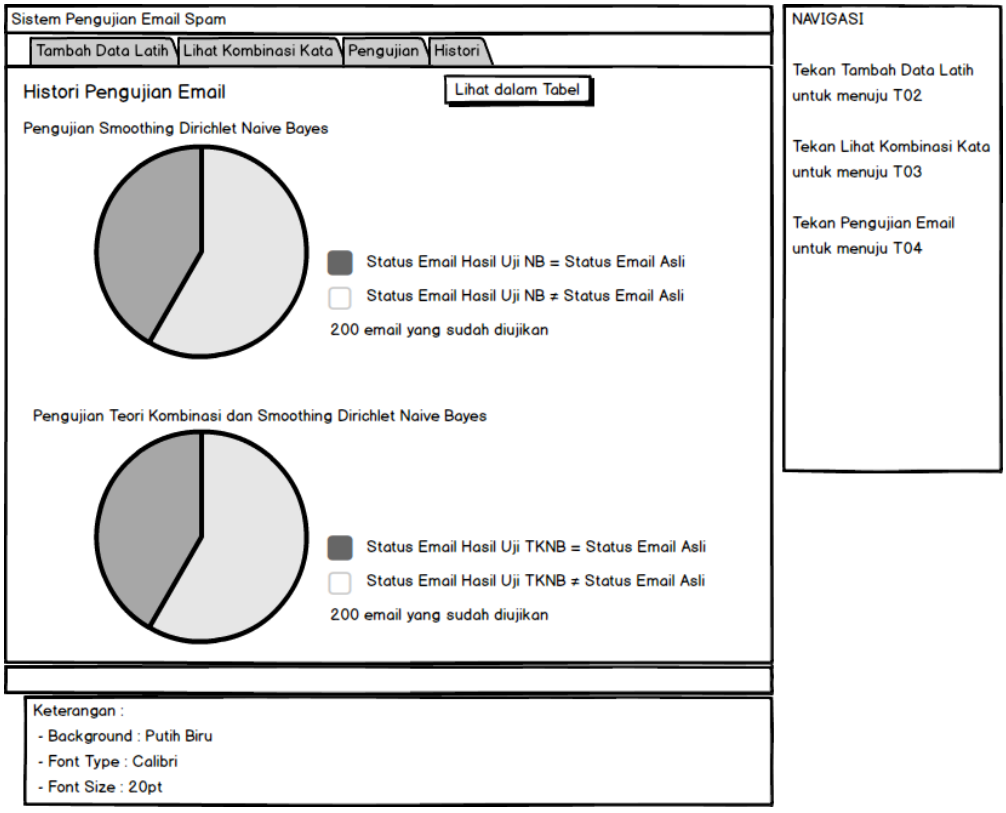
Gambar 3. 26 Perancangan Antarmuka Hasil Pengujian *Email*

h. Antarmuka Lihat Histori Pengujian dalam Tabel : T08



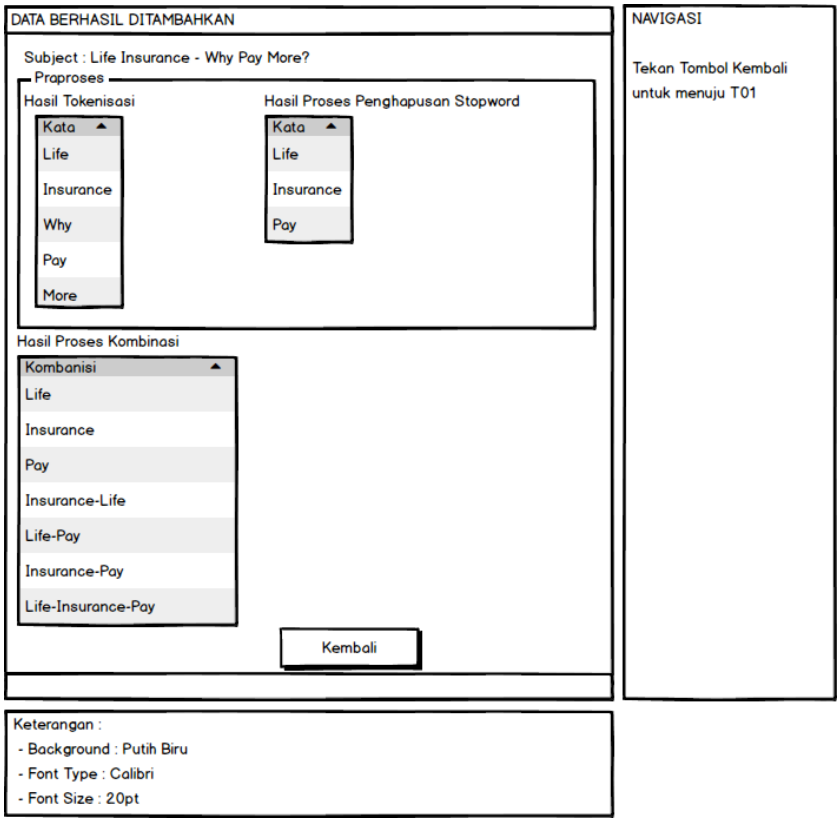
Gambar 3. 27 Perancangan Antarmuka Histori Pengujian dalam Tabel

i. Perancangan Antarmuka Lihat Histori Pengujian dalam Chart : T09



Gambar 3. 28 Perancangan Antarmuka Histori Pengujian dalam Chart

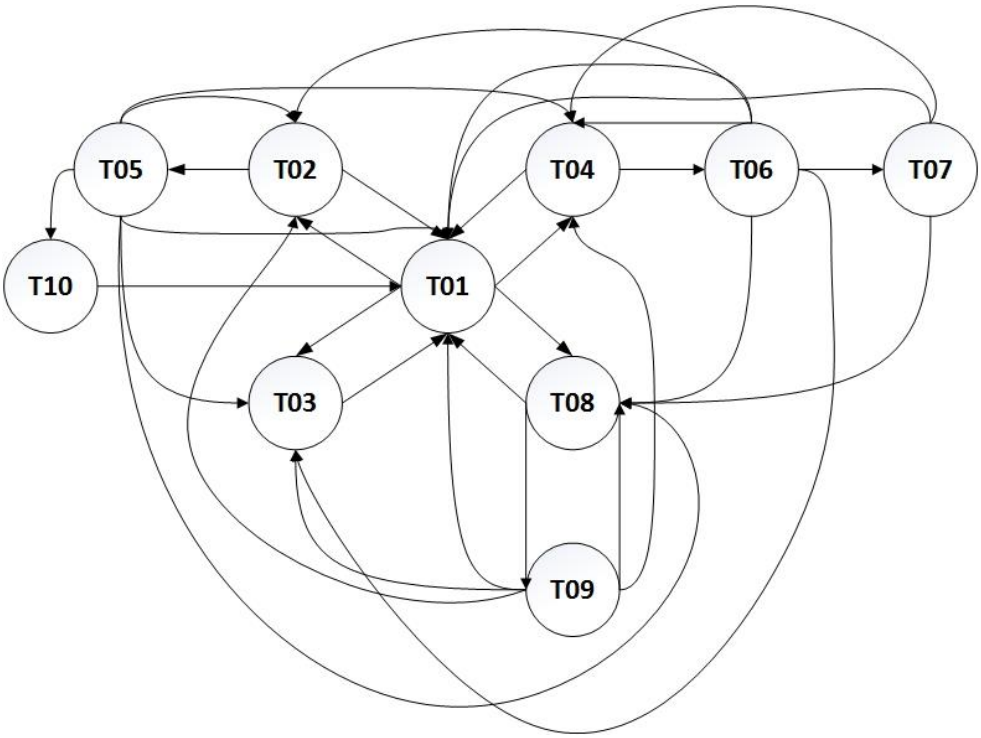
j. Perancangan Antarmuka Data Berhasil Ditambahkan



Gambar 3. 29 Perancangan Antarmuka Data Berhasil Ditambahkan

3.2.2. Jaringan Semantik

Jaringan semantik digunakan untuk menunjukan alur dari setiap menu dalam sistem, berikut jaringan semantik yang terdapat dalam sistem pada Gambar 3.30.



Gambar 3. 30 Jaringan Semantik