

**PENERAPAN JARINGAN SARAF PROPAGASI BALIK PADA
KRIPTOGRAFI SIMETRI UNTUK DATA TEKS**

Diajukan Sebagai Syarat Untuk Menyelesaikan Program Strata-1 Pada
Jurusan Teknik Informatika



Oleh:

MUHAMMAD YUSUF

09021381520043

**Jurusan Teknik Informatika
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA
2019**

LEMBAR PENGESAHAN TUGAS AKHIR

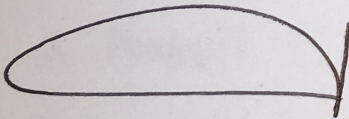
PENERAPAN JARINGAN SARAF PROPAGASI BALIK PADA KRIPTOGRAFI SIMETRI UNTUK DATA TEKS

Oleh:

MUHAMMAD YUSUF
NIM : 09021381520043

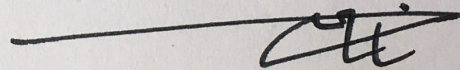
Palembang, Juli 2019

Pembimbing I,



Drs. Megah Mulya, M.T
NIP. 196602202006041001

Pembimbing II,



Osvari Arsalan, M. T
NIP. 198806282018031001

Mengetahui,
Ketua Jurusan Teknik Informatika,



Rifkie Primartha, MT
NIP. 197706012009121004

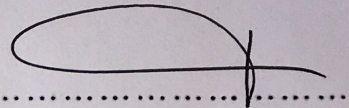
TANDA LULUS UJIAN SIDANG TUGAS AKHIR

Pada hari Jum'at tanggal 18 Juli 2019 telah dilaksanakan ujian sidang tugas akhir oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

Nama : Muhammad Yusuf
NIM : 09021381520043
Judul : Penerapan Jaringan Saraf Propagasi Balik pada Kriptografi Simetri untuk Data Teks

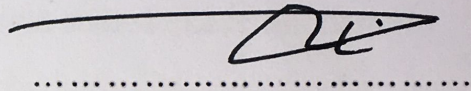
1. Pembimbing I

Drs. Megah Mulya, M. T
NIP. 196602202006041001



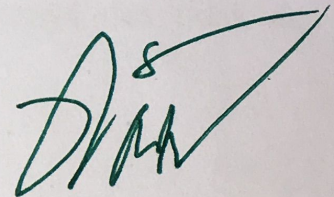
2. Pembimbing II

Osvari Arsalan, M. T
NIP. 198806282018031001



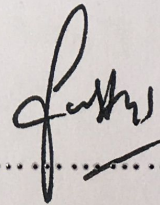
3. Penguji I

Samsuryadi, M. Kom, Ph. D
NIP. 197102041997021003



4. Penguji II

Desty Rodiah, M. T
NIP.



Mengetahui,
Ketua Jurusan Teknik Informatika



Rifkie Primartha, MT
NIP. 197706012009121004

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Yusuf
NIM : 09021381520043
Program Studi : Teknik Informatika
Judul Skripsi : Penerapan Jaringan Saraf Propagasi Balik
pada Kriptografi Simetri untuk Data Teks
Hasil Pengecekan Software
iThenticate/Turnitin : 6 %

Menyatakan bahwa Laporan Proyek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan proyek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.

Palembang, 24 Juli 2019



Muhammad Yusuf
NIM. 09021381520043

MOTTO DAN PERSEMBAHAN

MOTTO:

Being Youth is a Privilege That You
Have Once in A Lifetime

*Kupersembahkan karya tulis ini
kepada :*

*Orang tuaku tercinta
dan saudaraku tersayang*

Keluarga besarku

Sahabat dan teman seperjuanganku

Fakultas Ilmu Komputer

Universitas Sriwijaya

DAFTAR ISI

Halaman

<u>HALAMAN JUDUL</u>	i
ABSTRACT	ii
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR TABEL	vii
DAFTAR GAMBAR	ix

BAB I PENDAHULUAN

1.1 Pendahuluan	I-1
1.2 Latar Belakang Masalah	I-1
1.3 Rumusan Masalah	I-3
1.4 Tujuan Penelitian	I-4
1.5 Manfaat Penelitian	I-5
1.6 Batasan Penelitian	I-5
1.7 Objektif Penelitian	I-6
1.8 Sistematika Penulisan	I-6
1.9 Kesimpulan	I-8

BAB II KAJIAN LITERATUR

2.1 Pengantar	II-1
2.2 Data	II-2
2.2.1 Data Teks	II-2
2.3 Kriptografi	II-3
2.3.1 Kriptografi Simetris	II-5
2.3.2 Jenis-Jenis Kriptografi Simetris	II-6
2.3.3 Kriptografi Simetris untuk Enkripsi Teks	II-11
2.3.4 Kriptografi Asimetris	II-12
2.3.5 Jenis-Jenis Kriptografi Simetris	II-13
2.4 Perbandingan antara Kriptografi Simetris dan Asimetris	II-15
2.5 Kelemahan Kriptografi Saat Ini	II-19
2.6 Jaringan Saraf Tiruan	II-19
2.6.1 Jaringan Saraf Tiruan <i>Single-layer</i> dan <i>Multi-layer</i>	II-20
2.6.2 Jaringan Saraf Tiruan <i>backpropagation</i> pada Kriptografi	II-22
2.7 Penelitian Terkait	II-32
2.8 Kesimpulan	II-34

BAB III METODOLOGI PENELITIAN

3.1	Pendahuluan	III-1
3.2	Unit Penelitian	III-2
3.3	Metode Pengumpulan Data	III-2
3.3.1	Jenis dan Sumber Data	III-2
3.3.2	Metode Pengumpulan Data	III-3
3.3.3	Masukan yang digunakan	III-3
3.3.4	Penggunaan Kunci	III-4
3.4	Metode Penelitian / Kerangka Kerja	III-5
3.4.1	Fase 1 : Identifikasi Masalah	III-7
3.4.2	Fase 2 : Penerapan Metode	III-7
3.4.3	Fase 2 : Evaluasi Penelitian	III-8
3.5	Metode Pengembangan Perangkat Lunak	III-10
3.5.1	Fase Insepsi	III-11
3.5.2	Fase Elaborasi	III-11
3.5.3	Fase Konstruksi	III-12
3.5.4	Fase Transisi	III-12
3.6	Manajemen Proyek Penelitian	III-13

BAB IV PENGEMBANGAN PERANGKAT LUNAK

4.1	Pendahuluan	IV-1
4.2	Fase Insepsi	IV-2
4.2.1	Permodelan Bisnis	IV-2
4.2.2	Kebutuhan Sistem	IV-3
4.2.3	Analisis dan Desain	IV-6
4.3	Fase Elaborasi	IV-32
4.3.1	Permodelan Bisnis	IV-32
4.3.2	Kebutuhan Sistem	IV-34
4.3.3	Diagram <i>Sequence</i>	IV-34
4.4	Fase Konstruksi	IV-38
4.4.1	Diagram Kelas	IV-41
4.4.3	Implementasi	IV-40
4.5	Fase Transisi	IV-42
4.5.1	Permodelan Bisnis	IV-42
4.5.2	Kebutuhan Sistem	IV-43
4.5.3	Rencana Pengujian	IV-43
4.5.4	Implementasi	IV-48
4.6	Kesimpulan	IV-58

BAB V HASIL DAN ANALISIS PENELITIAN

5.1 Pendahuluan	V-1
5.2 Data Hasil Percobaan	V-2
5.2.1 Hasil Pengujian Data	V-1
5.3 Analisis Hasil Penelitian	V-5
5.3.1 Analisis Hasil Akurasi Penelitian	V-5
5.3.2 Analisis Waktu Pemrosesan	V-8
5.4 Kesimpulan	V-9

BAB VI KESIMPULAN DAN SARAN

6.1 Pendahuluan	VI-1
6.2 Kesimpulan	VI-1
6.3 Saran	VI-2

DAFTAR PUSTAKA	xi
----------------------	----

LAMPIRAN	xiii
----------------	------

DAFTAR TABEL

Halaman

II-1.	Perbandingan spesifikasi kriptografi kunci Simetris & Asimetris	II-15
II-2.	Penelitian terkait penerapan jaringan saraf tiruan pada kriptografi ...	II-24
III-1.	Daftar Data Uji	III-3
III-2.	Contoh Ukuran Kunci	III-4
III-3.	Rancangan Tabel Hasil Ekstraksi Ciri Setiap Data Pengujian.....	III-8
III-4.	Tabel Penjadwalan Penelitian	III-13
IV-1.	Kebutuhan Fungsional	IV-5
IV-2.	Kebutuhan Non Fungsional.....	IV-6
IV-3.	Tabel konversi data masukan	IV-8
IV-4.	Bobot antara lapisan masukan dan lapisan tersembunyi.....	IV-9
IV-5.	Bobot antara lapisan tersembunyi dan lapisan keluaran	IV-10
IV-6.	Definisi Aktor.....	IV-19
IV-7.	Definisi <i>Use Case</i>	IV-20
IV-8.	Skenario <i>Use Case</i> Proses Pelatihan	IV-22
IV-9.	Skenario <i>Use Case</i> Proses Pengujian	IV-23
IV-10.	Skenario <i>Use Case</i> Praproses Data	IV-24
IV-11.	Skenario <i>Use Case</i> Proses Distribusi kunci	IV-26
IV-12.	Skenario <i>Use Case</i> Proses Enkripsi	IV-27
IV-13.	Skenario <i>Use Case</i> Proses Dekripsi	IV-28
IV-14.	Implementasi Kelas	IV-40
IV-15.	Rencana Pengujian terhadap <i>Use Case</i> Pelatihan	IV-44
IV-16.	Rencana Pengujian terhadap <i>Use Case</i> Pengujian	IV-44
IV-17.	Rencana Pengujian terhadap <i>Use Case</i> Praproses Data	IV-45
IV-18.	Rencana Pengujian terhadap <i>Use Case</i> Distribusi Kunci	IV-46
IV-19.	Rencana Pengujian terhadap <i>Use Case</i> Enkripsi	IV-46
IV-20.	Rencana Pengujian terhadap <i>Use Case</i> Dekripsi	IV-47
IV-21.	Pengujian <i>Use Case</i> Pelatihan.....	IV-49
IV-22.	Pengujian <i>Use Case</i> pengujian.....	IV-50
IV-23.	Pengujian <i>Use Case</i> Praproses Data	IV-51
IV-24.	Pengujian <i>Use Case</i> distribusi kunci.....	IV-52
IV-25.	Pengujian <i>Use Case</i> Enkripsi	IV-54
IV-26.	Pengujian <i>Use Case</i> dekripsi.....	IV-55
IV-27.	Pengujian <i>Use Case</i> Praproses Wajah.....	IV-57
V-1.	Hasil Pengujian Data Uji.....	V-2
V-2.	Nilai Akurasi Plain Text.....	V-6
V-3.	Nilai Akurasi Cipher Text	V-6
V-4.	Tabel Hasil Percobaan Enkripsi dan Dekripsi	V-8

DAFTAR GAMBAR

Halaman

II-1.	Alur dari kriptografi	II-4
II-2.	Kriptografi Simetris	II-5
II-3.	Tipe-tipe kriptografi	II-7
II-4.	Arsitektur algoritma <i>blowfish</i>	II-10
II-5.	Kriptografi Asimetris	II-12
II-6.	Jaringan dengan lapisan tersembunyi.....	II-21
II-7.	Struktur yang terdapat pada suatu <i>neuron</i>	II-22
III-1.	Kerangka kerja penelitian	III-6
III-2.	Arsitektur RUP	III-10
IV-1.	Siklus RUP	IV-1
IV-2.	Arsitektur Jaringan Saraf Tiruan	IV-12
IV-3.	Fungsi Aktivasi <i>Sigmoid Biner</i>	IV-16
IV-4.	Proses Enkripsi dan Dekripsi menggunakan BPNN	IV-18
IV-5.	Diagram <i>Use Case</i>	IV-19
IV-6.	Diagram Aktivitas <i>Use Case</i> Proses Pelatihan	IV-29
IV-7.	Diagram Aktivitas <i>Use Case</i> Proses Pengujian	IV-30
IV-8.	Diagram Aktivitas <i>Use Case</i> Praproses data.....	IV-30
IV-9.	Diagram Aktivitas <i>Use Case</i> Distribusi Kunci	IV-31
IV-10.	Diagram Aktivitas <i>Use Case</i> Enkripsi	IV-31
IV-11.	Diagram Aktivitas <i>Use Case</i> Dekripsi	IV-32
IV-12.	Rancangan Antarmuka	IV-33
IV-13.	Diagram <i>Sequence</i> Pelatihan.....	IV-35
IV-14.	Diagram <i>Sequence</i> Pengujian.....	IV-36
IV-15.	Diagram <i>Sequence</i> Praproses Data.....	IV-36
IV-16.	Diagram <i>Sequence</i> Distribusi Kunci	IV-37
IV-17.	Diagram <i>Sequence</i> Enkripsi	IV-37
IV-18.	Diagram <i>Sequence</i> Dekripsi	IV-38
IV-19.	Kelas Diagram.....	IV-39
IV-20.	Implementasi Antarmuka	IV-42
V-1.	Grafik Hasil Akurasi Deskripsi Data	V-7
V-2.	Grafik Hasil Akurasi Enkripsi Data	V-7

PENERAPAN JARINGAN SARAF PROPAGASI BALIK PADA KRIPTOGRAFI SIMETRI UNTUK DATA TEKS

Oleh:

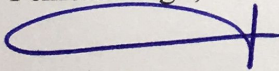
Muhammad Yusuf
09021381520043

ABSTRAK

Penelitian mengenai keamanan data telah banyak dilakukan dan dikembangkan dengan berbagai metode keamanan. Salah satu teknik pengamanan data yang ada yaitu kriptografi yang merupakan metode mengubah data asli menjadi data lain yang tidak memiliki arti bagi pihak yang tidak memiliki akses terhadap data. Terdapat dua metode kriptografi yaitu kriptografi kunci simetri dan kunci tidak simetri yang memiliki perbedaan pada jumlah kunci yang digunakan. Pada penelitian ini, metode kriptografi kunci simetri akan dikombinasikan dengan salah satu bidang kecerdasan buatan yaitu jaringan saraf propagasi balik (*backpropagation*) dengan tujuan menambah variasi keamanan pada metode kriptografi kunci simetri. Oleh karena itu, muncul pertanyaan bagaimana proses penggabungan dua metode tersebut dan menyesuaikan data yang digunakan terhadap sistem. Percobaan pada penelitian ini menggunakan data uji sebanyak 10 data teks. Hasil percobaan yang telah dilakukan pada penelitian ini menunjukkan bahwa sistem yang dibangun dapat menghasilkan *cipher text* yang berbeda-beda di setiap percobaan dengan nilai perbedaan terhadap data asli sebesar 85% serta akurasi enkripsi berada pada *range* 50% hingga 80%.

Kata kunci : Kriptografi simetri, jaringan saraf propagasi balik, data teks.

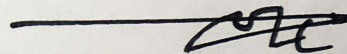
Pembimbing I,



Drs. Megah Mulya, M.T
NIP. 196602202006041001

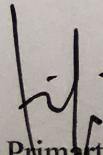
Palembang, Juli 2019

Pembimbing II,



Osvari Arsalan, M. T
NIP. 198806282018031001

Mengetahui,
Ketua Jurusan Teknik Informatika,



Rifkie Primartha, MT
NIP. 197706012009121004

BAB I

PENDAHULUAN

1.1 Pendahuluan

Pada bab ini akan membahas mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan objektif penelitian, serta struktur penulisan yang terdapat dalam penelitian yang digunakan sebagai gambaran secara umum mengenai penelitian yang akan dilakukan.

1.2 Latar Belakang Masalah

Ada beberapa macam data yang dapat diolah dan diproses oleh manusia untuk dapat menghasilkan sebuah informasi. Sebagai contohnya yaitu tulisan, gambar, suara, jaringan, dan macam-macam tipe data lainnya yang dapat diproses dan diolah oleh manusia. Menurut Tripati dan Agrawal (2014) bahwa kerahasiaan dari pengguna teknologi harus dan dapat dijaga sebagaimana keamanan serta privasi data dapat disediakan untuk pengguna. Perkembangan aplikasi berbasis internet meningkat pesat seiring dengan pentingnya mengamankan pengiriman pesan melalui jaringan tersebut (Gupta & Mittal, 2014).

Keamanan merupakan salah satu aspek yang krusial yang harus diperhatikan dalam hal pengolahan dan pemrosesan data. Rabah (2005) mengungkapkan bahwa masalah keamanan merupakan prioritas teratas yang akan dilakukan dalam mendesain sistem di masa depan. Untuk mengetahui serta menentukan jenis keamanan yang akan diterapkan maka proses memahami objektif sistem serta kelemahan sistem terhadap penyerang sistem harus dilakukan (Maurer, 2004). Serangan terhadap keamanan sistem terjadi karena adanya kelemahan (*vulnerabilities*) yang dimanfaatkan baik secara sengaja maupun tidak oleh pihak lain (Wiksten, 2003). Sejalan dengan peningkatan teknologi di bidang informasi digital juga meningkatkan teknologi di bidang komunikasi serta penyadapan atau *eavesdropping* (Tanveer, 2017)

Kriptografi merupakan metode yang menyediakan autentikasi (*authentication*), integritas (*integrity*), serta ketersediaan (*availability*) dari data yang diamankan (Tripathi dan Agrawal, 2014). Enkripsi merupakan teknik mengubah informasi berupa data teks (*plain text*) menjadi teks yang tidak memiliki informasi apa-apa yang berupa teks sandi (*cipher text*) sementara dekripsi merupakan proses kebalikan dari enkripsi untuk mendapatkan kembali data asli dari teks sandi yang diperoleh (Tanveer, 2017)

Metode yang digunakan dalam penelitian ini adalah kriptografi dengan jaringan saraf tiruan atau yang lebih dikenal dengan *neural cryptography*. Metode *neural cryptography* pertama kali diperkenalkan oleh fisikawan Kanter dan Kinzel pada tahun 2002 dengan memperkenalkan aturan penukaran antara dua pihak

dengan menggunakan jaringan saraf tiruan untuk dapat bersinkronisasi dengan cara saling belajar (*mutual learning*). Penelitian yang dilakukan oleh Ismail et al (2012) dengan judul *Satelite Image Encryption Using Neural Network Backpropagation* mengungkapkan bahwa metode ini memungkinkan terjadinya enkripsi dengan menggunakan kunci yang berubah-ubah tergantung struktur dari jaringan yang dibentuk.

Jaringan saraf tiruan merupakan teknik yang handal karena memiliki kemampuan meniru serta belajar terhadap perhitungan yang sangat kompleks (Komal et. al, 2015). Proses enkripsi data melalui jaringan saraf tiruan harus melewati proses pelatihan terhadap bobot yang diberikan untuk menghasilkan bobot yang diinginkan (Sari et. al, 2008). Penggunaan jaringan saraf tiruan pada kriptografi dapat mengurangi kebutuhan terhadap penggunaan kunci serta penerapan algoritma untuk keamanan terhadap kunci yang digunakan namun tetap menghasilkan keamanan tingkat tinggi (Mahesha, 2016)

Berdasarkan penjelasan singkat diatas, pada penelitian ini akan membahas mengenai penerapan jaringan saraf *backpropagation* pada *symmetric cryptography* terhadap data teks.

1.3 Rumusan Masalah

Salah satu kemampuan dari jaringan saraf tiruan adalah kemampuan sinkronisasi dengan menggunakan neuron pada jaringannya yang diyakini dapat digunakan membantu perhitungan pada kriptografi (Ismail et. al, 2012). Salah satu

teknik penyerangan yang sering dilakukan adalah *brute force* yang merupakan teknik penyerangan terhadap data secara acak dan terus menerus melacak algoritma atau kunci yang digunakan untuk menyembunyikan data sesungguhnya. Model pada jaringan saraf tiruan diyakini dapat membantu menemukan metode lain dalam melakukan proses enkripsi dan dekripsi data karena jaringan saraf tiruan memungkinkan *neuron* yang terdapat pada jaringan untuk menghasilkan *cipher text* yang bersifat dinamik (Al-nima et.al., 2009)

Salah satu jenis dari jaringan saraf tiruan yang sering digunakan saat ini dan dapat diterapkan kriptografi adalah metode *backpropagation*. Diantara jenis-jenis jaringan saraf tiruan yang memiliki banyak *layer* atau *multi-layer*, *backpropagation* menunjukkan performa terbaik dalam hal proses gabungan sekuensial (*composite sequential process*) yang dikarenakan oleh kemampuan *backpropagation* untuk meminimalisir kegagalan ekpektasi terhadap hasil akhir pada akhir proses (Alsmadi, Mutasem, 2009)

Untuk mengatasi permasalahan di atas disusun beberapa pertanyaan penelitian sebagai berikut:

1. Bagaimana proses penerapan jaringan saraf tiruan pada bidang kriptografi?
2. Bagaimana cara menyesuaikan data masukan teks pada jaringan yang dibentuk untuk kebutuhan enkripsi dan dekripsi?

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mencari alternatif untuk mengembangkan proses enkripsi pada data teks dengan menggunakan bantuan dari kecerdasan buatan yang dalam hal ini adalah jaringan saraf tiruan *backpropagation*

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah hasil dari penelitian dapat diterapkan untuk melakukan teknik enkripsi data teks yang menghasilkan *cipher text* yang dinamik serta dapat dikembangkan sebagai landasan untuk menambah variasi metode enkripsi dan dekripsi pada macam-macam tipe data dengan menggunakan bantuan dari kecerdasan buatan.

1.6 Batasan Penelitian

Batasan yang diterapkan dalam penelitian ini antara lain adalah:

1. Pada penelitian ini hanya akan membahas proses enkripsi dan dekripsi dari data teks menggunakan metode jaringan saraf tiruan *backpropagation*. Nantinya penelitian ini akan berfokus pada proses enkripsi dan dekripsi serta menganalisa hasil yang didapat dari proses penggabungan jaringan saraf tiruan dan kriptografi.
2. Data yang digunakan sebagai masukan pada sistem yang dibuat merupakan potongan kata atau kalimat dalam huruf latin yang terdapat dalam aturan ASCII dengan tipe data *String* berupa karakter alfabetik.

3. Hasil dari penelitian ini berupa analisa performa dari penerapan algoritma *backpropagation* pada proses enkripsi dan dekripsi.

1.7 Objektif penelitian

Objektif pada penelitian ini adalah antara lain:

- i) Mendesain kriptografi dengan menggunakan algoritma *backpropagation* pada data teks
- ii) Menganalisa performa dari *backpropagation* pada kriptografi.

1.8 Sistematika penulisan

Sistematika penulisan yang terdapat dalam penelitian ini antara lain:

BAB I. PENDAHULUAN

Pada bab ini akan membahas mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, objektif penelitian, serta sistematika penulisan yang terdapat dalam penelitian ini.

BAB II. KAJIAN LITERATUR

Pada bab ini akan membahas mengenai landasan gagasan atau dasar-dasar pembahasan dalam melakukan penelitian ini. Literatur yang digunakan didapat dari berbagai penelitia ilmiah

yang sudah pernah dilakukan sebelumnya yang akan digunakan sebagai referensi dalam mengembangkan penelitian ini seperti penerapan jaringan saraf tiruan untuk membangun sistem kriptografi.

BAB III. METODOLOGI PENELITIAN

Pada bab ini akan membahas tentang alur yang dilakukan dalam melakukan penelitian atau tahapan yang menjadi dasar serta panduan yang akan dilakukan dalam mengembangkan penelitian ini. Tahapan yang dijelaskan dalam bab ini akan dijelaskan dalam bentuk kerangka kerja (*framework*). Bab ini juga akan menyertakan perancangan manajemen proyek yang dirancang untuk melaksanakan penelitian.

BAB IV. PENGEMBANGAN PERANGKAT LUNAK

Pada bab ini akan dibahas mengenai implementasi terhadap metode yang diusulkan sesuai dengan objektif yang sudah dibangun sebelumnya yang berdasarkan metode pembangunan perangkat lunak *Rational Unified Process* (RUP) dengan disertai penjelasan di setiap fasenya seperti fase inepsi, elaborasi, kontruksi, serta transisi.

BAB V. HASIL DAN ANALISIS PENELITIAN

Pada bab ini akan membahas mengenai Analisa terhadap hasil yang didapat dari pembangunan sistem kriptografi menggunakan jaringan saraf tiruan dengan metode *backpropagation* pada data teks serta performa yang dihasilkan.

BAB VI. KESIMPULAN DAN SARAN

Pada bab ini akan membahas tentang rangkuman seluruh penelitian yang telah dilakukan berupa kesimpulan yang didapat dari penelitian. Pada bab ini juga akan memberikan saran yang dapat dilakukan untuk mengembangkan penelitian selanjutnya di masa yang akan mendatang.

1.9 Kesimpulan

Pada pembahasan ini menjelaskan mengenai alasan mengapa penelitian berupa penerapan jaringan saraf tiruan *backpropagation* pada kriptografi simetris pada data gambar dilakukan disertai dengan objektif, tujuan, serta batasan yang telah ditetapkan.

REFERENCES

- Abdullah, A. M. (2017) 'Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data', (June).
- Adhie, R. P., Hutama, Y. and Ahmar, A. S. (2018) 'Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC) Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC)'.
- Al-nima, R. R., Muhanad, L. and Hassan, S. Q. (2009) 'Data encryption Using Backpropagation Neural Network'.
- Arief, Ibrahim. (no date) 'Perancangan Algoritma Kriptografi Kunci Simetri dengan Menggunakan Jaringan Saraf Tiruan'
- Aris, A. (2005) 'Aplikasi Pengolahan Citra dan Jaringan Syaraf Tiruan Untuk Pengenalan Pola Tulisan Tangan', pp. 1–4.
- Atmaja, R. D. *et al.* (2016) 'An Image Processing Method to Convert RGB Image into Binary', 3(2), pp. 377–382. doi: 10.11591/ijeecs.v3.i2.pp377-382.
- Bora, D. J. and Gupta, D. A. K. (2014) 'A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm', *International Journal of Computer Trends and Technology*, 10(2), pp. 108–113. doi: 10.14445/22312803/IJCTT-V10P119.
- Chandra, S. (2014) 'A Study and Analysis on Symmetric Cryptography'.
- Cryptosystems, A. (no date) 'Advantages and Disadvantages of Asymmetric and Symmetric Cryptosystems'.
- Devi, S. (2015) 'Encoding N-party Man-In-Middle Attack for Diffie – Hellman Algorithm in a Client-Server Paradigm', 6(6), pp. 4860–4865.
- Garg, N. (2014) 'Comparison of Asymmetric Algorithms in Cryptography', 3(4), pp. 1190–1196.
- Gurney, K. (no date) *An introduction to neural networks An introduction to neural networks*.

- Ismail, I. A. *et al.* (2012) ‘Satellite Image Encryption Using Neural Networks Backpropagation’, (October), pp. 13–15.
- Khaldi, A. (2018) ‘Diffie-Hellman Key Exchange through Steganographed Images’, (May).
- Kherad, F. J. *et al.* (2010) ‘A new symmetric cryptography algorithm to secure e-commerce transactions’, (June). doi: 10.1109/ICFTE.2010.5499388.
- Komal, T. *et al.* (2015) ‘Encryption and Decryption using Artificial Neural Network’, 2(4), pp. 2393–2395. doi: 10.17148/IARJSET.2015.2419.
- Lecturer, S. (2016) ‘Authentication based two level Encryption & Decryption of an Image using Artificial Neural Network . Artificial Neural’, (9).
- Matuki, T. *et al.* (2007) ‘Three dimensional medical images of the lungs and brain recognized by artificial neural networks’, pp. 1117–1121.
- Neidhardt, E. (no date) ‘Asymmetric Cryptography for Mobile Devices’.
- Pavithra, S. (2012) ‘Throughput Analysis of Symmetric Algorithms’, 1577(02), pp. 1574–1577.
- Rezaeipour, D. (2010) ‘The block cipher algorithm-properties , encryption efficiency analysis and security evaluation ENCRYPTION EFFICIENCY ANALYSIS AND SECURITY Institute for Mathematical Research’, (January).
- Security, C. (2004) ‘Symmetric Key cryptosystem’, pp. 1–19.
- Sengupta, A. and Pal, T. K. (2009) *Fuzzy Preference Ordering of Interval Numbers in Decision Problems*.
- Sharif, S. O. and Mansoor, S. P. (2010) ‘Performance analysis of Stream and Block cipher algorithms’, pp. 522–525.
- Tarigan, J., Diedan, R. and Suryana, Y. (2017) ‘ScienceDirect ScienceDirect ScienceDirect Plate Recognition Using Backpropagation Neural Network and Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm Genetic Algorithm’, *Procedia Computer Science*. Elsevier B.V., 116, pp. 365–372. doi: 10.1016/j.procs.2017.10.068.
- ‘Theory and Implementation of Data Encryption Standard : A Review’ (2014), (May). doi: 10.3923/itj.2005.307.325.

- Tripathi, R. and Agrawal, S. (2014) 'Comparative Study of Symmetric and Asymmetric Cryptography Techniques', 1(6), pp. 68–76.
- Volna, E., Kotyrba, M. and Janosek, M. (2012) 'CRYPTOGRAPHY BASED ON NEURAL NETWORK', (February 2015). doi: 10.7148/2012-0386-0391.
- Young, I. T., Gerbrands, J. J. and Vliet, L. J. Van (2007) 'Fundamentals of Image Processing', pp. 1–112.

```
package
fullyconnectednetw
ork;
```

```
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.Scanner;

public class Main {
    private final double[][]
output;

    public final double[][][]
weights;

    private final double[][]
bias;

    private final double[][]
errorValue;

    private final double[][]
derivative;

    private final double[][]
array;

    public final int[] neurons;

    public final int inputSize;

    public final int outputSize;

    public final int networkSize;

    public static String key;

    public double[] weightawal =
DataProcessing.per26(DataProcessi
ng.stringToDouble(key.length(),
key));

    public NeuralNetwork(int...
neurons){
        this.neurons = neurons;

        this.inputSize =
neurons[0];
```

```
        this.networkSize =
neurons.length;

        this.outputSize =
neurons[networkSize-1];

        this.output = new double
[networkSize][];

        this.weights = new double
[networkSize][][];

        this.bias = new double
[networkSize][];

        this.errorValue = new
double [networkSize][];

        this.derivative = new
double [networkSize][];

        this.array = new
double[networkSize][];

        for(int i=0;
i<networkSize; i++){
            this.output[i] = new
double [neurons[i]];

            this.errorValue[i] =
new double [neurons[i]];

            this.derivative[i] =
new double [neurons[i]];

            this.bias[i] =
Weights.createRandomArray(neurons
[i], 0.0, 1.0);

            if(i > 0){

                System.out.println(neurons[i]
+" "+ neurons[i-1]);

                weights[i] =
Weights.createWeight(neurons[i],
neurons[i-1], weightawal);
            }
        }
    }
}
```



```

        public double[]
        calculate(double[] input
        /*,JTextArea epocTextArea*/) {

            if(input.length !=
            this.inputSize) return null;

            this.output[0] = input;

            for(int layer = 1;
            layer<networkSize; layer++){

                if(layer==1){

                    System.out.print("Output pada
                    Hidden layer\t");

                }

                else {

                    System.out.print("Output
                    pada Output layer\t");

                }

                for(int neuron = 0;
                neuron < neurons[layer];
                neuron++){

                    double sum =
                    bias[layer][neuron];

                    for(int
                    prevNeuron = 0; prevNeuron <
                    neurons[layer-1]; prevNeuron++){

                        sum+=

                        output[layer-
                        1][prevNeuron]*weights[layer][neu
                        ron][prevNeuron];

                    }

                    output[layer][neuron] =
                    sigmoid(sum);

                    derivative[layer][neuron] =
                    output[layer][neuron] * (1 -
                    output[layer][neuron]);

                    System.out.print(output[layer][ne
                    uron]+\t");

                }

            }

            System.out.println("");

            }

            return
            output[networkSize-1];

        }

        public double[]
        encrypt(double...
        processedTarget){

            double[] arrHidden = new
            double[processedTarget.length];

            for(int j = 0;
            j<processedTarget.length; j++){

                double a
                =output[1][j]; //return
                output[1][j];

                arrHidden[j] = a;

            }

            return arrHidden;

        }

        public double[]
        decrypt(double...
        processedTarget){

            double[] arrOutput = new
            double[processedTarget.length];

            for(int j = 0;
            j<processedTarget.length; j++){

                double b
                =output[2][j];

                arrOutput[j] = b;

            }

            return arrOutput;

        }

        public double[] run(double...
        processedTarget){

            this.output[0] =
            processedTarget;

            System.out.print("Output
            pada Hidden layer\t");

            for(int j = 0;
            j<processedTarget.length; j++){

```

```

        double a
=output[1][j];

System.out.print("\t"+a);

    }

    System.out.println();

    System.out.print("Output
pada Output layer\t");

        for(int j = 0;
j<processedTarget.length; j++){

            double b
=output[2][j];

System.out.print("\t"+b);

        }

        return null;
    }

    public void train (double[]
input, double[] target, double
eta /*, JTextArea
epocJTextArea*/){

        if(input.length !=
inputSize ||target.length !=
outputSize)

            return;

        calculate(input);
        backpropError(target);
        updateWeights(eta);
    }

    public void
backpropError(double[] target){

        for(int neuron = 0;
neuron < neurons[networkSize-1];
neuron++ ){

errorValue[networkSize-1][neuron]
= (output[networkSize-1][neuron]-
target[neuron])

            *derivative[networkSize-
1][neuron];

        }

        for(int layer =
networkSize-2; layer > 0; layer--
){

            for(int neuron = 0;
neuron < neurons[layer];
neuron++){

                double sum = 0;

                for(int
nextNeuron = 0; nextNeuron <
neurons[layer+1]; nextNeuron++){

                    sum
+=weights[layer+1][nextNeuron][ne
uron] *
errorValue[layer+1][nextNeuron];

                }

                this.errorValue[layer][neuron]=su
m*derivative[layer][neuron];

            }

        }

    }

    public void
updateWeights(double eta){

        for(int layer = 1; layer
<networkSize; layer++){

            for(int neuron = 0;
neuron < neurons[layer];
neuron++){

                for(int
prevNeuron = 0; prevNeuron <
neurons[layer-1]; prevNeuron++){

                    double delta
= - eta * output[layer-
1][prevNeuron] *
errorValue[layer][neuron];

                    weights[layer][neuron][prevNeuron
] += delta;

                }

            }

        }

    }

```



```

        double delta = -
eta * errorValue[layer][neuron];

bias[layer][neuron] += delta;
    }
}

```

```

private double sigmoid
(double x){

    return 1d / (1 +
Math.exp(-x)); //fungsi sigmoid
yang digunakan
}

```

```

private String
numberToChar2(double[] input) {

    String hasil = " ";

    for(int j = 0;
j<inputSize; j++){

        int l = (int)
Math.ceil(input[j]);

        hasil = hasil +
String.valueOf((char)(l + 'A' -
1));

    }

    return hasil;

}

```

```

public String
numberToChar(double[] input) {

    String hasil = " ";

    for(int j = 0;
j<inputSize; j++){

        int l = (int)
Math.ceil(input[j]*26);

```

```

        hasil = hasil +
String.valueOf((char)(l + 'A' -
1));

    }

    return hasil;

}

```

```

public double nilaiAkurasi
(String input, String plainText){

```

```

System.out.println("\nFUNGSI
NILAI AKURASI");

```

```

    double jmlKarakterSama =
0.0;

```

```

    for(int i=0;
i<input.length(); i++)

```

```

    {

        if(input.length()==plainText.leng
th()){

```

```

System.out.println("index ke-
"+i+" adalah "+input.charAt(i)+"
dan "+plainText.charAt(i));

```

```

if(input.charAt(i)==plainText.cha
rAt(i))

```

```

        {

            jmlKarakterSama =
jmlKarakterSama+(1.0);

        }

```

```

    }

    else

    {

```

```

System.out.println("input :
"+input+" dengan panjang
"+input.length());

```

```

System.out.println("plaintext :
"+plainText+" dengan panjang
"+plainText.length());

```

```

    }

}

```

```

        double nilaiAkurasi =
(jmlKarakterSama /
input.length())*100;

System.out.println("jumlah
karakter sama adalah
"+jmlKarakterSama);

        System.out.println("nilai
akurasi adalah "+nilaiAkurasi);

        System.out.println("total
karakter adalah
"+input.length());

        return nilaiAkurasi;
    }

    public double
nilaiPerubahan(String input,
String cipherText){

System.out.println("\nFUNGSI
NILAI PERUBAHAN");

        double jmlKarakterBeda =
0.0;

        for(int i=0;
i<input.length(); i++)
        {

if(input.length()==cipherText.len
gth()){

System.out.println("index ke-
"+i+" adalah "+input.charAt(i)+"
dan "+cipherText.charAt(i));

if(input.charAt(i)!=
cipherText.charAt(i))
        {

jmlKarakterBeda =
jmlKarakterBeda+(1.0);

        }

        }

        }

        double nilaiPerubahan =
(jmlKarakterBeda /
input.length())*100;

System.out.println("jumlah
karakter sama adalah
"+jmlKarakterSama);

        System.out.println("nilai
perubahan adalah
"+nilaiPerubahan);

        System.out.println("total
karakter adalah
"+input.length());

        return nilaiPerubahan;
    }

    public static void
main(String[] args) throws
IOException {

        Training tf = new
Training();

        tf.setVisible(true);
    }
}

```


weights.java

```

package fullyconnectednetwork;

public class Weights {

    public static double[]
createArray (int size, double
init_value){
        if(size < 1){
            return null;
        }
        double[] ar = new
double[size];
        for(int i=0; i<size;
i++){
            ar[i] = init_value;
        }
        return ar;
    }

    public static double[]
createRandomArray(int size,
double minValue, double
maxValue){
        if(size<1){
            return null;
        }
        double[] ar = new
double[size];
        for(int i=0; i<size;
i++){
            ar[i] =
randomValue(minValue, maxValue);
        }
        return ar;
    }

    public static double[][]
createWeight(int sizeX, int
sizeY, double[]w0){
        if(sizeX < 1 || sizeY <
1){
            return null;
        }
        int k=0;
        double[][] ar = new
double[sizeX][sizeY];
        for(int i=0; i<sizeX;
i++){
            for(int j=0; j<sizeY;
j++){
                if(k>=w0.length)
                    k=0;
                ar[i][j] = w0[k];
                k++;
            }
        }
        return ar;
    }

    public static double
randomValue(double minValue,
double maxValue){
        return
Math.random()*(maxValue-
minValue)+minValue;
    }

    public static Integer[]
randomValues(int minValue, int
maxValue, int amount){
        minValue --;

        if(amount > (maxValue-
minValue)){
            return null;
        }

```

```

        Integer[] values = new
Integer[amount];
        for(int i = 0; i<amount;
i++){
            int n =
(int)(Math.random() * (maxValue-
minValue-1)+ minValue);

while(containsValue(values, n)){
            n =
(int)(Math.random() * (maxValue-
minValue-1)+ minValue);
            }
            values[i] = n;
        }
        return values;
    }
}

```

```

    public static <T extends
Comparable<T>> boolean
containsValue(T[] ar, T value){
        for(int i = 0 ; i <
ar.length ; i++){
            if(ar[i] != null){

if(value.compareTo(ar[i]) == 0){
                return true;
            }
        }
        return false;
    }
}

```

```

    public static int
indexOfHighestValue(double[]
values){
        int index=0;

        for(int i = 1 ; i <
values.length ; i++){
            if(values[i] >
values[index]){
                index = i;
            }
        }
    }
}

```


Initialization.jav

a

```

package fullyconnectednetwork;

import java.util.Scanner;

public class Initialization {
    private String input;
    private String key;
    private String weight;
    private String cipherText;
    private String plainText;
    private String
ActivationFunction;
    private double[][]
epochValue;
    private double nilaiAkurasi;
    private double
nilaiPerubahan;

    public void setNilaiAkurasi
(double nilaiAkurasi){
        this.nilaiAkurasi =
nilaiAkurasi;
    }

    public void setNilaiPerubahan
(double nilaiPerubahan){
        this.nilaiPerubahan =
nilaiPerubahan;
    }

    public void
setEpochValue(double
[][]epochValue){
        this.epochValue =
epochValue;
    }

    public void setAF(String
ActivationFunction){
        this.ActivationFunction =
ActivationFunction;
    }

    public void
setCipherText(String cipherText){
        this.cipherText =
cipherText;
    }

    public void
setPlainText(String plainText){
        this.plainText =
plainText;
    }

    public void setWeight(String
weight){
        this.weight = weight;
    }

    public void setInput(){
        Scanner scan= new
Scanner(System.in);

        System.out.println("Please type
the text as the input below");
        //System.out.println();
        input = scan.nextLine();
        this.input = input;
    }

    public void setKey(){
        Scanner scan= new
Scanner(System.in);

        System.out.println("Please type
the word as the key below");

```

```
        //System.out.println();
        key = scan.nextLine();
        this.key = key;
    }

    public String
getCipherText(){
        return this.cipherText;
    }

    public String getPlainText(){
        return this.plainText;
    }

    public String getWeight(){
        return this.weight;
    }

    public String getInput(){
        return this.input;
    }

    public String getKey(){
        return this.key;
    }

    public String getAF(){
        return
this.ActivationFunction;
    }

    public double[][]
getEpochValue(){
        return this.epochValue;
    }

        public double
getNilaiAkurasi(){
            return this.nilaiAkurasi;
        }

        public double
getNilaiPerubahan(){
            return
this.nilaiPerubahan;
        }
    }
```


DataProcessing.java

a

```

package fullyconnectednetwork;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.Scanner;

public class DataProcessing {

    public static double[]
    stringToDouble (int size, String
    word) {

        if(size < 1){
            return null;
        }

        double[] inputData = new
        double[size];

        for(int i=0;
        i<word.length();i++){

            int ascii =
            word.toCharArray()[i];

            ascii = (ascii - 96);

            inputData[i] = ascii;
        }

        return inputData;
    }

    public String[]
    splitStringEvery(String s, int
    interval) {

```

```

        int arrayLength = (int)
        Math.ceil(((s.length() /
        (double)interval)));

        String[] result = new
        String[arrayLength];

        int j = 0;

        int lastIndex =
        result.length - 1;

        for (int i = 0; i <
        lastIndex; i++) {

            result[i] =
            s.substring(j, j + interval);

            j += interval;

        } //Add the last bit

        result[lastIndex] =
        s.substring(j);

        return result;
    }

    public static String getInput
    (){

        Scanner scan= new
        Scanner(System.in);

        System.out.println("Please type
        the text as the input below");

        //System.out.println();

        String
        str=scan.nextLine();

        return str;
    }

    public static String getKey
    (){

        Scanner scan= new
        Scanner(System.in);

        System.out.println("Please type
        the word as the key below");

```

```

        //System.out.println();
        String
str=scan.nextLine();

        return str;
    }

    public static double[]
per26(double[] input){

        double[] arr = new
double[input.length];

        for(int i=0;
i<input.length; i++){
            arr[i] = (input[i] /
26.0);

        }

        return arr;
    }

    public static String deleteZ
(String data){

        data =
data.replaceAll("z", "");

        return data;
    }

    public static String addZ
(String data){

        data =
data.replaceAll("\\s", "");

while(data.length()%5!=0){

    addZ = "z";

    data =
data + addZ;

}

```

Details.java

```

package fullyconnectednetwork;

public class Details {
    public double nilaiAkurasi
    (String input, String plainText){

        System.out.println("\nFUNGSI
        NILAI AKURASI");

        String messages =
        "failed";

        double jmlKarakterSama =
        0.0;

        for(int i=0;
        i<input.length(); i++)
        {

            if(input.length()==plainText.leng
            th()){

                System.out.println("index ke-
                "+i+" adalah "+input.charAt(i)+"
                dan "+plainText.charAt(i));

                if(input.charAt(i)==plainText.cha
                rAt(i))

                    {

                        jmlKarakterSama =
                        jmlKarakterSama+(1.0);

                    }

                //else return 111.111;

                else
                {

                    System.out.println("input :
                    "+input+" dengan panjang
                    "+input.length());

                    System.out.println("plaintext :

                    "+plainText+" dengan panjang
                    "+plainText.length());

                }

            }

            System.out.println("jumlah
            karakter sama adalah
            "+jmlKarakterSama);

            System.out.println("nilai
            akurasi adalah "+nilaiAkurasi);

            System.out.println("total
            karakter adalah
            "+input.length());

            return nilaiAkurasi;

        }

        public double
        nilaiPerubahan(String input,
        String cipherText){

            System.out.println("\nFUNGSI
            NILAI PERUBAHAN");

            double jmlKarakterBeda =
            0.0;

            for(int i=0;
            i<input.length(); i++)
            {

                if(input.length()==cipherText.len
                gth()){

                    System.out.println("index ke-
                    "+i+" adalah "+input.charAt(i)+"
                    dan "+cipherText.charAt(i));

                    if(input.charAt(i)!=
                    cipherText.charAt(i))
                }

            }

        }

    }
}

```



```
        {

jmlKarakterBeda =
jmlKarakterBeda+(1.0);

        }

    }

    //else return 111.111;

    else

    {

System.out.println("input :
"+input+" dengan panjang
"+input.length());

System.out.println("plaintext :
"+cipherText+" dengan panjang
"+cipherText.length());

        }

    }

        double nilaiPerubahan =
(jmlKarakterBeda /
input.length())*100;

System.out.println("jumlah
karakter beda adalah
"+jmlKarakterBeda);

        System.out.println("nilai
perubahan adalah
"+nilaiPerubahan);

        System.out.println("total
karakter adalah
"+input.length());

        return nilaiPerubahan;

    }

}
```

Training.java

```

private void
browseButtonActionPerformed(java.
awt.event.ActionEvent evt) {

    FileReader FR = null;

    try {

        JFileChooser FC = new
JFileChooser();

        int returnVal =
FC.showOpenDialog(browseButton);

        String path =
FC.getSelectedFile().getAbsoluteP
ath();

        String line = null;

        StringBuffer isitext
= new StringBuffer();

        FR = new
FileReader(path);

        BufferedReader BR;

if(path.endsWith(".txt")){

        BR = new
BufferedReader(FR);

        try {

            while((line =
BR.readLine()) != null) {

isitext.append(line+"\n");

            }

inputArea.setText(isitext.substri
ng(0, isitext.length()-1));

        } catch
(IOException ex) {

Logger.getLogger(Training.class.g
etName()).log(Level.SEVERE, null,
ex);

        }

    }

    else{

JOptionPane.showMessageDialog(nul
l, "Invalid Input");

        }

        // TODO add your
handling code here:

    } catch
(FileNotFoundException ex) {

Logger.getLogger(Training.class.g
etName()).log(Level.SEVERE, null,
ex);

        } finally {

            try {

                FR.close();

            } catch (IOException
ex) {

Logger.getLogger(Training.class.g
etName()).log(Level.SEVERE, null,
ex);

            }

        }

    }

private void
TrainButtonActionPerformed(java.a
wt.event.ActionEvent evt) {

DataProcessing.noSpace (NeuralNetw
ork.key =
keyTextField.getText());

        String cipherText = "";

        String plainText = "";

        int epoch =
Integer.parseInt(epochTextField.g
etText());

        //int AF =
Integer.parseInt(init.getAF());

        //double af = AF;

        String comboBoxValue =
afComboBox.getSelectedItem().toSt
ring();

```

```

        double AF =
        Double.parseDouble(comboBoxValue)
        ;

        //double[] input = new
        double[]{0.61, 0.03, 0.46, 0.34};

        //double[] target = new
        double[]{0.61, 0.03, 0.46, 0.34};

        //String[] data =
        {"INDONESIA"};

        //String sentences =
        "rumah";

        String input =
        inputArea.getText();

        //int size =
        sentences.length();

        NeuralNetwork net = new
        NeuralNetwork(5, 5,
        5); //NETWORK_LAYER_SIZES

        Details det = new
        Details();

        input =
        DataProcessing.addZ(input);
        //membuat input kelipatan 5

        String [] blockInput =
        input.split("(?<=\\G.{5})");

        //System.out.println("Nilai bobot
        pada setiap layer : \n");

        for(int
        i=1;i<3;i++){

            for(int
            j=0;j<net.weights[i].length;j++){

                for(int
                k=0;k<net.weights[i][j].length
                ;k++){

                    if(i==1){

                        weightModel.insertRow(weightModel
                        .getRowCount(), new
                        Object[]{"Hidden
                        Layer",j,k,net.weights[i][j][k]});
                    }

                    else

                    weightModel.insertRow(weightModel
                    .getRowCount(), new
                    Object[]{"Output
                    Layer",j,k,net.weights[i][j][k]});
                }

            }

        }

        //System.out.println("nilai dari
        bobot "+i+", "+j+", "+k+" adalah
        "+net.weights[i][j][k]);
    }

    for(int index=0;
    index<(input.length()/5);
    index++){

        String word =
        blockInput[index];

        double[] subInput =
        DataProcessing.stringToDouble(wor
        d.length(), word);

        //System.out.println("input yang
        ke-"+(index+1)+" yaitu "+word+"
        dengan nilai
        "+Arrays.toString(subInput));

        //inputTextArea.setText("input
        yang ke-"+(index+1)+" yaitu
        "+word+" dengan nilai
        "+Arrays.toString(subInput));

        //-----
        TABEL-----
        --

        inputModel.insertRow(inputModel.g
        etRowCount(), new
        Object[]{index+1,word,Arrays.toSt
        ring(subInput)});
    }

```

```

        double[]
processedInput =
DataProcessing.per26(subInput);

        double[]
processedTarget = processedInput;

        for(int i = 0;
i<epoch; i++)
        {

System.out.println();

System.out.println("Epoch ke -
"+(i+1));

net.train(processedInput,
processedTarget, AF/*,
epochTextArea*/);

}

        }

        progressNumber.setText("100 %");

        String enkripsi =
net.numberToChar(net.encrypt(proc
essedTarget));

        cipherText =
cipherText+enkripsi;

        String dekripsi =
net.numberToChar(net.decrypt(proc
essedTarget));

        plainText =
plainText+dekripsi;

init.setNilaiAkurasi(det.nilaiAku
rasi(input.replaceAll("\\s", ""),
plainText.toLowerCase().replaceAl
l("\\s", ""));

init.setNilaiPerubahan(det.nilaiP
erubahan(input.replaceAll("\\s",
""),

```

```

cipherText.toLowerCase().replaceA
ll("\\s", ""));

```

```

System.out.println("nilai
perubahan pada Training.java
adalah :
"+init.getNilaiPerubahan());

```

```

System.out.println("nilai akurasi
pada Training.java adalah :
"+init.getNilaiAkurasi());

```

```

}

```

```

init.setCipherText(cipherText);

```

```

init.setPlainText(plainText);

```

```

}

```