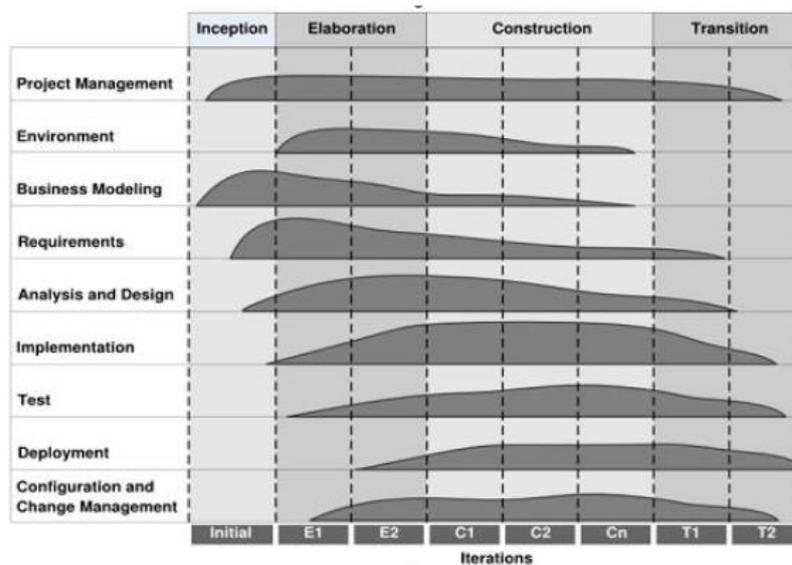


BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Pada bab sebelumnya telah dijelaskan bahwa penelitian yang dilakukan akan menggunakan diimplementasikan dalam bentuk pengembangan perangkat lunak berdasarkan metode serta analisa masalah yang telah ditetapkan sebelumnya. Perangkat lunak yang dikembangkan merupakan pemrograman berorientasi obyek dengan menggunakan prinsip *Rational Unified Process* (RUP) dengan terdiri dari beberapa fase seperti yang tampak pada gambar IV-1. Fase yang terdapat pada metode RUP yaitu fase inepsi, fase elaborasi, fase konstruksi, dan fase transisi yang meliputi proses pemodelan, kebutuhan (*requirement*), Analisa dan desain, implementasi, hingga pengujian yang terdapat pada setiap fase pengembangan.



Gambar IV-1. Siklus RUP (BusinessAnalystFaq.com)

4.2 Fase Insepsi

Fase insepsi merupakan fase awal yang terdapat dalam pengembangan perangkat lunak yang berupa kegiatan untuk menetapkan batasan-batasan yang dapat dilakukan sistem, kebutuhan yang diperlukan oleh sistem baik secara fungsional maupun non-fungsional hingga membentuk diagram-diagram yang akan digunakan untuk memperoleh perangkat lunak yang diinginkan.

4.2.1 Permodelan Bisnis

Proses enkripsi dan dekripsi dengan menggunakan bantuan jaringan saraf tiruan pada dasarnya menggunakan bantuan jaringan saraf tiruan dengan metode *backpropagation* untuk mengenali data masukan berdasarkan target yang telah ditentukan dalam hal ini adalah masukan itu sendiri. Jaringan saraf tiruan yang dibentuk mempunyai target yang sama dengan masukan karena tujuan pelatihan jaringan adalah memastikan data yang diolah dapat diubah menjadi bentuk atau nilai lain (dekripsi) dan dapat dikembalikan seperti semula agar dapat dikenali kembali (enkripsi). Pada dasarnya kriptografi menggunakan jaringan saraf tiruan dibagi menjadi tiga tahapan yaitu tahap pengolahan data, tahap pelatihan data, serta tahap pelatihan data yang meliputi enkripsi dan dekripsi terhadap data.

Sebelum data dapat dimasukkan ke dalam tahapan pelatihan dan pengujian data, tahapan awal berupa pemrosesan data untuk memastikan bahwa data yang diolah dapat diterima oleh jaringan yang dibentuk sebagai masukan. Jaringan saraf tiruan yang dibentuk bekerja dengan baik dengan *range* masukan mulai dari 0 hingga 1. Oleh karena itu, tahapan pertama adalah memastikan bahwa data

memiliki format numerik dan berada pada *range* 0 sampai dengan 1. Pada tahapan pelatihan data, masukan yang telah diolah kemudian dilatih agar dapat mengenali target yang telah ditetapkan yang merupakan masukan itu sendiri dengan tujuan agar keluaran yang dihasilkan dapat berupa data asal. Tahapan enkripsi serta dekripsi memanfaatkan hasil jaringan pelatihan. Keluaran yang dihasilkan dari oleh jaringan tersembunyi akan dimanfaatkan sebagai teks sandi (*cipher text*) sedangkan keluaran dari lapisan keluaran akan dimanfaatkan sebagai teks asli (*plain text*).

4.2.2 Kebutuhan Sistem

Kebutuhan sistem berdasarkan pemodelan bisnis yang telah dirancang untuk pengembangan perangkat lunak dalam penelitian ini antara lain fitur praproses teks, pelatihan (*training*) data, enkripsi data, dan dekripsi data.

4.2.2.1 Fitur Praproses Data Teks

Data asli yang berupa masukan merupakan data dalam bentuk teks atau *string* sehingga data masukan perlu di proses terlebih dahulu agar data memiliki format bilangan riil (*float*) serta berada pada jarak 0 sampai 1. Dalam proses ini, setiap karakter yang terdapat dalam kata diubah menjadi nilai angka dengan cara mencari bentuk lain dari karakter alfabet. Nilai alfabet memiliki jarak 1 sampai 26 yang merepresentasikan alfabet ke-N dalam susunan alfabetik. Misanya karakter ke-10 dalam susunan alfabetik memiliki representasi nilai “j” dan lain sebagainya. Dengan cara ini setiap karakter yang bertindak sebagai input mempunyai nilai riil. Selanjutnya nilai yang didapat akan diubah ke dalam *range* 0 hingga 1 dengan cara

membagi setiap bilangan dengan 26 sehingga nilai minimum yang didapat adalah $1/26$ yaitu 0.03 dan nilai maksimum yang didapat adalah 1 yang merupakan nilai dari $26/26$.

4.2.2.2 Fitur Pelatihan Data

Pada fitur ini, data yang sudah diolah dimasukkan ke dalam algoritma pelatihan yaitu *backpropagation*. Fitur ini digunakan untuk menghasilkan bobot yang diinginkan yang dapat digunakan untuk mengenali target yang telah ditetapkan. Bobot awal yang akan digunakan dalam pelatihan merupakan kunci (*key*) yang digunakan untuk melakukan proses pelatihan. Selanjutnya, hasil yang didapatkan nantinya akan memasuki tahapan pengolahan enkripsi dan dekripsi teks.

4.2.2.3 Fitur Enkripsi

Setelah memasuki fitur pelatihan, data yang didapat sudah dapat dienkripsi setelah sistem telah memiliki nilai kesalahan minimum (*Root Mean Square Error*) atau telah mencapai batas perulangan (*epoch*) yang telah ditetapkan. Nilai yang didapat dari lapisan keluaran pada jaringan tersembunyi akan digunakan untuk proses dekripsi dengan cara mengembalikan nilai riil ke proses asal agar mendapatkan nilai karakter yang bersesuaian. Cara ini merupakan kebalikan dari proses pengolahan data yaitu dengan cara mengalikan nilai yang didapat dengan 26 dan kemudian mencari nilai yang berkesesuaian dengan nilai ke-N dalam alfabetik.

4.2.2.4 Fitur Dekripsi

Serupa dengan fitur enkripsi, fitur dekripsi akan mengubah nilai yang terdapat pada lapisan keluaran ke dalam bentuk karakter yang berkesesuaian dan menggabungkannya menjadi teks asli (*plain text*). *Pengguna* dapat menggunakan fitur ini untuk menguji apakah sistem dapat melakukan proses enkripsi dan dekripsi dengan baik sebagaimana mestinya.

Fitur-fitur yang telah ditentukan di atas merupakan bagian dari kebutuhan fungsional yang dimiliki oleh sistem. Kebutuhan fungsional merupakan fungsi pokok yang harus ada setiap fungsinya agar sistem dapat bekerja sebagaimana yang telah ditetapkan berdasarkan kebutuhan pengguna. Kebutuhan fungsional sistem yang dibangun ditunjukkan pada tabel IV-1. Kebutuhan lainnya selain kebutuhan fungsional adalah kebutuhan non-fungsional yang merupakan fungsi tambahan yang berguna untuk menunjang performa dari perangkat lunak yang dihasilkan. Table IV-2 menunjukkan kebutuhan non-fungsional pada perangkat lunak yang dibangun.

Tabel IV–1. Kebutuhan Fungsional

No.	Kebutuhan
1	Sistem dapat menerima masukan berupa teks asli (<i>plain text</i>) dan kunci (<i>key</i>).
2	sistem dapat melakukan praproses data teks.
3	Sistem dapat melakukan proses pelatihan
4	Sistem dapat melakukan enkripsi terhadap data yang dihasilkan.

5	Sistem dapat melakukan dekripsi terhadap teks sandi (<i>cipher text</i>) yang didapatkan.
---	---

Tabel IV–2. Kebutuhan Non Fungsional

No.	Kebutuhan
1	Sistem memiliki antarmuka (<i>interface</i>) yang menarik dan mudah digunakan (<i>pengguna friendly</i>)
2	Sistem dapat memberikan pemberitahuan apabila terdapat kesalahan pada proses yang dilakukan.

4.2.3 Analisis dan Desain

Proses analisis dan desain pada perangkat lunak merupakan proses lanjutan setelah mendefinisikan kebutuhan terhadap sistem. Proses analisis merupakan proses menganalisis terhadap kebutuhan yang sudah didefinisikan serta melakukan analisis terhadap proses pengembangan sistem selanjutnya. Diagram-diagram seperti *usecase* dan *ssequence* diagram akan didesain pada tahapan ini berdasarkan kebutuhan yang telah didefinisikan juga sebelumnya.

4.2.3.1 Analisis Praproses Data

Tahapan pengolahan data sebelum dapat masuk pada jaringan saraf tiruan merupakan tahapan proposes data teks dengan cara melakukan partisi atau pembagian jumlah data menjadi blok berdasarkan jumlah neuron pada suatu jaringan dan mengubah data bertipe *string* menjadi *float*. Pada penelitian ini,

penggunaan karakter dengan huruf besar dan kecil tidak memiliki perbedaan pada representasi nilai alfabetik sehingga antara karakter ‘a’ dan ‘A’ merupakan huruf pertama pada alfabet. Praproses data terbagi menjadi empat proses antara lain:

1. Menghilangkan karakter lain pada masukan.

Karakter spasi atau tanda baca yang terdapat pada masukan merupakan petunjuk yang dapat digunakan untuk memahami pesan yang dimasukkan dengan cara membagi tiap kata dengan menggunakan karakter spasi yang dapat menunjukkan jumlah kata yang dimasukkan serta jumlah karakter yang terdapat dalam setiap kata atau menunjukkan jenis kata atau kalimat yang sedang dienkripsi. Menghilangkan karakter spasi dan tanda baca pada masukan merupakan upaya meningkatkan keamanan. Contohnya adalah *string* “universitas sriwijaya!” akan diubah menjadi “universitassriwijaya”.

2. Mengelompokkan data menjadi blok.

Tipe kriptografi yang digunakan dalam penelitian ini adalah *block cipher* yang membagi data dengan jumlah yang sama. Pada penelitian ini ukuran blok yang dibentuk adalah 5 karakter yang sesuai dengan jaringan yang dibentuk.

T	E	K	N	I	K	I	N	F	O	R	M	A	T	I	K	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Blok masukan diatas akan dipartisi menjadi blok-blok yang mempunyai 5 karakter pada setiap bloknya.

Blok pertama

T	E	K	N	I
---	---	---	---	---

Blok kedua

K	I	N	F	O
---	---	---	---	---

Blok ketiga

R	M	A	T	I
---	---	---	---	---

Blok keempat

K	A	Z	Z	Z
---	---	---	---	---

Untuk menjaga agar setiap blok mempunyai jumlah karakter sebanyak 5 dengan tujuan untuk mempertahankan bentuk jaringan yang dibentuk maka blok terakhir yang tidak mencukupi syarat akan ditambahkan karakter tambahan yaitu “z”.

3. Mengubah tipe data.

Masukan yang bertipe *String* akan diubah menjadi tipe data *Double* karena jaringan saraf yang dibentuk hanya menerima masukan bertipe *non-String*. Setiap nilai dari karakter yang terdapat pada masukan akan dikonversi menjadi nilai yang bersesuaian pada alfabet.

4. Mengubah nilai data pada jangkauan yang diinginkan.

Seperti tampak pada tabel IV-3 bahwa setiap karakter pada alfabet yang memiliki nilai konversi dari *range* 1 sampai 26 akan dibagi dengan jumlah total dari alfabet yaitu 26 untuk mendapatkan nilai dengan *range* 0 hingga 1.

Tabel IV-3. Tabel konversi data masukan

Karakter	Nilai	Konversi pecahan
a	1	$1 / 26 = 0.03$
b	2	$2 / 26 = 0.07$
c	3	$3 / 26 = 0.11$
d	4	$4 / 26 = 0.15$

4.2.3.2 Analisis dan Desain Distribusi Kunci

Pada pembuatan sistem kriptografi penelitian ini, kunci yang digunakan yaitu kunci simetris yang artinya proses enkripsi dan dekripsi menggunakan penggunaan kunci yang sama. Kunci yang ditentukan nantinya akan dijadikan sebagai bobot yang digunakan dalam pelatihan jaringan sehingga berhasil atau tidaknya proses enkripsi dan dekripsi juga tergantung pada kunci yang ditentukan. Jaringan yang dibentuk pada penelitian ini yaitu $5 \times 5 \times 5$ yang artinya setiap *neuron* akan membutuhkan 5 buah bobot yang nantinya akan digunakan untuk meneruskan masukan yang diberikan sehingga jumlah total bobot yang dibutuhkan adalah 50 buah bobot (10 buah *neuron* yang berpasangan dengan 5 buah *neuron* setelahnya sehingga 10×5).

Proses pendistribusian kunci akan dibagikan pada setiap bobot secara sekuensial berdasarkan urutan setiap karakter pada kunci yang diberikan. Contohnya setiap karakter pada kunci "INDONESIA" akan didistribusikan satu persatu pada bobot sesuai urutan *layer-neuron*.

Tabel IV–4. Bobot antara lapisan masukan dan lapisan tersembunyi

Input layer	Hidden layer	Neuron 0	Neuron 1	Neuron 2	Neuron 3	Neuron 4
Neuron 0		I	N	D	O	N
Neuron 1		E	S	I	A	I
Neuron 2		N	D	O	N	E
Neuron 3		S	I	A	I	N
Neuron 4		D	O	N	E	S

Tabel IV–5. Bobot antara lapisan tersembunyi dan lapisan keluaran

Hidden layer	Output layer	Neuron 0	Neuron 1	Neuron 2	Neuron 3	Neuron 4
Neuron 0		I	A	I	N	D
Neuron 1		O	N	E	S	I
Neuron 2		A	I	N	D	O
Neuron 3		N	E	S	I	A
Neuron 4		I	N	D	O	N

4.2.3.3 Analisis Kunci yang Digunakan

Jumlah bobot yang terdapat pada masing-masing lapisan masukan menuju lapisan tersembunyi dan lapisan tersembunyi menuju lapisan keluaran berjumlah sebanyak 25 buah sehingga jumlah bobot yang ada yaitu 50 buah. Kunci yang akan

dikonversi menjadi bobot merupakan karakter dengan ukuran 8-bit pada setiap karakter sehingga ukuran maksimum kunci yaitu 400-bit.

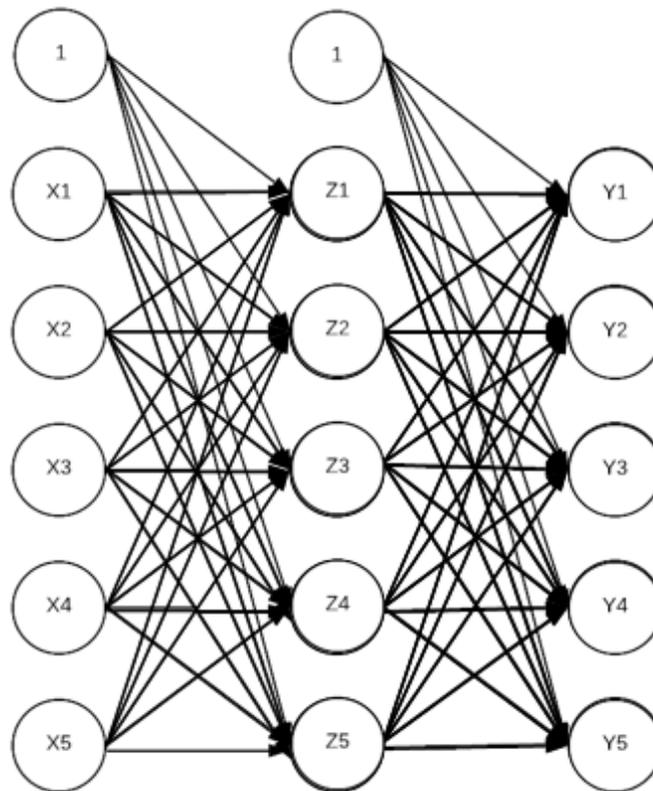
4.2.3.4 Analisis dan Desain Arsitektur Jaringan Saraf Tiruan

Arsitektur jaringan saraf tiruan yang dibangun pada penelitian ini merupakan jaringan saraf tiruan dengan layer 5x5x5 yaitu sistem JST mempunyai 3-layer dengan memiliki 5 buah neuron pada setiap lapisannya. Jaringan memiliki satu buah lapisan masukan atau *input layer*, lapisan keluaran atau *output layer*, serta memiliki satu buah lapisan tersembunyi atau *hidden layer* diantara lapisan masukan dan lapisan keluaran seperti yang terlihat pada gambar IV-1. Menurut Arief (*tanpa tahun*) dalam penelitiannya yang berjudul “Perancangan Algoritma Kriptografi Kunci Simetri dengan Menggunakan Jaringan Saraf Tiruan” bahwa laju pembelajaran sebesar 0.8 memiliki tingkat kesalahan kurang dari 4% pada rata-rata percobaan dengan nilai momentum antara 0.2 hingga 0.8. Selain itu, penelitian tersebut menunjukkan bahwa jaringan dengan jumlah node 5 memiliki tingkat kesalahan dekripsi sebesar 17.37%. Jaringan ini juga memiliki satu buah bias dengan nilai satu pada masing-masing lapisan masukan dan lapisan tersembunyi. Secara menyeluruh, jaringan saraf tiruan yang dibentuk mempunyai arsitektur dengan komponen sebagai berikut:

- a. Masing-masing lapisan memiliki 5 buah neuron.
- b. Fungsi aktivasi yang digunakan adalah *sigmoid biner*.
- c. *Learning rate* sebesar 0.8.
- d. *Minimum error* sebesar 0.0001.

- e. Jumlah *epoch* sebanyak 10000 kali.

Perulangan yang dilakukan pada proses pelatihan akan berhenti apabila telah melakukan jumlah perupangan sesuai dengan jumlah *epoch* yang ditentukan yaitu 10000 kali atau ketika proses pelatihan telah mencapai nilai kesalahan minimum sebesar 0.0001.



Gambar IV-2. Arsitektur Jaringan Saraf Tiruan

4.2.3.5 Analisis Algoritma Backpropagation

Proses pelatihan yang terjadi pada algoritma jaringan saraf tiruan *backpropagation* meliputi tiga fase yaitu fase maju (*feedforward*), fase propagasi balik (*backpropagation*), serta fase memperbarui bobot dan bias.

1. Fase Maju (*feedforward*).

Pada tahapan ini, seluruh masukan akan langsung diproses pada setiap lapisan dan langsung diteruskan ke lapisan berikutnya sesuai dengan bobot yang terdapat dalam neuron. Pada lapisan terakhir yaitu lapisan keluaran, keluaran yang didapatkan akan dibandingkan dengan target yang telah ditentukan pada proses pelatihan jaringan. Apabila keluaran pada lapisan keluaran tidak sama dengan target maka proses lanjut ke fase yang kedua yaitu propagasi bali (*backpropagation*). Pada fase ini terdapat empat tahapan antara lain:

langkah (i) Hitung jumlah bobot yang terdapat pada setiap lapisan tersembunyi (z).

$$z_{in_j} = v_{0i} + \sum_{i=1}^n x_i v_{ij} \quad (IV-1)$$

langkah (ii) Hitung sinyal keluaran pada setiap *neuron* dengan fungsi aktivasi yang dipakai pada lapisan tersembunyi.

$$z_j = f_{(z_{in_j})} \quad (IV-2)$$

langkah (iii) Hitung jumlah bobot yang terdapat pada setiap lapisan keluaran (y).

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (IV-3)$$

langkah (iv) Hitung sinyal keluaran pada setiap *neuron* dengan fungsi aktivasi yang dipakai pada lapisan keluaran.

$$y_k = f_{(y_{in_k})} \quad (IV-4)$$

2. Fase Propagasi Balik (*backpropagation*).

Pada fase ini, nilai yang didapat dikembalikan kembali atau dilakukan proses perhitungan kembali terhadap bobot-bobot yang ada dengan tujuan untuk menyesuaikan dengan bobot yang dapat mengenali target. Dengan kata lain bahwa pada tahapan ini nilai yang telah diperoleh pada lapisan keluaran akan didistribusikan kembali ke setiap *neuron* yang ada pada lapisan sebelum lapisan keluaran. Pada fase ini terdapat 8 tahapan yaitu:

langkah (v) Hitung nilai basis kesalahan (*error*) berdasarkan perbedaan target pada pola masukan.

$$\delta_k = (t_k - y_k) f'_{(y_{in_k})} \quad (IV-5)$$

langkah (vi) Hitung nilai bobot berdasarkan aturan perbaikan bobot (*weight correction term*).

$$\Delta w_{ik} = \alpha \delta_k z_j \quad (IV-6)$$

langkah (vii) Hitung nilai bias berdasarkan aturan perbaikan bias (*bias correction term*).

$$\Delta w_{0k} = \alpha \delta_k \quad (IV-7)$$

langkah (viii) Hitung nilai masukan dari unit pada lapisan atas untuk setiap lapisan tersembunyi.

$$\delta_{in_j} = w_{0k} + \sum_{k=1}^m \delta_k w_{jk} \quad (\text{IV-8})$$

langkah (ix) Hitung kesalahan dengan mengalikan turunan dari fungsi aktivasi.

$$\delta_j = \delta_{in_j} f'_{(z_{in_j})} \quad (\text{IV-9})$$

langkah (x) Hitung bobot yang tepat.

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (\text{IV-10})$$

langkah (xi) Hitung bias yang tepat.

$$\Delta v_{0j} = \alpha \delta_j \quad (\text{IV-11})$$

3. Fase perubahan bobot dan bias.

Pada tahapan ini, seluruh bobot dan bias baru yang telah didapatkan akan diterapkan pada setiap *neuron* sesuai dengan nilai yang baru didapatkan. Tahapan yang terdapat pada fase ini antara lain adalah:

langkah (xii) Memperbarui bobot dan *bias* pada lapisan tersembunyi.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (\text{IV-12})$$

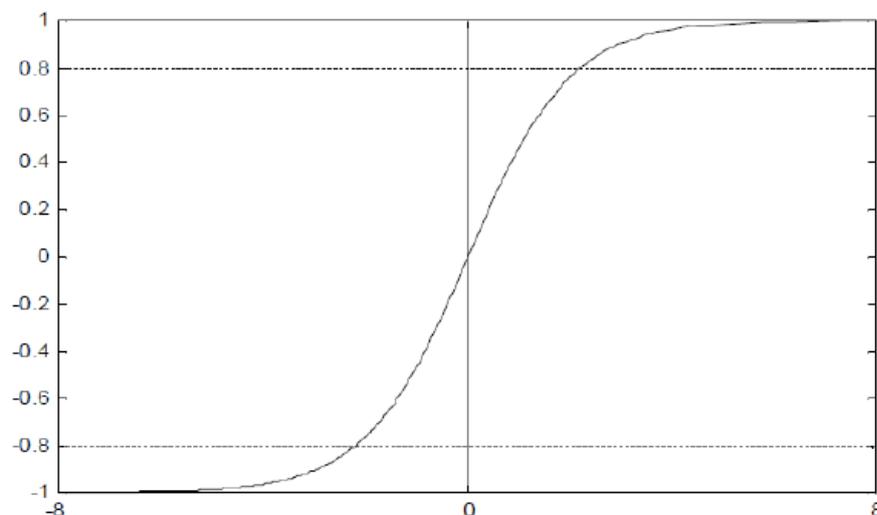
langkah (xiii) Memperbarui bobot dan *bias* pada lapisan keluaran.

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (\text{IV-13})$$

4.2.3.6 Analisis Fungsi Aktivasi

Pada setiap keluaran yang terdapat baik pada lapisan tersembunyi maupun lapisan keluaran memiliki operator matematik berupa fungsi aktivasi yang bertujuan untuk mengaktifkan atau menonaktifkan setiap neuron yang terdapat pada setiap lapisan yang menerima bobot dari lapisan sebelumnya. menurut Arief (tanpa tahun) bahwa fungsi aktivasi menunjukkan kemampuan jaringan untuk dapat menyesuaikan bobot dan keluaran berdasarkan target yang telah ditentukan. Fungsi aktivasi yang digunakan dalam jaringan saraf tiruan harus dapat bersifat terus menerus (kontinu), dapat didiferensiasi, dan tidak menurun apabila secara monoton.

Karena masukan yang digunakan diubah dalam jangkauan 0 hingga 1, maka fungsi aktivasi yang digunakan adalah *sigmoid biner*. Julpan et. al (2015) mengungkapkan dalam penelitiannya yang berjudul “*analisis fungsi aktivasi sigmoid biner dan sigmoid bipolar dalam algoritma backpropagation pada prediksi kemampuan siswa*” bahwa fungsi aktivasi *sigmoid biner* mempunyai kemampuan akurasi yang baik ketika digunakan pada algoritma *backpropagation*.



Gambar IV-3. Fungsi Aktivasi *Sigmoid Biner* (Fausett, L., 1994)

Fungsi aktivasi biner serta turunan dari fungsi aktivasi biner didefinisikan sebagai berikut:

$$y = f(x) = \frac{1}{(1+e^{-x})} \quad (\text{IV-14})$$

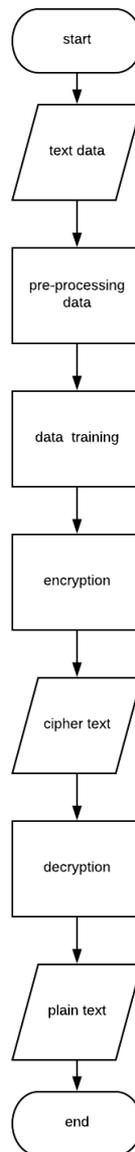
$$f(x) = f'(x) = (1 - f(x)) \quad (\text{IV-15})$$

4.2.3.7 Analisis Perangkat Lunak

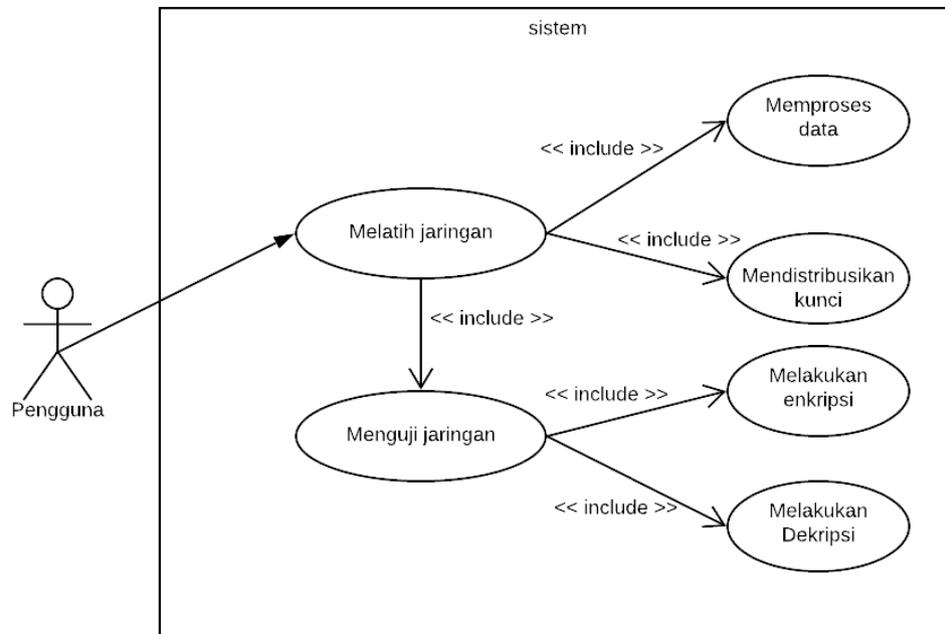
Bagian ini membahas mengenai analisis perangkat lunak enkripsi dan dekripsi data teks dengan menggunakan algoritma jaringan saraf tiruan. *backpropagation*. Analisis perangkat lunak meliputi arsitektur perangkat lunak dan kebutuhan-kebutuhan perangkat lunak.

a. Dekripsi umum perangkat lunak

Perangkat lunak yang dikembangkan dalam penelitian ini adalah perangkat lunak enkripsi dan dekripsi data teks dengan menggunakan jaringan saraf tiruan *backpropagation*. Pemrograman dilakukan dengan konsep *Object Oriented Programming* (OOP) dengan merepresentasikan data menjadi objek-objek. Bahasa pemrograman yang digunakan adalah *java* berbasis desktop secara *offline*. Gambar IV-3 menunjukkan cara kerja sistem ini yaitu dengan menerima masukan berupa data teks kemudian data yang dimasukkan akan dilatih menggunakan jaringan saraf tiruan untuk selanjutnya diuji untuk melakukan proses enkripsi serta dekripsi.



Gambar IV-4. Proses Enkripsi dan Dekripsi menggunakan BPNN

b. Diagram *use case*Gambar IV-5. Diagram *Use Case*

c. Tabel Definisi Aktor

Pada pengembangan perangkat lunak ini terdapat satu orang aktor yang dapat mengakses sistem yaitu *pengguna*. Deskripsi aktor dijelaskan pada tabel IV-4 sebagai berikut.

Tabel IV-6. Definisi Aktor

No.	Aktor	Deskripsi
1.	Pengguna	Aktor ini memiliki akses untuk berhubungan langsung pada sistem untuk melakukan proses

		pelatihan dan pengujian yang meliputi proses enkripsi dan dekripsi.
--	--	---

d. Tabel Definisi *Use Case***Tabel IV-7.** Definisi *Use Case*

No.	<i>Use Case</i>	Definisi
1.	Proses Pelatihan	Kegiatan yang dilakukan untuk melatih masukan yang diberikan dengan menggunakan bobot yang telah ditetapkan terhadap target yang sudah ditentukan.
2.	Proses Pengujian	Kegiatan yang dilakukan untuk menguji jaringan yang telah dilatih untuk mengenali masukan yang telah dilatih sebelumnya sebelum memasuki tahapan enkripsi dan dekripsi.
3.	Praproses data	Kegiatan yang dilakukan untuk mengubah data teks menjadi nilai yang dapat diterima oleh jaringan.
4.	Distribusi kunci	Kegiatan yang dilakukan untuk membagi bobot yang didapatkan dari kunci yang telah ditentukan sebelumnya.
5.	Enkripsi	Kegiatan pengujian jaringan untuk menghasilkan teks sandi setelah proses pelatihan jaringan.
6.	Dekripsi	Kegiatan pengujian jaringan untuk menghasilkan teks asli setelah proses pelatihan jaringan.

e. Skenario *Use Case*

Skenario merupakan penjelasan hubungan antara aktor terhadap sistem dan sebaliknya. Berikut ini merupakan skenario setiap *use case* yang telah didefinisikan sebelumnya yang menggambarkan interaksi yang terdapat pada *use case* tersebut.

1) Skenario *use case* melakukan proses pelatihan jaringan.

No	: 001
Nama <i>Use Case</i>	: Proses pelatihan
Aktor	: <i>pengguna</i>
Tujuan	: Untuk mendapatkan bobot yang dapat digunakan pada proses pengujian.
Deskripsi	: Kegiatan ini merupakan proses melatih jaringan menggunakan masukan yang diterima dengan menggunakan target yang telah ditetapkan.
Kondisi Awal	: Keluaran yang belum sama atau mendekati target yang ditentukan.
Kondisi Akhir	: Jaringan telah mencapai <i>epoch</i> maksimum atau telah mencapai nilai kesalahan minimum.

Tabel IV-8. Skenario *Use Case* Proses Pelatihan

Peneliti	Sistem
1. memasukkan masukan yang akan dilatih.	
2. memasukkan kunci yang akan digunakan sebagai bobot pada proses pelatihan.	
3. menentukan jumlah <i>epoch</i> yang akan dilakukan.	
4. menekan tombol “train”	
	5. membaca masukan yang sudah diproses
	6. melakukan proses pelatihan.

2) Skenario *use case* melakukan proses pengujian jaringan.

No : 002

Nama *Use Case* : Proses pengujian

Aktor : *pengguna*

Tujuan : Bertujuan untuk menguji jaringan yang telah dilatih untuk melakukan proses enkripsi dan dekripsi terhadap data teks yang diberikan.

- Deskripsi : Kegiatan ini merupakan kegiatan lanjutan setelah jaringan dilatih yaitu pengujian jaringan.
- Kondisi Awal : Belum dapat melakukan enkripsi dan dekripsi data.
- Kondisi Akhir : Telah dapat melakukan enkripsi dan enkripsi berdasarkan proses pelatihan yang dilakukan sebelumnya.

Tabel IV-9. Skenario *Use Case* Proses Pengujian

Peneliti	Sistem
1. Menekan tombol “testing” untuk melakukan pengujian terhadap data yang telah dilatih.	
	2. Membaca keluaran yang didapatkan
	3. Mengubah format data sesuai masukan.

3) Skenario praproses data.

No : 003

Nama <i>Use Case</i>	: Praproses data teks
Aktor	: <i>pengguna</i>
Tujuan	: Untuk mendapatkan tipe data yang dapat dikenali serta bekerja secara maksimal dalam jaringan yang dibentuk.
Deskripsi	: Kegiatan ini merupakan kegiatan awal penelitian yaitu mengolah data sesuai dengan kebutuhan jaringan yang dibangun.
Kondisi Awal	: Data masih dalam bentuk teks dan belum dapat dikenali oleh jaringan.
Kondisi Akhir	: Data sudah dipartisi menjadi beberapa blok dan dalam bentuk angka dengan <i>range</i> 0 sampai 1 dan telah dapat diproses oleh jaringan.

Tabel IV-10. Skenario *Use Case* Praproses Data

Peneliti	Sistem
1. menekan tombol " <i>browse</i> " atau mengetik pada kolom yang telah disediakan.	
2. menekan tombol " <i>train</i> "	

	3. membaca masukan <i>string</i> yang diterima
	4. membagi tiap karakter kedalam beberapa blok.
	5. mengubah tipe data menjadi <i>double</i> .
	6. mengubah data ke dalam <i>range</i> 0 sampai 1.

4) Skenario proses distribusi kunci.

No : 004

Nama *Use Case* : Proses distribusi kunci

Aktor : *pengguna*

Tujuan : Untuk menentukan bobot jaringan sesuai dengan kunci yang telah ditentukan.

Deskripsi : Kegiatan ini merupakan kegiatan sebelum pelatihan yaitu menentukan bobot yang akan digunakan dalam proses pelatihan.

Kondisi Awal : Belum ada bobot pada jaringan dan jaringan belum dapat dilatih sebagaimana mestinya.

Kondisi Akhir : Bobot jaringan sudah ditentukan dan proses pelatihan jaringan telah dapat dilakukan.

Tabel IV-11. Skenario *Use Case* Proses Distribusi kunci.

Peneliti	Sistem
1. memasukkan kunci pada kolom yang disediakan.	
2. menekan tombol “ <i>train</i> ”	
	3. mengubah tipe data kunci
	4. mengubah kunci ke dalam <i>range</i> 0 sampai 1.
	5. mendistribusikan kunci berdasarkan urutan bobot pada jaringan.

5) Skenario proses enkripsi.

No : 005

Nama *Use Case* : Proses enkripsi

Aktor : *Pengguna*

Tujuan : Bertujuan untuk menghasilkan teks sandi atau *cipher text*.

- Deskripsi : Kegiatan ini termasuk pada kegiatan pengujian data untuk mengubah data yang dihasilkan menjadi teks sandi.
- Kondisi Awal : Data masukan masih dapat dikenali.
- Kondisi Akhir : Data masukan telah berubah menjadi teks sandi yang tidak dapat dikenali.

Tabel IV-12. Skenario *Use Case* Proses Enkripsi

Peneliti	Sistem
1. Menekan tombol “ <i>test</i> ”	
	2. membaca keluaran yang terdapat pada jaringan tersembunyi.
	3. melakukan konversi terhadap nilai yang terdapat pada jaringan tersembunyi.

6) Skenario proses dekripsi.

No : 006

Nama *Use Case* : Proses dekripsi.

Aktor : *Pengguna*

- Tujuan : Untuk mengubah teks sandi yang dihasilkan oleh jaringan menjadi teks asli kembali agar dapat dikenali.
- Deskripsi : Kegiatan ini merupakan kegiatan pengujian jaringan untuk dapat mengenali teks asli yang digunakan sebagai masukan pada jaringan.
- Kondisi Awal : Teks yang dihasilkan masih belum dapat dikenali atau masih berupa teks sandi (*cipher text*)
- Kondisi Akhir : Teks sandi telah diubah menjadi teks asli (*plain text*) yang dapat dikenali melalui proses pengujian jaringan.

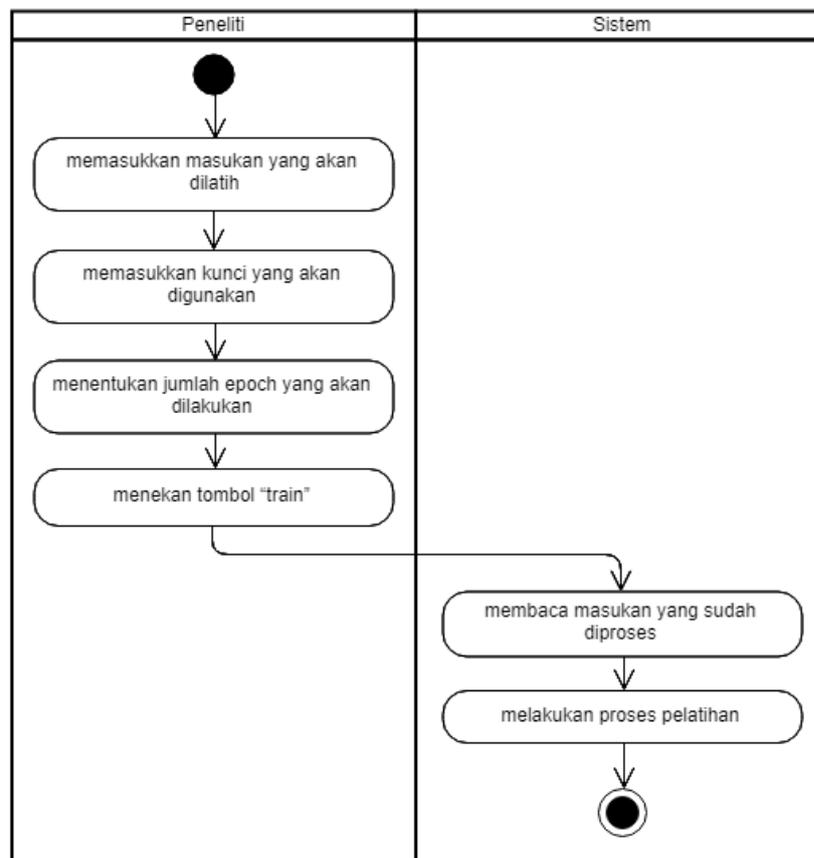
Tabel IV-13. Skenario *Use Case* Proses Dekripsi

Peneliti	Sistem
1. Menekan tombol “ <i>test</i> ”	
	2. membaca keluaran pada neuron jaringan keluaran.

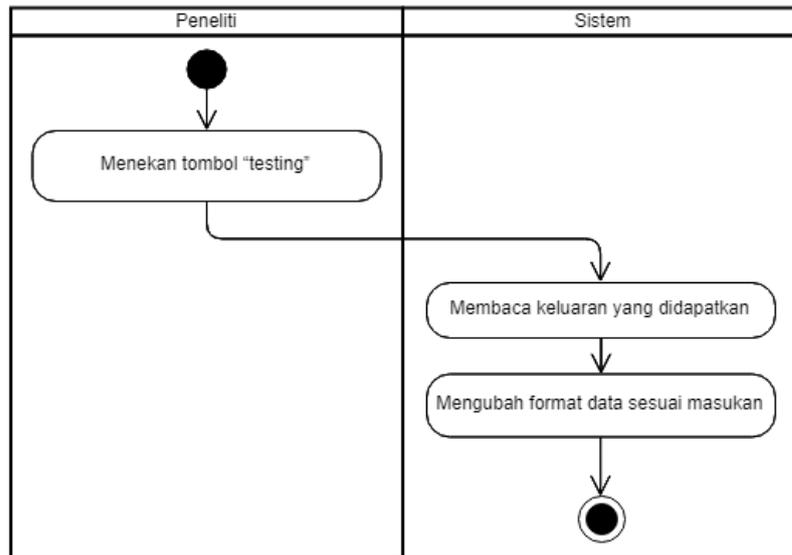
	3. melakukan konversi terhadap nilai yang terdapat pada jaringan keluaran.
--	--

f. Diagram Aktivitas

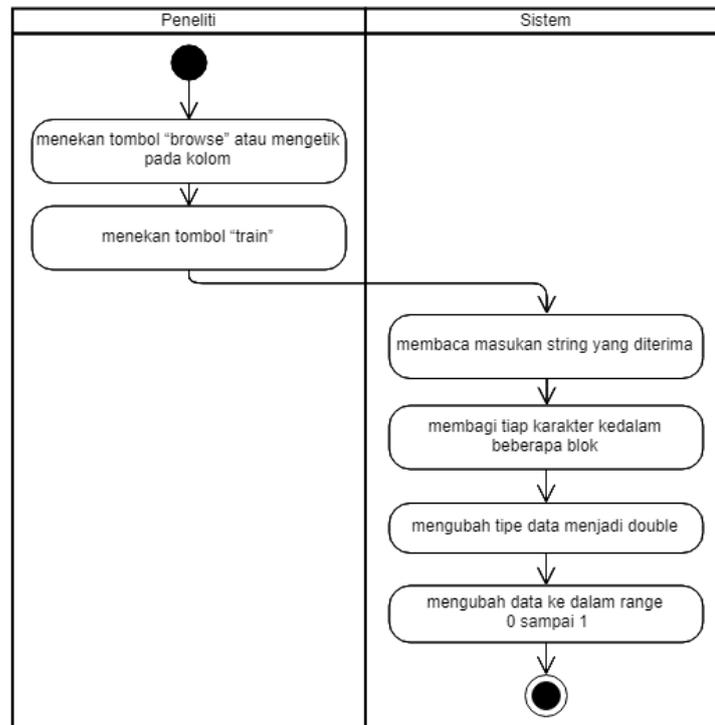
Diagram aktivitas merupakan gambaran mengenai urutan kegiatan yang dilakukan oleh *pengguna* terhadap sistem yang dibangun. Pada bagian ini akan menampilkan diagram aktivitas yang terjadi pada proses pelatihan, proses pengujian, proses praproses data, proses distribusi kunci, proses enkripsi, dan proses dekripsi.



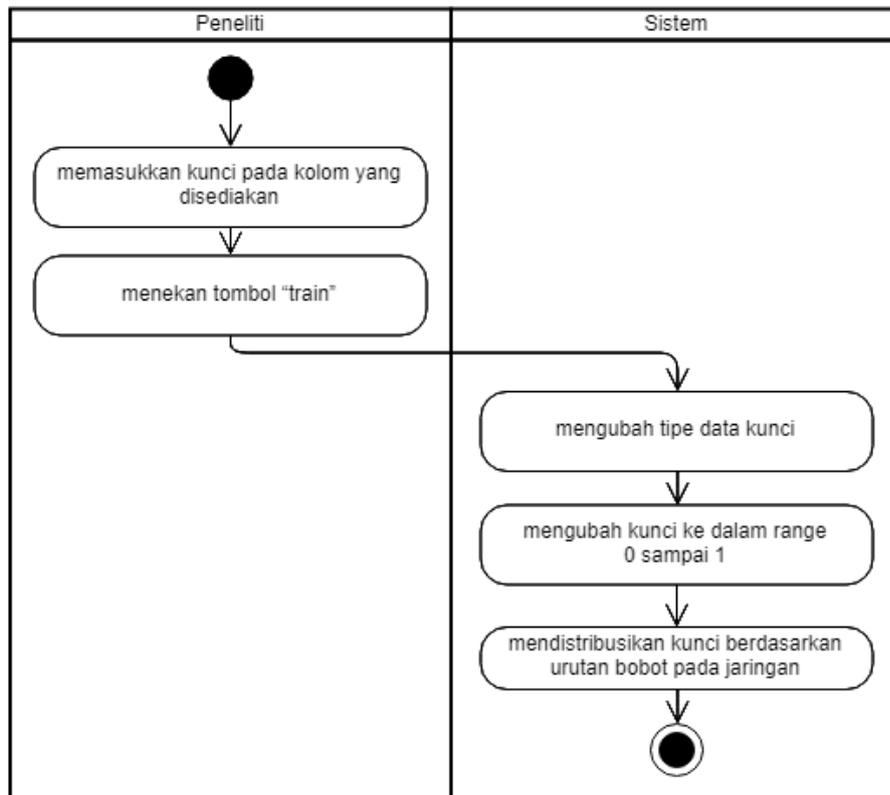
Gambar IV-6. Diagram Aktivitas *Use Case* Proses Pelatihan



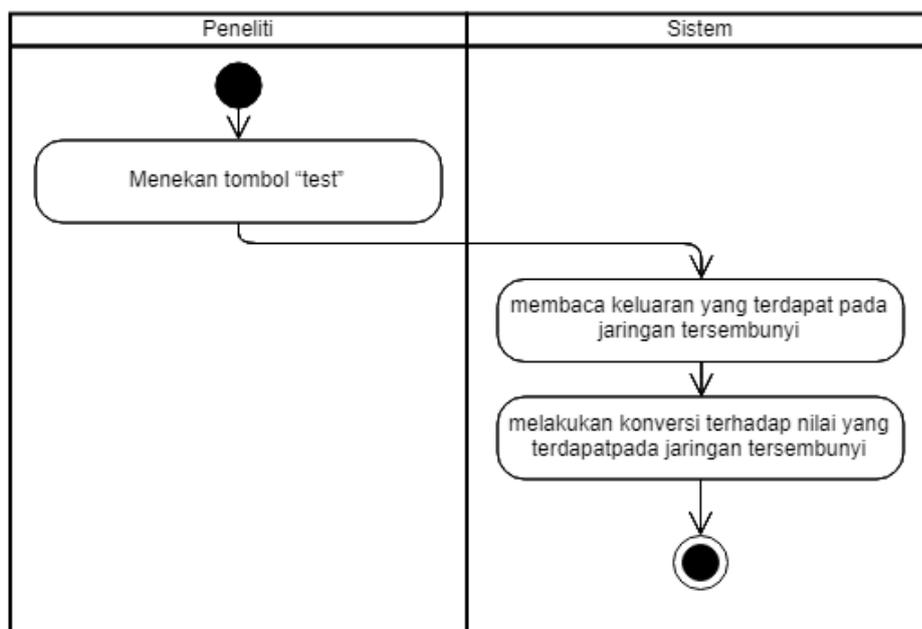
Gambar IV-7. Diagram Aktivitas *Use Case* Proses Pengujian



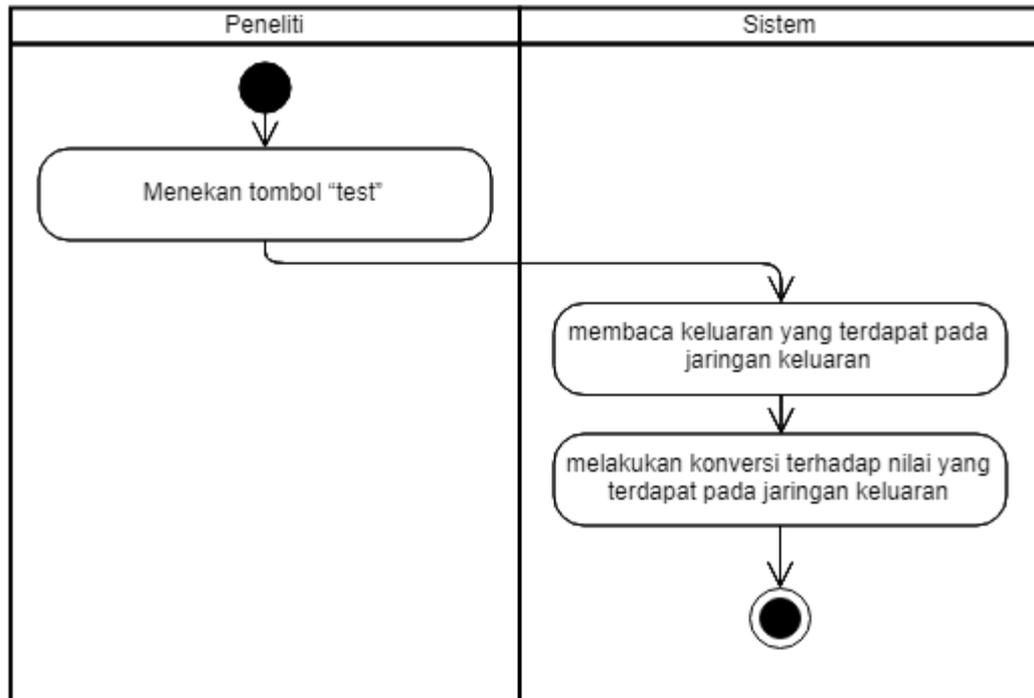
Gambar IV-8. Diagram Aktivitas *Use Case* Praproses data



Gambar IV-9. Diagram Aktivitas *Use Case* Distribusi Kunci



Gambar IV-10. Diagram Aktivitas *Use Case* Enkripsi



Gambar IV-11. Diagram Aktivitas *Use Case* Dekripsi

4.3 Fase Elaborasi

Tahapan ini meliputi proses perancangan data, tatap muka (*interface*), *sequence* diagram, dll yang akan digunakan dalam merancang perangkat lunak.

4.3.1 Pemodelan Bisnis

Subbab ini menjelaskan mengenai perancangan sistem yang dibangun yang meliputi perancangan data beserta tatap muka (*interface*) yang dilandasi oleh hasil analisa yang didapatkan dalam tahapan sebelumnya.

4.3.1.1 Perancangan Data

Data yang digunakan dalam penelitian ini dan akan dimasukkan ke dalam jaringan adalah data teks yang hanya mengandung karakter alfabetik saja. Data tersebut bersifat *offline* yang diambil melalui situs <http://www.textfiles.com/directory.html>.

4.3.1.2 Perancangan Antarmuka

Antarmuka (*interface*) yang dibangun merupakan rancangan antarmuka pada proses pelatihan yang mencakup praproses data dan distribusi kunci serta proses pengujian yang mencakup proses enkripsi dan dekripsi. Rancangan antarmuka sistem yang dibangun ditunjukkan oleh gambar berikut ini:

The image shows a graphical user interface (GUI) for a system. On the left side, there is a text input area with a 'type text' label and a 'browse' button. Below this are three input fields: 'enter key', 'number of epoch', and 'activation function' (with a dropdown arrow). There are also 'clear', 'train', 'restart', and 'test' buttons. On the right side, there is a large rectangular area with four tabs: 'details', 'epoch', 'encryption', and 'decryption'. Below this area are two smaller rectangular boxes labeled 'Cipher Text' and 'PlainText'.

Gambar IV-12. Rancangan Antarmuka

4.3.2 Kebutuhan Sistem

Dalam proses pengembangan perangkat lunak menggunakan Bahasa pemrograman *Java* yang bersifat *Object Oriented Programming* (OOP). Spesifikasi perangkat keras yang digunakan dalam pengembangan perangkat lunak antara lain:

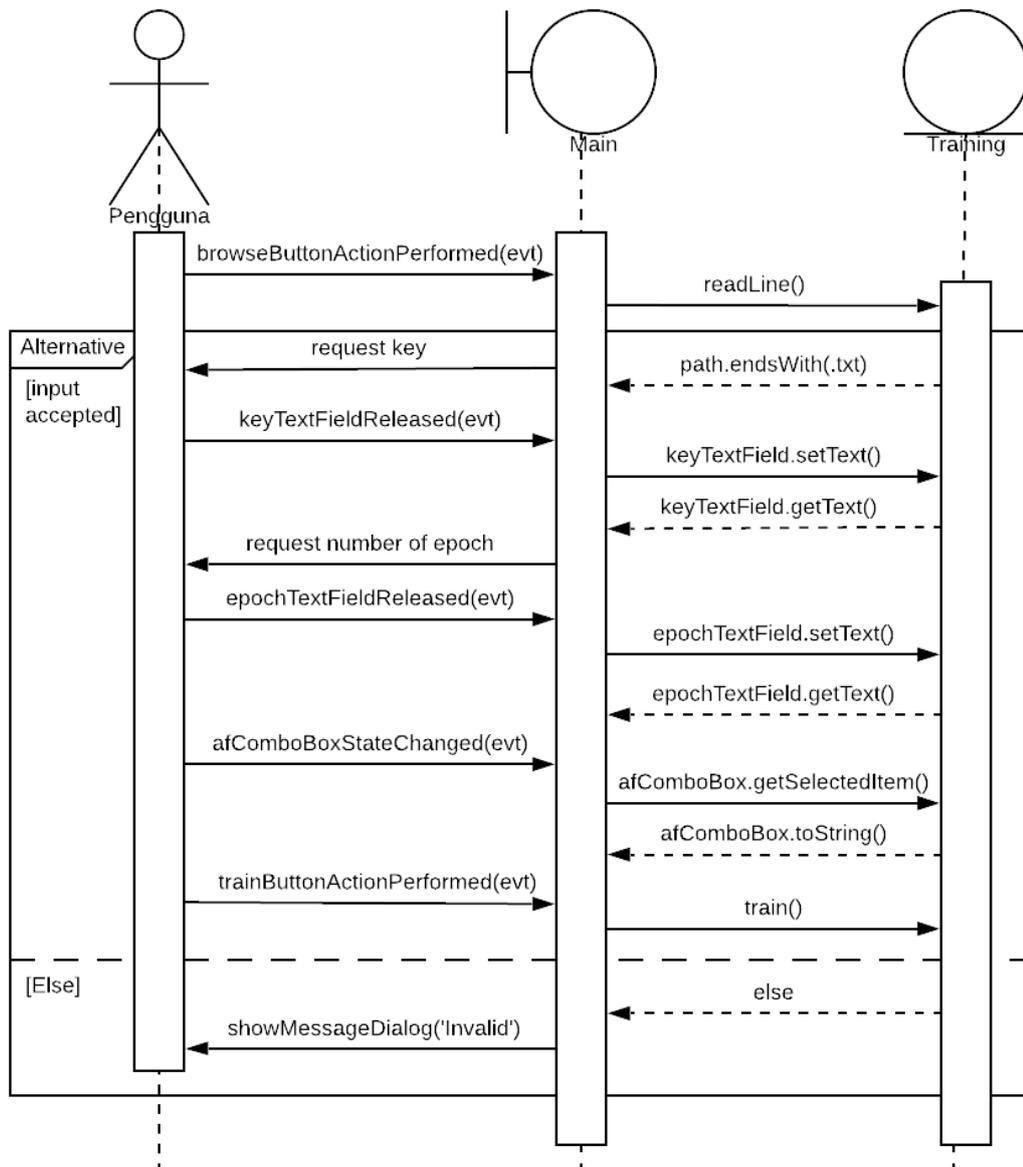
1. Laptop merk Toshiba model *Satelite* E45t-B
2. *Processor* Intel® core™ i5-5200 CPU @2.20GHz
3. RAM 4 GB
4. *Hard disk* 1 TB

Perangkat lunak yang digunakan dalam pengembangan perangkat lunak antara lain:

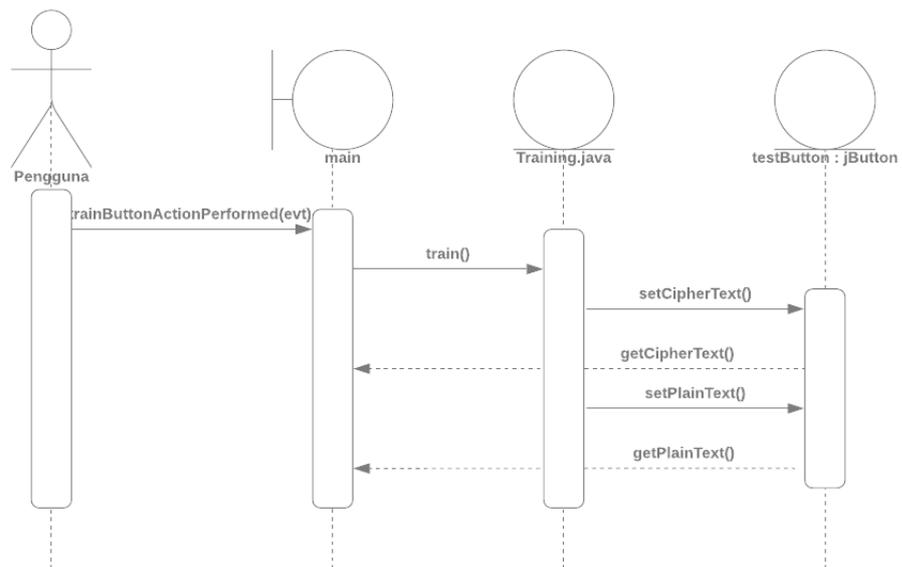
1. Sistem operasi *Windows* 8.1 64-bit
2. NetBeans IDE 8.2
3. Java 1.8.0_171

4.3.3 Sequence Diagram

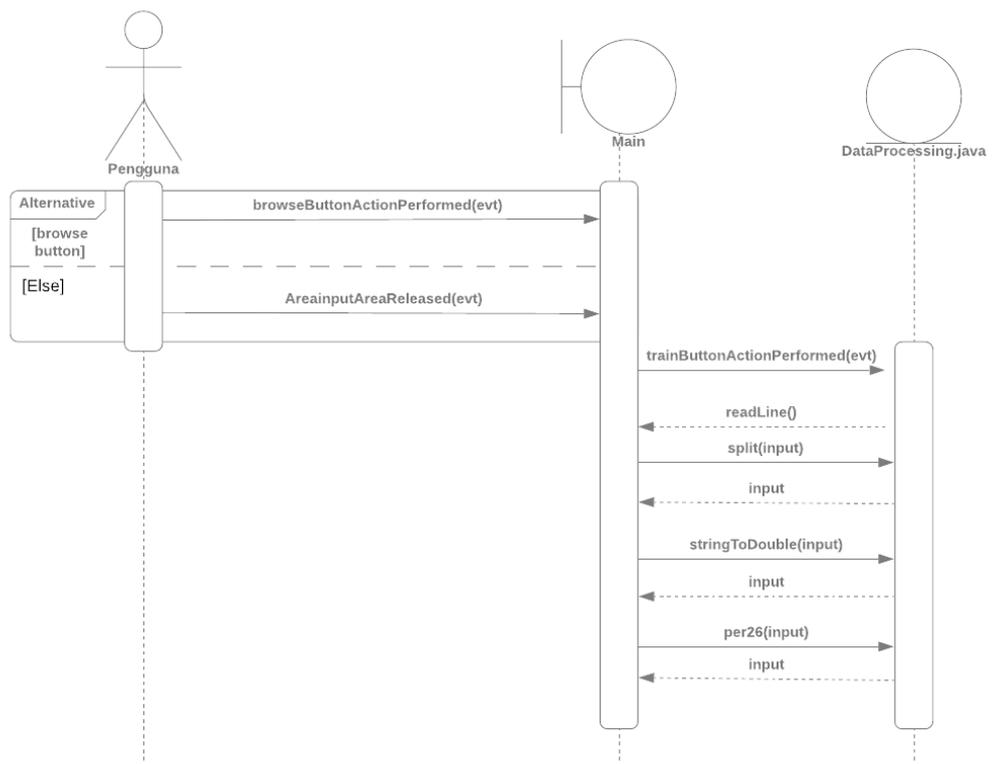
Sequence diagram merupakan diagram yang menggambarkan urutan kejadian antara objek dan kelas yang digunakan dalam pengembangan perangkat lunak. *Usecase* diagram yang terdapat pada pembahasan ini adalah *usecase* diagram pelatihan dan *usecase* diagram pengujian.



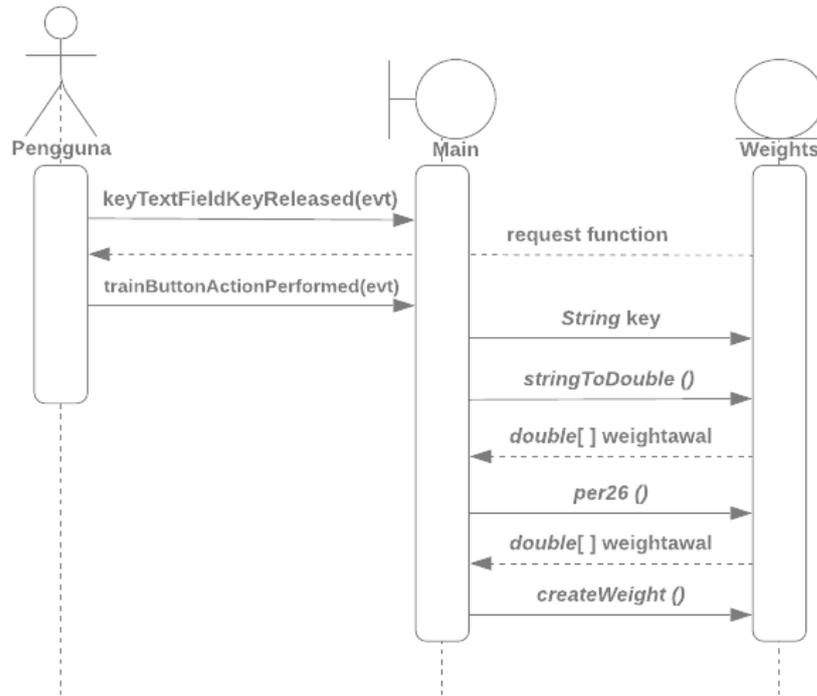
Gambar IV-13. Diagram Sequence Pelatihan



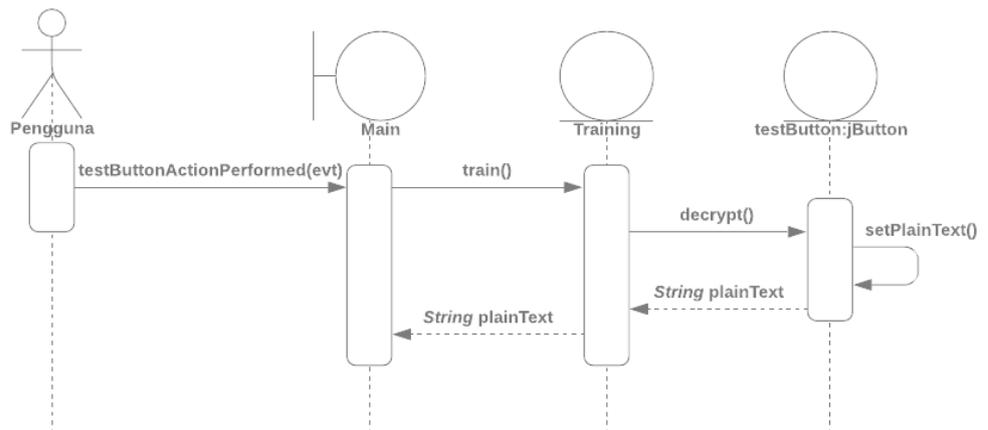
Gambar IV-14. Diagram *Sequence* Pengujian



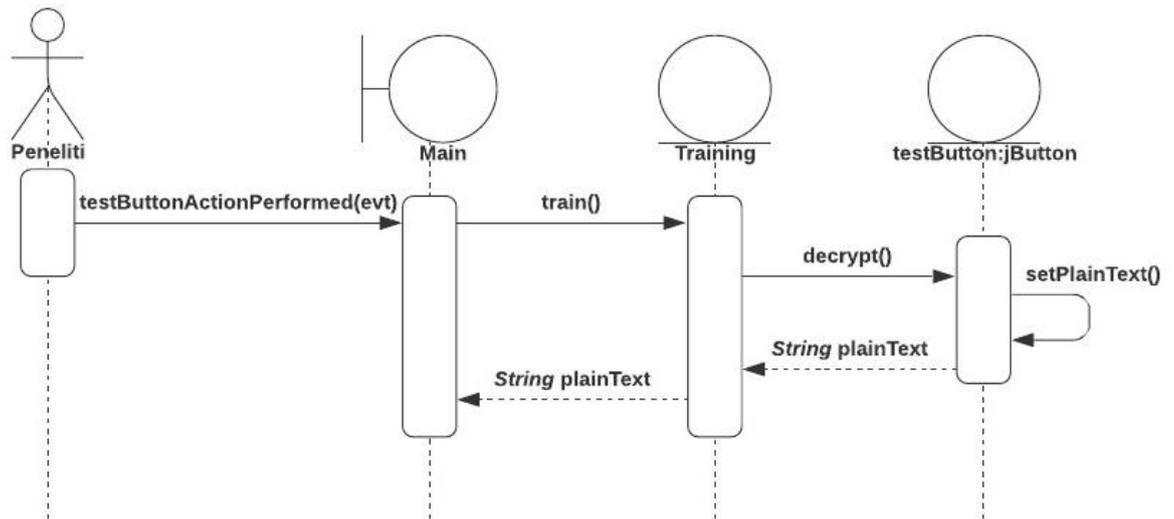
Gambar IV-15. Diagram *Sequence* Praproses Data



Gambar IV-16. Diagram *Sequence* Distribusi Kunci



Gambar IV-17. Diagram *Sequence* Enkripsi



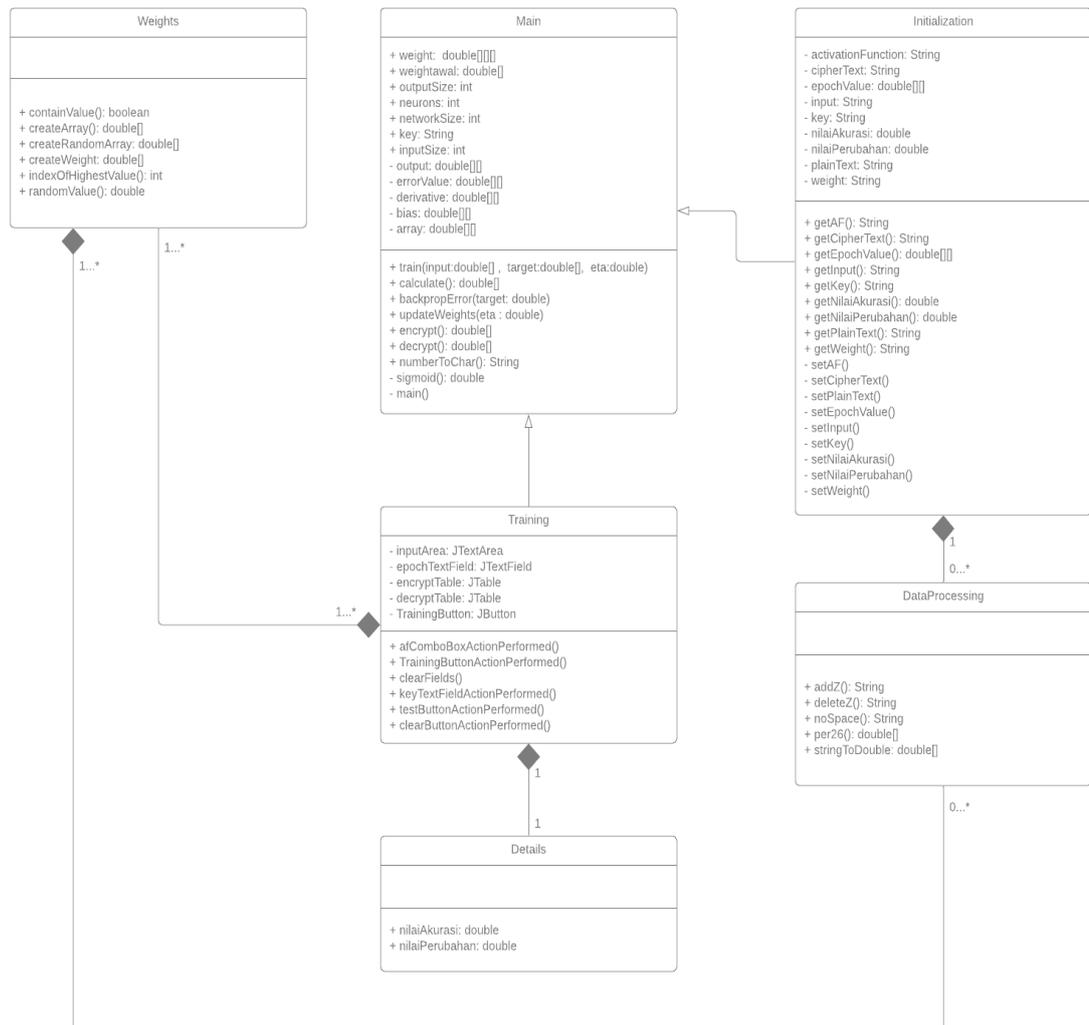
Gambar IV-18. Diagram *Sequence* Dekripsi

4.3 Fase Konstruksi

Fase ini merupakan tahapan pembuatan atau konstruksi pengembangan perangkat lunak berdasarkan desain dan rancangan yang telah didefinisikan pada tahapan awal pengembangan perangkat lunak.

4.4.1 Diagram Kelas (*class diagram*)

Penggunaan diagram kelas ialah untuk menjelaskan hubungan yang terjadi antara kelas yang dibuat dengan sistem. Berikut merupakan diagram kelas yang dipakai dalam pengembangan perangkat keras ini:



Gambar IV-19. Kelas Diagram

4.4.3 Implementasi

Tahapan ini merupakan tahapan pengembangan berupa implementasi terhadap rancangan antarmuka, *use case*, diagram kelas, dan pendefinisian kebutuhan lain yang telah ditentukan pada fase sebelumnya.

4.4.3.1 Implementasi Kelas

Pengembangan perangkat lunak ini menggunakan pendekatan berorientasi objek dengan menggunakan Bahasa pemrograman java. Proses pengembangan berdasarkan kelas-kelas yang dibentuk serta telah dihubungkan satu sama lain dengan menggunakan kelas diagram. Tabel berikut menjelaskan mengenai kelas-kelas yang dipakai serta penggunaannya dalam pengembangan perangkat lunak ini.

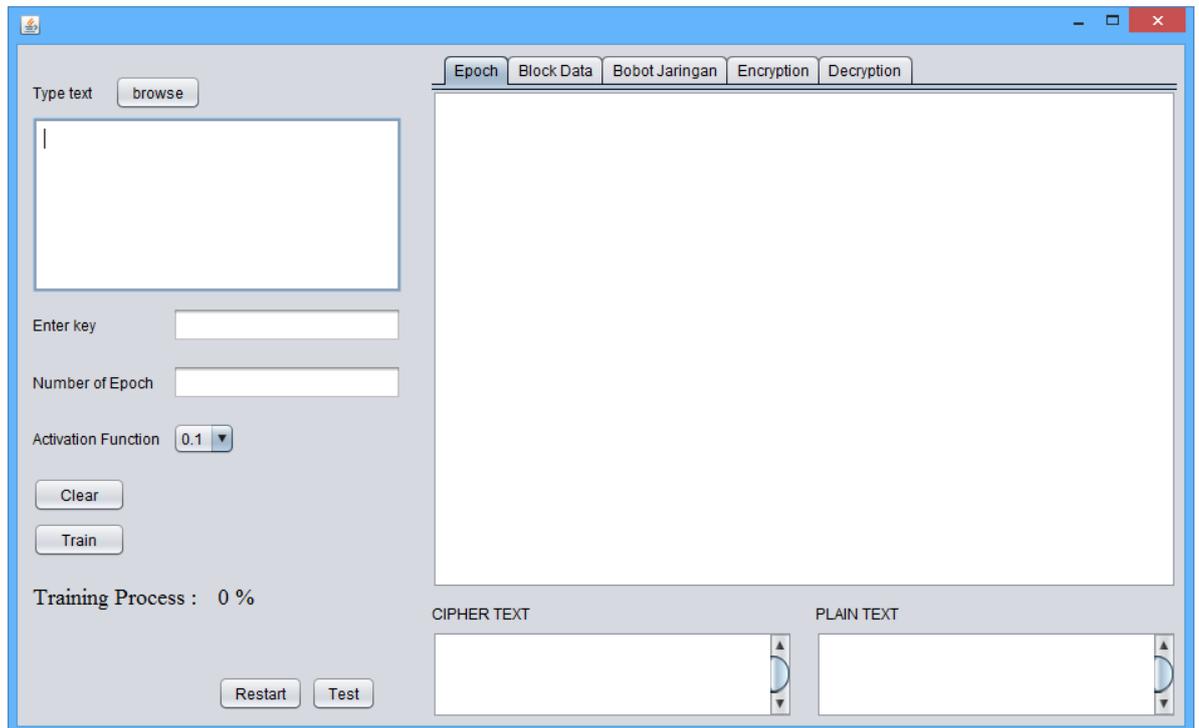
Table IV-14. Implementasi Kelas

No.	Kelas	Nama file	Keterangan
1	Neural Network	NeuralNetwork.java	Kelas kontrol yang digunakan untuk membentuk serta mengatur struktur jaringan yang akan digunakan dalam algoritma.
2	Weights	Weights.java	Kelas kontrol yang digunakan untuk mengolah serta mengatur distribusi bobot yang dimasukkan sebagai kunci ke dalam jaringan yang telah dibentuk.

3	Data Processing	DataProcessing.java	Kelas kontrol yang dipakai untuk mengolah data agar dapat diterima oleh jaringan yang telah dibentuk.
4	Initialization	Initialization.java	Merupakan kelas yang dibentuk untuk menampung seluruh data baik yang belum diproses maupun data yang telah diproses oleh jaringan.
5	Details	Details.java	Kelas yang digunakan untuk mengolah detail informasi dari proses pelatihan dan pengujian yang telah dilakukan.
6	Training	Training.java	Kelas yang digunakan untuk menggabungkan metode yang dipakai serta mengatur proses pelatihan dan pengujian.

4.4.3.2 Implementasi Antarmuka

Antarmuka yang telah dirancang sebelumnya akan diimplementasikan dengan berdasarkan kelas-kelas yang telah ditentukan. Gambar berikut merupakan implementasi antarmuka perangkat lunak yang digunakan.



Gambar IV-20. Implementasi Antarmuka.

4.5 Fase Transisi

Fase transisi menitikberatkan pada proses pengujian (*testing*) terhadap perangkat lunak yang telah dibangun dengan membandingkan kebutuhan yang telah ditentukan pada fase-fase sebelumnya.

4.5.1 Pemodelan Bisnis

Pengujian yang akan dilakukan terhadap perangkat lunak adalah pengujian *black box* atau pengujian yang dilakukan terhadap hasil yang didapat dengan membandingkan dengan kebutuhan fungsional ataupun *use case diagram* yang telah didefinisikan sebelumnya terhadap data uji yang digunakan pada penelitian.

4.5.2 Kebutuhan Sistem

Dalam melakukan proses pengujian, kebutuhan perangkat yang digunakan dapat menggunakan perangkat yang sama dengan perangkat pengembangan perangkat lunak. Spesifikasi perangkat yang digunakan antara lain sebagai berikut:

1. Laptop merk Toshiba model *Satelite* E45t-B
2. *Processor* Intel® core™ i5-5200 CPU @2.20GHz
3. RAM 4 GB
4. *Hard disk* 1 TB

Sedangkan perangkat lunak (*software*) yang digunakan dalam proses pengujian adalah sebagai berikut:

1. Sistem operasi *Windows* 8.1 64-bit
2. NetBeans IDE 8.2
3. Java 1.8.0_171

4.5.3 Rencana Pengujian

Rencana pengujian merupakan rancangan uji yang akan digunakan dalam proses pengujian. Gambar IV-X menjelaskan mengenai rencana pengujian yang akan diterapkan terhadap perangkat lunak yang dibangun.

4.5.3.1 Rencana Pengujian *Use Case* Proses Pelatihan

Rencana pengujian proses pelatihan berdasarkan *use case* yang telah didefinisikan pada fase elaborasi dapat dilihat pada tabel berikut.

Tabel IV-15. Rencana Pengujian terhadap *Use Case* Pelatihan

No.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1	UC1-01	Pengujian sistem dengan menggunakan algoritma <i>backpropagation</i> dengan <i>epoch</i> sebanyak 10000.	<i>Black box</i>	Pengujian Unit
2	UC1-02	Pengujian sistem dengan bobot yang diambil dengan menggunakan kunci yang ditentukan oleh <i>pengguna</i> .	<i>Black box</i>	Pengujian Unit

4.5.3.2 Rencana Pengujian *Use Case* Proses Pengujian

Rencana pengujian proses pengujian berdasarkan *use case* yang telah didefinisikan pada fase elaborasi dapat dilihat pada tabel berikut.

Tabel IV-16. Rencana Pengujian terhadap *Use Case* Pengujian

No.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1	UC2-01	Pengujian sistem dengan membandingkan jumlah data sebelum dan sesudah pengujian.	<i>Black box</i>	Pengujian Unit

2	UC2-02	Pengujian sistem dengan membandingkan <i>cipher text</i> dengan masukan.	<i>Black box</i>	Pengujian Unit
---	--------	--	------------------	----------------

4.5.3.3 Rencana Pengujian *Use Case* Proses Praproses Data

Rencana pengujian proses praproses data berdasarkan *use case* yang telah didefinisikan pada fase elaborasi dapat dilihat pada tabel berikut.

Tabel IV-17. Rencana Pengujian terhadap *Use Case* Praproses Data

No.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1	UC3-01	Pengujian sistem dengan menggunakan masukan yang tidak berupa <i>String</i> atau file dengan ekstensi .txt.	<i>Black box</i>	Pengujian Unit
2	UC3-02	Pengujian sistem dengan masukan yang memiliki karakter bukan kelipatan 5.		

4.5.3.4 Rencana Pengujian *Use Case* Proses Distribusi Kunci

Rencana pengujian proses pengujian berdasarkan *use case* yang telah didefinisikan pada fase elaborasi dapat dilihat pada tabel berikut.

Tabel IV-18. Rencana Pengujian terhadap *Use Case* Distribusi Kunci

No.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1	UC4-01	Pengujian sistem dengan menggunakan kunci yang tidak berupa <i>String</i> .	<i>Black box</i>	Pengujian Unit
2	UC4-02	Pengujian sistem dengan menggunakan kunci yang berukuran sama dengan masukan.	<i>Black box</i>	Pengujian Unit
3	UC4-03	Pengujian sistem dengan kunci satu karakter.	<i>Black box</i>	Pengujian Unit

4.5.3.5 Rencana Pengujian *Use Case* Proses Enkripsi

Rencana pengujian proses pengujian berdasarkan *use case* yang telah didefinisikan pada fase elaborasi dapat dilihat pada tabel berikut.

Tabel IV-19. Rencana Pengujian terhadap *Use Case* Enkripsi

No.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1	UC5-01	Membandingkan <i>cipher text</i> yang didapat dengan target yang ditentukan.	<i>Black box</i>	Pengujian Unit

2	UC5-02	Pengujian sistem dengan membandingkan <i>cipher text</i> yang didapat dari beberapa kali pengujian dengan masukan dan kunci yang sama.	<i>Black box</i>	Pengujian Unit
3	UC5-03	Pengujian terhadap karakter tambahan “z” pada <i>cipher text</i> yang didapat.	<i>Black box</i>	Pengujian Unit

4.5.3.6 Rencana Pengujian Use Case Proses Dekripsi

Rencana pengujian proses pengujian berdasarkan *use case* yang telah didefinisikan pada fase elaborasi dapat dilihat pada tabel berikut.

Tabel IV-20. Rencana Pengujian terhadap *Use Case* Dekripsi

No.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1	UC6-01	Membandingkan <i>cipher text</i> dengan <i>plain text</i> yang didapat.	<i>Black box</i>	Pengujian Unit
2	UC6-02	Membandingkan <i>plain text</i> yang didapat dengan target yang ditentukan.	<i>Black box</i>	Pengujian Unit
3	UC6-03	Pengujian sistem dalam hal jumlah data yang didapat pada proses dekripsi data.	<i>Black box</i>	Pengujian Unit

4.5.4 Implementasi

Implementasi yang dilakukan berupa pengujian langsung terhadap sistem yang dibuat berdasarkan rancangan atau rencana pengujian yang telah dibuat pada tahapan sebelumnya.

4.5.4.1 Pengujian *Use Case* Proses Pelatihan.

Pengujian *use case* pada proses pelatihan ditunjukkan oleh tabel berikut.

Tabel IV-21. Pengujian *Use Case* Pelatihan

ID	Deskripsi	Langkah Pengujian	Keluaran yang diharapkan	Hasil sebenarnya	keterangan
UC1-01	Pengujian sistem dengan jumlah <i>epoch</i> yang ditentukan.	<ol style="list-style-type: none"> 1. memasukkan jumlah <i>epoch</i> yang diharapkan yaitu 10000. 2. menekan tombol “train”. 	Hasil pada setiap <i>neuron</i> di setiap jaringan setelah 10000 kali proses perulangan.	Sistem melakukan proses perulangan sebanyak 10000 kali.	Terpenuhi
UC1-02	Pengujian sistem dengan menggunakan bobot berdasarkan kunci yang dimasukkan.	<ol style="list-style-type: none"> 1. Memasukkan kunci yang akan digunakan 2. Menekan tombol “train”. 	Bobot pada neuron sesuai dengan urutan karakter pada kunci yang dipakai.	Jaringan yang dibangun mempunyai bobot sesuai dengan kunci yang dipakai dan	Terpenuhi.

				berurutan pada setiap jaringan.	
--	--	--	--	---------------------------------	--

4.5.4.2 Pengujian *Use Case* Proses Pengujian.

Pengujian *use case* pada proses pengujian ditunjukkan oleh tabel berikut.

Tabel IV-22. Pengujian *Use Case* pengujian

ID	Deskripsi	Langkah Pengujian	Keluaran yang diharapkan	Hasil sebenarnya	keterangan
UC2-01	Pengujian sistem dengan melakukan perhitungan terhadap jumlah data yang didapat pada proses pengujian.	Menekan tombol pengujian.	Jumlah data pada setiap proses sama banyak.	Jumlah data masukan, <i>cipher text</i> , dan <i>plain text</i> yang didapat sama.	Terpenuhi

UC2-02	Pengujian sistem dengan membandingkan <i>cipher text</i> dengan masukan.	Menekan tombol pengujian.	<i>Cipher text</i> yang didapat pada setiap karakter berbeda dengan masukan.	Setiap karakter pada masukan tidak ada kesamaan terhadap <i>cipher text</i> yang didapat.	Terpenuhi.
--------	--	---------------------------	--	---	------------

4.5.4.3 Pengujian *Use Case* Proses Praproses Data.

Pengujian *use case* pada proses praproses data ditunjukkan oleh tabel berikut.

Tabel IV-23. Pengujian *Use Case* Praproses Data

ID	Deskripsi	Langkah Pengujian	Keluaran yang diharapkan	Hasil sebenarnya	keterangan
UC3-01	Pengujian sistem untuk menguji tipe data atau ekstensi file yang dapat diterima oleh sistem.	1. memasukkan masukan <i>non-string</i> atau ekstensi file bukan <i>.txt</i> . 2. menekan tombol “ <i>train</i> ”.	Sistem menampilkan pesan bahwa masukan tidak dapat diproses.	Pelatihan tidak dapat dilakukan dan sistem menampilkan <i>pop-up message</i> .	Terpenuhi

UC3-02	Pengujian sistem terhadap masukan yang memiliki jumlah karakter bukan merupakan kelipatan 5.	1. memasukkan masukan <i>string</i> yang berjumlah bukan kelipatan lima 2. menekan tombol “ <i>train</i> ”.	Data terakhir yang diproses oleh jaringan memiliki karakter tambahan “z”.	Terdapat penambahan karakter “z” sesuai dengan slot kosong pada blok terakhir.	Terpenuhi.
--------	--	--	---	--	------------

4.5.4.4 Pengujian *Use Case* Proses Distribusi Kunci.

Pengujian *use case* pada proses distribusi kunci ditunjukkan oleh tabel berikut.

Tabel IV-24. Pengujian *Use Case* distribusi kunci

ID	Deskripsi	Langkah Pengujian	Keluaran yang diharapkan	Hasil sebenarnya	keterangan
UC4-01	Pengujian sistem terhadap kunci <i>non-string</i>	1. memasukkan kunci <i>non-string</i> 2. menekan tombol “ <i>train</i> ”.	Sistem tidak dapat memproses kunci yang dimasukkan.	Sistem menampilkan <i>pop-up message</i> bahwa kunci salah.	Terpenuhi

UC4-02	Pengujian sistem dengan jumlah kunci melebihi jumlah bobot yang tersedia.	1. memasukkan kunci dengan jumlah karakter melebihi jumlah bobot yang tersedia. 2. menekan tombol “ <i>train</i> ”.	Sisa kunci yang tersedia tidak terpakai.	Hanya karakter pada kunci yang berkesuaian terhadap jumlah bobot tersedia saja yang diambil oleh jaringan.	Terpenuhi.
UC4-03	Pengujian sistem dengan nilai bobot yang sama.	1. memasukkan kunci dengan satu karakter. 2. menekan tombol “ <i>train</i> ”.	Sistem tetap dapat melakukan proses pelatihan.	Proses pelatihan tetap berjalan dengan nilai bobot yang sama pada setiap jaringan yang ada..	Terpenuhi.

4.5.4.5 Pengujian *Use Case* Proses Enkripsi.

Pengujian *use case* pada proses enkripsi ditunjukkan oleh tabel berikut.

Tabel IV-25. Pengujian *Use Case* Enkripsi

ID	Deskripsi	Langkah Pengujian	Keluaran yang diharapkan	Hasil sebenarnya	keterangan
UC5-01	Pengujian sistem terhadap masukan awal yang diberikan.	Menekan tombol enkripsi	Setiap karakter pada <i>cipher text</i> berbeda dengan setiap karakter yang berkesuaian pada masukan awal.	Semua karakter masukan berubah pada proses enkripsi.	Terpenuhi
UC5-02	Pengujian sistem terhadap proses enkripsi berulang dengan masukan dan kunci yang sama.	Melakukan beberapa kali proses pelatihan dan pengujian dengan masukan dan kunci yang sama.	Menghasilkan <i>cipher text</i> yang berbeda-beda	Setiap <i>cipher text</i> yang dihasilkan selalu menghasilkan kombinasi karakter yang berbeda pada tiap proses.	Terpenuhi.

UC5-03	Pengujian sistem terhadap <i>cipher text</i> pada karakter tambahan “z”	Melakukan beberapa kali proses pelatihan dan pengujian dengan masukan dan kunci yang sama yang akan menghasilkan karakter tambahan “z”.	Karakter tambahan “z” selalu berubah menjadi karakter yang berbeda setiap proses enkripsi.	<i>Cipher text</i> yang dihasilkan pada beberapa kali percobaan menampilkan karakter yang berbeda ketika karakter tambahan “z” dienkripsi.	Terpenuhi.
--------	---	---	--	--	------------

4.5.4.6 Pengujian *Use Case* Proses Dekripsi.

Pengujian *use case* pada proses dekripsi ditunjukkan oleh tabel berikut.

Tabel IV-26. Pengujian *Use Case* dekripsi

ID	Deskripsi	Langkah Pengujian	Keluaran yang diharapkan	Hasil sebenarnya	keterangan
----	-----------	-------------------	--------------------------	------------------	------------

UC6-01	Pengujian sistem terhadap <i>plain text</i> yang dihasilkan dengan <i>cipher text</i> .	Menekan tombol dekripsi	<i>Plain text</i> yang dihasilkan berbeda dengan <i>cipher text</i> yang ada.	Setiap karakter pada <i>cipher text</i> berbeda dengan <i>plain text</i> yang berkesesuaian yang dihasilkan.	Terpenuhi
UC6-02	Pengujian sistem terhadap <i>plain text</i> yang dihasilkan dengan masukan awalan.	Menekan tombol dekripsi	<i>Plain text</i> yang dihasilkan sama dengan masukan awal yang diberikan.	<i>Plain text</i> yang dihasilkan tidak sepenuhnya berhasil mengenali masukan awal yang diberikan.	Tidak Terpenuhi.
UC6-03	Pengujian sistem terhadap jumlah data pada tiap proses.	Melakukan proses pelatihan dan pengujian.	Jumlah data pada tiap proses enkripsi dan dekripsi sama dengan masukan awal yang diberikan.	Jumlah data konsisten atau sama banyak pada tiap proses.	Terpenuhi.

4.6 Kesimpulan

Proses implementasi algoritma pada perangkat lunak atau sistem yang dibangun menggunakan metode pembangunan perangkat lunak RUP. Proses diawali dengan fase inepsi yang berfokus kepada pendefinisian kebutuhan sistem yang mencakupi proses analisa serta desain. Fase lanjutan dari inepsi adalah fase elaborasi yang merupakan lanjutan perancangan sistem seperti membangun antarmuka (*interface*) dan diagram yang diperlukan. Fase pengembangan yang ketiga adalah fase kontruksi yang berupa pengimplementasian terhadap rancangan yang dibangun yaitu tahapan *coding*. Fase yang terakhir adalah fase transisi yang berfokus kepada rancangan pengujian dan proses pengujian terhadap sistem yang dibangun terhadap kebutuhan yang telah didefinisikan pada fase awal pengembangan sistem.