

**PENERAPAN EXTREME LEARNING MACHINE DAN
MODIFIKASI SIMULATED ANNEALING UNTUK IDENTIFIKASI
PENYAKIT TANAMAN JARAK PAGAR**

TESIS

Untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun oleh:

Triando Hamonangan Saragih
NIM: 166150100111016



PROGRAM STUDI MAGISTER ILMU KOMPUTER

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2018

PENGESAHAN

PENERAPAN EXTREME LEARNING MACHINE DAN MODIFIKASI SIMULATED ANNEALING UNTUK IDENTIFIKASI PENYAKIT TANAMAN JARAK PAGAR

TESIS

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Magister Ilmu Komputer/Informatika

Disusun oleh:

Triando Hamonangan Saragih

NIM: 166150100111016

Thesis ini telah diuji dan dinyatakan lulus pada 25 Juni 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D

NIP. 19720919 199702 1 001

Prof.Dr.Ir. Abdul Latief Abadi, M.S

NIP. 19550821 198002 1 002

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah Tesis ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah Tesis ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (Magister) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 15 Juli 2018

Triando Hamonangan Saragih

NIM: 166150100111016

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Segala puji syukur untuk Tuhan Yang Maha Esa karena atas rahmat dan berkatNya penulis dapat menyelesaikan Tesis yang berjudul **“PENERAPAN EXTREME LEARNING MACHINE DAN MODIFIKASI SIMULATED ANNEALING UNTUK IDENTIFIKASI PENYAKIT TANAMAN JARAK PAGAR”**. Tesis ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Master Komputer di Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar - besarnya kepada semua pihak yang telah memberikan bantuan baik lahir maupun batin selama penulisan tesis ini. Penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada:

1. Tuhan yang Maha Esa karena atas kehendak dan nikmat-Nya laporan tesis ini telah selesai dengan baik.
2. Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku dosen pembimbing I tesis yang telah memberikan waktu, ilmu dan saran untuk menyelesaikan tesis ini.
3. Prof.Dr.Ir. Abdul Latief Abadi, M.S. selaku dosen pembimbing II tesis yang telah memberikan waktu, ilmu dan saran untuk menyelesaikan tesis ini.
4. Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika.
5. Dr. Eng. Fitra A. Bachtiar, S.T, M.Eng. selaku Ketua Program Studi Magister Ilmu Komputer.
6. Orangtua dan saudara yang mendoakan dan telah memberikan bantuan moral maupun material kepada penulis untuk menyelesaikan tesis ini.
7. Keluarga besar Magister Ilmu Komputer dan PMK yang telah memberikan dukungan dan bantuan selama masa kuliah hingga selesainya tesis ini.
8. Orang-orang yang selalu mendoakan dan membantu penulis yang tidak dapat diucapkan satu persatu, terimakasih atas semua dukungannya.

Penulis mengucapkan banyak terimakasih dan memohon maaf apabila dalam penyusunan laporan ini terdapat banyak kekurangan. Penulis mengharapkan adanya saran maupun kritik yang berguna untuk di masa yang akan datang. Semoga tesis ini dapat memberikan manfaat bagi semua pihak.

Malang,

Penulis

triando.simarmata@gmail.com

ABSTRAK

Tanaman Jarak Pagar merupakan tanaman yang memiliki banyak fungsi dan kegunaan untuk keperluan sehari-hari seperti biodiesel dan alat-alat kecantikan, akan tetapi tanaman ini tidak dapat juga terlepas dari penyakit. Sistem pakar dapat diterapkan dalam melakukan identifikasi sehingga dapat membantu baik petani maupun penyuluh untuk melakukan identifikasi penyakit. Metode yang dapat digunakan salah satunya adalah metode *Extreme Learning Machine*. *Extreme Learning Machine* sudah pernah dilakukan dan hasil akurasi yang diberikan masih perlu peningkatan. Optimasi nilai bobot pada *Extreme Learning Machine* dapat meningkatkan nilai akurasi. Optimasi dilakukan menggunakan *Simulated Annealing* dan menggunakan pohon keputusan memberikan hasil yang lebih baik dari sebelumnya, dengan rata-rata akurasi terbaik sebesar 90,955% dan akurasi maksimal sebesar 94,74%.

Kata Kunci: *Jarak Pagar, Extreme Learning Machine, Simulated Annealing, Identifikasi, Pohon Keputusan*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	iv
DAFTAR GAMBAR.....	v
BAB 1 PENDAHULUAN.....	Error! Bookmark not defined.
1.1 Latar Belakang.....	Error! Bookmark not defined.
1.2 Rumusan Masalah.....	Error! Bookmark not defined.
1.3 Tujuan	Error! Bookmark not defined.
1.4 Manfaat.....	Error! Bookmark not defined.
1.5 Batasan Masalah.....	Error! Bookmark not defined.
1.6 Sistematika Pembahasan	Error! Bookmark not defined.
BAB 2 LANDASAN PUSTAKA	Error! Bookmark not defined.
2.1 Penyakit Tanaman Jarak Pagar	Error! Bookmark not defined.
2.2 Penelitian Sebelumnya	Error! Bookmark not defined.
2.2.1 Fuzzy Neural Network (FNN).....	Error! Bookmark not defined.
2.2.2 Simulated Annealing-Fuzzy Neural Network.....	Error! Bookmark not defined.
2.2.3 Multi Layer Perceptron.....	Error! Bookmark not defined.
2.2.4 Extreme Learning Machine	Error! Bookmark not defined.
2.3 Extreme Learning Machine	Error! Bookmark not defined.
2.3.1 Arsitektur Neural Network	Error! Bookmark not defined.
2.3.2 Model Neural Network.....	Error! Bookmark not defined.
2.3.3 Proses Belajar Neural Network	Error! Bookmark not defined.
2.3.4 Fungsi Aktivasi.....	Error! Bookmark not defined.
2.4 Simulated Annealing	Error! Bookmark not defined.
2.5 Perumusan Hasil Pengujian	Error! Bookmark not defined.
BAB 3 METODOLOGI	Error! Bookmark not defined.



3.1 Metode Penelitian	Error! Bookmark not defined.
3.2 Studi Pustaka.....	Error! Bookmark not defined.
3.3 Pengumpulan Data	Error! Bookmark not defined.
3.4 Pengolahan Data.....	Error! Bookmark not defined.
3.5 Pemodelan Metode	Error! Bookmark not defined.
3.6 Pengujian	Error! Bookmark not defined.
3.7 Metode Perbandingan	Error! Bookmark not defined.
BAB 4 PERANCANGAN.....	Error! Bookmark not defined.
4.1 Deskripsi Perancangan Simulated Annealing-Extreme Learning Machine Error! Bookmark not defined.	
4.2 Contoh Solusi Manual	Error! Bookmark not defined.
4.3 Modifikasi <i>Simulated Annealing-Extreme Learning Machine</i>	Error! Bookmark not defined.
Bookmark not defined.	
4.3.1 Representasi Solusi	Error! Bookmark not defined.
4.3.2 Inisialisasi representasi solusi	Error! Bookmark not defined.
4.3.3 Perhitungan nilai <i>fitness</i>	Error! Bookmark not defined.
4.3.4 Mutasi	Error! Bookmark not defined.
4.3.5 <i>Extreme Learning Machine</i> (ELM) ...	Error! Bookmark not defined.
4.4 Proses Pengujian.....	Error! Bookmark not defined.
BAB 5 PENGUJIAN ALGORITME PEMBANDING.....	Error! Bookmark not defined.
5.1 Pengujian <i>Fuzzy Neural Network</i> (FNN)...	Error! Bookmark not defined.
5.2 Pengujian <i>Simulated Annealing-Fuzzy Neural Network</i>	Error! Bookmark not defined.
5.3 Pengujian <i>Multi Layer Perceptron</i> (MLP).....	Error! Bookmark not defined.
5.4 Pengujian <i>Extreme Learning Machine</i> (ELM).....	Error! Bookmark not defined.
BAB 6 HASIL DAN DISKUSI UNTUK ALGORITME UTAMA	Error! Bookmark not defined.
6.1 Pengujian <i>Extreme Learning Machine</i> dan Modifikasi <i>Simulated Annealing</i>	Error! Bookmark not defined.
6.2 Validasi Hasil Akurasi Dengan Pakar	Error! Bookmark not defined.
BAB 7 PENUTUP	Error! Bookmark not defined.
7.1 Kesimpulan.....	Error! Bookmark not defined.
7.2 Saran	Error! Bookmark not defined.

DAFTAR PUSTAKA.....Error! Bookmark not defined.

LAMPIRAN A HASIL VALIDASI KEPUTUSAN SISTEM DAN DATA KASUSError!

Bookmark not defined.

LAMPIRAN B KODE PROGRAM REPRESENTASI PERMUTASI YANG DIUSULKAN

Error! Bookmark not defined.



DAFTAR TABEL

Tabel 3.1 Tabel Nilai Kepercayaan Penyakit Tanaman Jarak Pagar **Error! Bookmark not defined.**

Tabel 3.2 Contoh Data Latih Penyakit..... **Error! Bookmark not defined.**

Tabel 4.1 Contoh representasi solusi **Error! Bookmark not defined.**

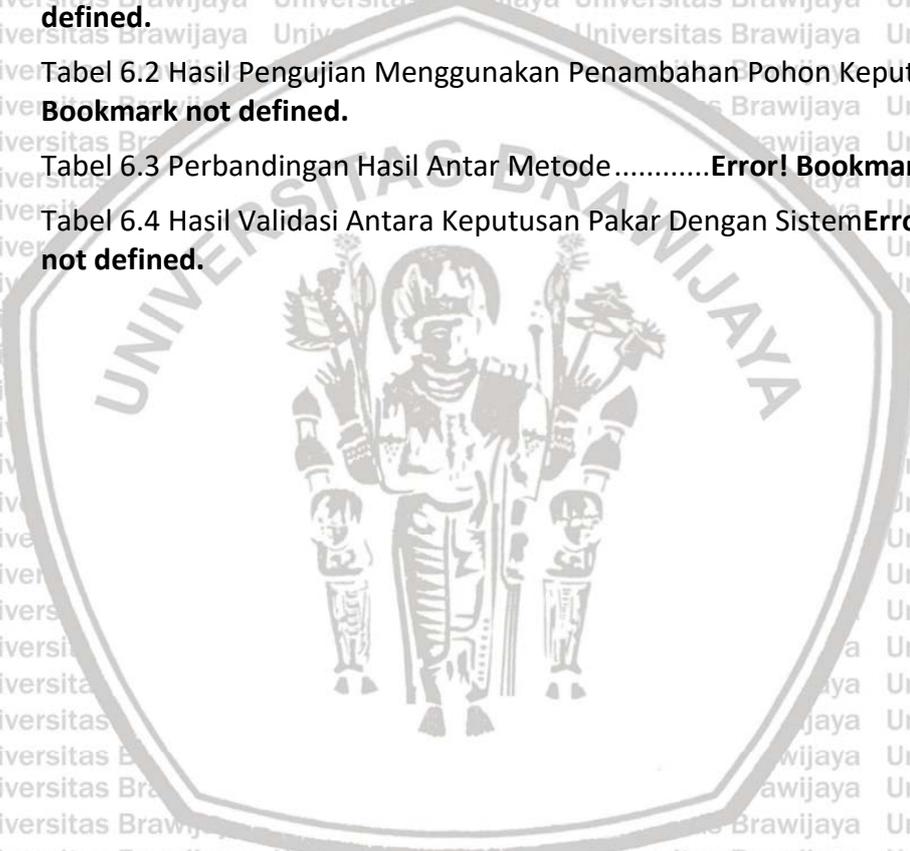
Tabel 4.2 Contoh representasi solusi **Error! Bookmark not defined.**

Tabel 6.1 Tabel Pohon Keputusan Gejala Terhadap Penyakit **Error! Bookmark not defined.**

Tabel 6.2 Hasil Pengujian Menggunakan Penambahan Pohon Keputusan **Error! Bookmark not defined.**

Tabel 6.3 Perbandingan Hasil Antar Metode..... **Error! Bookmark not defined.**

Tabel 6.4 Hasil Validasi Antara Keputusan Pakar Dengan Sistem **Error! Bookmark not defined.**



DAFTAR GAMBAR

- Gambar 2.1 Gejala antraknosa pada jarak pagar.....**Error! Bookmark not defined.**
- Gambar 2.2 Gejala bercak daun cercospora pada daun.....**Error! Bookmark not defined.**
- Gambar 2.3 Gejala penyakit busuk arang.....**Error! Bookmark not defined.**
- Gambar 2.4 Gejala busuk pucuk**Error! Bookmark not defined.**
- Gambar 2.5 Gejala penyakit embun tepung pada tanaman jarak pagar.....**Error! Bookmark not defined.**
- Gambar 2.6 Gejala hawar daun *Xanthomonas* sp. pada daun jarak pagar.....**Error! Bookmark not defined.**
- Gambar 2.7 Gejala layu bakteri pada tanaman jarak pagar.....**Error! Bookmark not defined.**
- Gambar 2.8 Gejala layu *Fusarium* pada tanaman jarak pagar.....**Error! Bookmark not defined.**
- Gambar 2.9 Arsitektur *Extreme Learning Machine* ..**Error! Bookmark not defined.**
- Gambar 2.10 Arsitektur Jaringan *Neural Network*....**Error! Bookmark not defined.**
- Gambar 2.11 Persamaan Fungsi Aktivasi Sigmoid**Error! Bookmark not defined.**
- Gambar 2.12 *Swap Mutation***Error! Bookmark not defined.**
- Gambar 2.13 *Insert Mutation***Error! Bookmark not defined.**
- Gambar 2.14 *Scramble Mutation*.....**Error! Bookmark not defined.**
- Gambar 2.15 *Inversion Mutation*.....**Error! Bookmark not defined.**
- Gambar 3.1 Diagram Tahapan Penelitian.....**Error! Bookmark not defined.**
- Gambar 4.1 Contoh data kasus 1.....**Error! Bookmark not defined.**
- Gambar 4.2 Proses Modifikasi *Simulated Annealing-Extreme Learning Machine***Error! Bookmark not defined.**
- Gambar 4.3 Diagram alir modifikasi *Simulated Annealing-Extreme Learning Machine*.....**Error! Bookmark not defined.**
- Gambar 4.4 Diagram alir inialisasi representasi solusi.....**Error! Bookmark not defined.**
- Gambar 4.5 Diagram alir mutasi.....**Error! Bookmark not defined.**
- Gambar 4.6 Diagram Alir Metode *Extreme Learning Machine*.....**Error! Bookmark not defined.**
- Gambar 5.1 Grafik Pengujian Iterasi.....**Error! Bookmark not defined.**
- Gambar 5.2 Grafik Pengujian *Hidden Layer*.....**Error! Bookmark not defined.**

Gambar 5.3 Grafik Pengujian *Learning Rate*.....**Error! Bookmark not defined.**

Gambar 5.4 Grafik Pengujian Temperatur.....**Error! Bookmark not defined.**

Gambar 5.5 Grafik Pengujian *Cooling Factor*.....**Error! Bookmark not defined.**

Gambar 5.6 Grafik Pengujian Nilai K.....**Error! Bookmark not defined.**

Gambar 5.7 Grafik Pengujian *Epoch***Error! Bookmark not defined.**

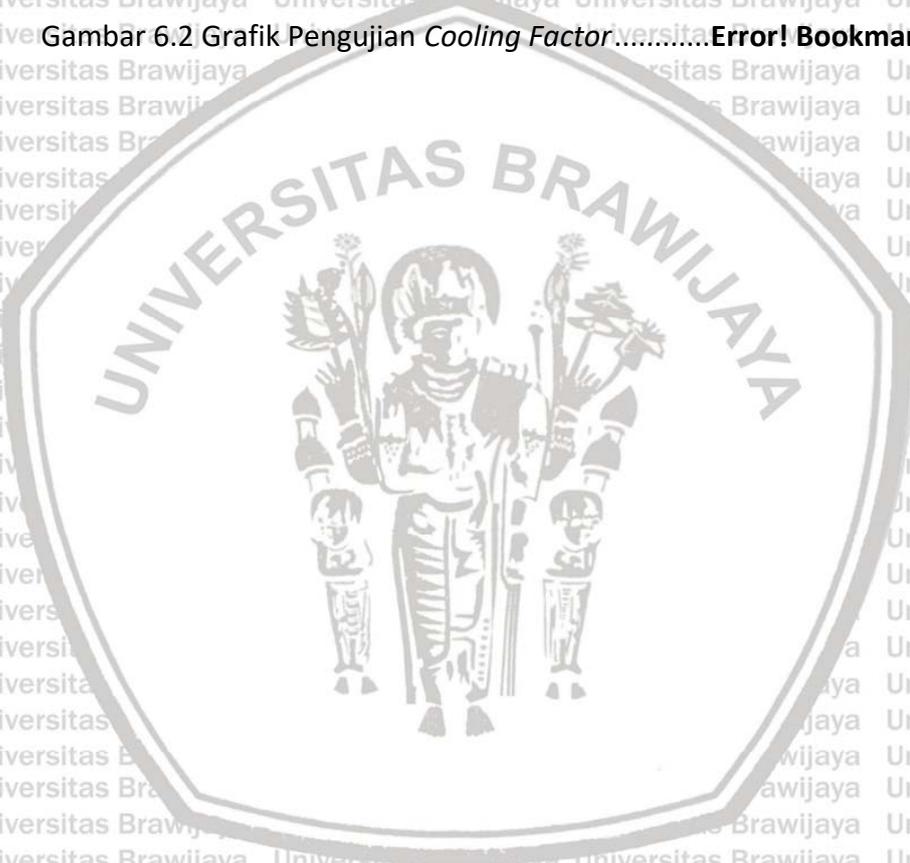
Gambar 5.8 Grafik Pengujian *Hidden Layer*.....**Error! Bookmark not defined.**

Gambar 5.9 Grafik Pengujian *Learning Rate*.....**Error! Bookmark not defined.**

Gambar 5.10 Grafik Pengujian *Neuron***Error! Bookmark not defined.**

Gambar 6.1 Grafik Pengujian Temperatur.....**Error! Bookmark not defined.**

Gambar 6.2 Grafik Pengujian *Cooling Factor*.....**Error! Bookmark not defined.**





BAB 1 PENDAHULUAN

Pada bab pendahuluan akan menjelaskan latar belakang penelitian yang dilakukan. Pada bab ini juga menjelaskan rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

1.1 Latar Belakang

Jarak Pagar adalah tanaman yang dapat bertahan dalam kondisi kering sehingga dapat hidup di dalam kondisi surah hujan rendah. Jarak Pagar biasa ditemukan di daerah Afrika bagian Selatan dan Tengah, India Selatan dan Asia Tenggara (Yulianti & Hidayah, 2015). Jarak Pagar dapat digunakan menjadi pengganti bahan bakar diesel (Fernández, Fiori, Ramos, Pérez, & Rodríguez, 2015). Jarak pagar bisa digunakan juga pada bijinya yang terdapat minyak untuk pembuatan lilin, sabun, kosmetik dan deterjen. Kandungan minyak pada tanaman jarak lebih banyak 63% daripada minyak kedelai, bunga matahari dan minyak sawit.

Terlepas daripada hasil yang diberikan dari tanaman jarak pagar, tanaman ini masih dapat terserang penyakit. Penyakit yang menyerang tanaman ini bisa memberikan dampak pada bagian tanaman seperti daun, batang, akar dan buah. Banyaknya penyakit yang menyerang jarak pagar menjadi masalah karena dapat menurunkan hasil kualitas dari tanaman tersebut (Rodrigues et al., 2015). Masih kurangnya jumlah pakar serta kurangnya pengetahuan petani tentang jarak pagar menjadi pengaruh besar terhadap tanaman jarak pagar (Yulianti & Hidayah, 2015).

Berkembangnya teknologi informasi dan zama sekarang bisa menjadi keuntungan bagi masyarakat. Sistem yang bisa dapat terhubung langsung dengan pengguna membuat pengguna bisa merasakan kenyamanan dalam melakukan sesuatu. Tersedianya sebuah sistem secara online bisa memberikan manfaat bagi masyarakat dalam mempermudah untuk melayani masyarakat secara khusus.

Permasalahan ini bisa diselesaikan dengan memasukkan pengetahuan pakar yang ada ke dalam sebuah sistem cerdas (Masaleno & Hasan, 2012). Sistem cerdas memproses pengetahuan yang telah dimasukkan kemudian mengeluarkan sebuah keputusan berdasarkan algoritme yang digunakan. Berdasarkan hasil dari keputusan itulah dapat dibandingkan dengan keputusan dari seorang pakar seberapa bagus algoritme yang kita pakai (Sutojo, Mulyanto, & Suhartono, 2011).

Neural Network merupakan salah metode komputasi yang bisa digunakan dalam melakukan identifikasi (Lin, Hsieh, & Hu, 2013). *Neural Network* biasa digunakan pada metode klasifikasi dengan dimensi yang begitu tinggi (Zamani, Amaliah, & Munif, 2012). Beberapa metode *Neural Network* yang dapat melakukan identifikasi seperti *Extreme Learning Machine* (Gerlein et al., 2016; Mateo et al., 2013; Yoon et al., 2013), *Fuzzy Neural Network* (Zurada, 2013; Lin, Hsieh, & Hu, 2013) dan *Multi Layer Perceptron* (Zamani, Amaliah, & Munif, 2012). Metode *Extreme Learning Machine* sering dipakai sebagai metode dalam melakukan identifikasi yang dilakukan oleh sistem dan memberikan hasil yang lebih baik pada dimensi yang begitu tinggi (Huang, Zhu, & Siew, 2006; Podolak et al., 2018; Xu & Wang, 2016).

Pemberian nilai bobot pada *Neural Network* masih menggunakan nilai *random*. Nilai *random* bobot dapat memungkinkan kerja *Neural Network* tidak maksimal sehingga memungkinkan *neural network* tidak memberikan hasil identifikasi maksimal. Permasalahan mengenai pemberian nilai bobot pada *Neural Network* dapat diselesaikan menggunakan *Simulated Annealing* (Precup *et al.*, 2013). *Simulated Annealing* bekerja mengoptimasi berdasarkan pencarian tetangga terdekat (Alfarisy *et al.*, 2017; Oliveira *et al.*, 2007; Precup *et al.*, 2013). *Simulated Annealing* memiliki waktu komputasi yang lebih cepat dibanding dengan metode optimasi lainnya (Oliveira *et al.*, 2007).

Berdasarkan penelitian sebelumnya, *Simulated Annealing* digunakan untuk optimasi pada kasus *Deep Learning* dan memberikan nilai *error* yang lebih rendah dibanding dengan tanpa menggunakan optimasi *Simulated Annealing* (Rere, Fanany, & Arymurthy, 2015). *Simulated Annealing* dapat digunakan untuk mengoptimasi nilai bobot pada *Neural Network* (Fajri *et al.*, 2017). Optimasi nilai bobot pada *Neural Network* bertujuan untuk meningkatkan nilai akurasi yang dilakukan oleh *Neural Network*. Pada penelitian yang dilakukan menggunakan *Fuzzy Neural Network* untuk identifikasi penyakit jarak pagar (Saragih *et al.*, 2017) mengalami peningkatan pada saat menggunakan *Simulated Annealing* pada nilai bobot (Fajri *et al.*, 2017). Oleh karena itu, peneliti ingin menggunakan metode *Simulated Annealing* untuk optimasi nilai bobot.

Pohon keputusan merupakan sebuah logika berpikir dimana memberikan sebuah keputusan berdasarkan sebab akibat. Pohon keputusan membantu sebuah sistem untuk memberikan sebuah keputusan berdasarkan syarat-syarat terpenuhi (Jenhani, Amor, & Elouedi, 2008). Pada penelitian sebelumnya untuk penelitian kasus diabetes (Kaewthai, Thammaboosadee, & Kiattisin, 2015) melakukan penambahan pohon keputusan pada *Neural Network* dan memberikan hasil yang baik. Oleh karena itu, akan dilakukan penambahan pohon keputusan pada *Extreme Learning Machine* dan modifikasi *Simulated Annealing*.

Berdasarkan permasalahan yang sudah dijelaskan, peneliti mencoba untuk menggunakan modifikasi *Simulated Annealing-Extreme Learning* pada identifikasi penyakit tanaman jarak pagar. Modifikasi digunakan untuk memberikan solusi optimum yang baik. Penambahan pohon keputusan juga dilakukan agar memberikan keputusan yang lebih akurat agar memberikan hasil akurasi yang lebih baik. Dengan menggunakan algoritme ini dapat memberikan solusi optimum yang lebih baik dari sebelumnya dan memberikan identifikasi yang valid.

1.2 Rumusan Masalah

Berdasarkan dasar permasalahan yang sudah dipaparkan pada latar belakang, dapat diambil rumusan masalah yang ingin diteliti untuk diselesaikan.

1. Bagaimana parameter yang digunakan pada modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar yang menghasilkan akurasi terbaik?
2. Bagaimana pengaruh dari penambahan pohon keputusan pada modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar?

3. Bagaimana tingkat akurasi yang dimiliki oleh modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar?

1.3 Tujuan

Mengacu pada rumusan masalah yang sudah dijabarkan terhadap permasalahan yang sudah dijelaskan, maka diharapkan dapat mencapai beberapa tujuan sebagai berikut.

1. Mengetahui parameter yang digunakan untuk memberikah hasil akurasi terbaik dari modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar.
2. Mengetahui seberapa besar pengaruh penambahan pohon keputusan pada modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar.
3. Mengetahui tingkat akurasi dari modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar.

1.4 Manfaat

Berdasarkan paparan dari masalah yang sudah dijeleskan, maka penelitian ini diharapkan memberikan solusi optimal dengan waktu komputasi yang lebih cepat. Hasil dari penelitian ini diharapkan dapat juga mengurangi tingkat ketidakpastian dan subyektifitas pengetahuan dari pakar sehingga memberikan keputusan yang valid.

1.5 Batasan Masalah

1. Data yang diambil berdasarkan fakta lapangan yang terjadi di perkebunan wilayah Karangploso dan Situbundo Jawa Timur.
2. Data yang diambil berdasarkan bagian dari tanaman yang meliputi buah, daun, batang, dan akar.
3. Data yang digunakan yaitu data penyakit tanaman dan tidak termasuk dengan data yang disebabkan oleh pengaruh fisiologis maupun hama.

1.6 Sistematika Pembahasan

Bagian ini membahas struktur dari sistematika pembahasan yang ingin dijelaskan mulai dari bab pendahuluan hingga penutup. Pada bagian ini diharpkan pembaca dapat mengerti dan memahami sistematika pembahasan isi pada penelitian ini.

1. BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang penulisan tesis ini, rumusan masalah, tujuan, dan manfaat dari tesis, batasan masalah yang dalam penyelesaian tesis serta sistematika pembahasan.

2. BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi tentang dasar teori dan kajian pustaka dalam penelitian modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar, mencakup *Neural Network* dan metode optimasi SA serta membahas penelitian sebelumnya yang menggunakan metode yang berbeda. Pada bab ini juga akan dibahas hasil akurasi sistem pada metode pembandingan yang digunakan.

3. BAB III METODOLOGI

Bab ini membahas langkah dari penelitian untuk deteksi penyakit tanaman jarak pagar.

4. BAB IV PERANCANGAN

Membahas bagaimana arsitektur perancangan yang ingin digunakan berdasarkan proses yang terkait, sistematika perhitungan, persamaan yang digunakan serta arsitektur metode

5. BAB V PENGUJIAN METODE PEMBANDING

Pada bab ini membahas hasil pengujian metode pembandingan yang digunakan pada penelitian. Tujuan dari pembahasan ini yaitu melihat seberapa baik akurasi metode pembandingan yang digunakan.

6. BAB VI HASIL DAN DISKUSI

Pada bab ini membahas hasil dari pengujian model modifikasi *Simulated Annealing-Extreme Learning Machine*. Pada bab ini akan dilihat bagaimana hasil akurasi pada metode yang digunakan kemudian dibandingkan dengan metode pembandingan. Pada bab ini kemudian dilihat validitas dari pakar terhadap hasil akurasi metode yang didapat

7. BAB VI PENUTUP

Pada bab ini membahas kesimpulan dan saran.

BAB 2 LANDASAN PUSTAKA

Landasan kepustakaan pada penulisan tesis ini berisikan uraian tentang Jarak Pagar, pembahasan tentang penelitian sebelumnya serta dasar teori tentang metode yang terkait.

2.1 Penyakit Tanaman Jarak Pagar

Beberapa penyakit yang menyerang tanaman jarak pagar, antara lain : antraknosa, bercak daun, busuk arang, busuk pujuk, embun tepung, hawar bakteri, layu bakteri, layu fusarium (Yulianti & Hidayah, 2015). Penyakit yang disebutkan merupakan penyakit yang bukan bersifat fisiologis atau masuk kategori hama. Penyakit ini sering terjadi di daerah Indonesia karena penyakit ini sering muncul di iklim tropis (Yulianti & Hidayah, 2015).

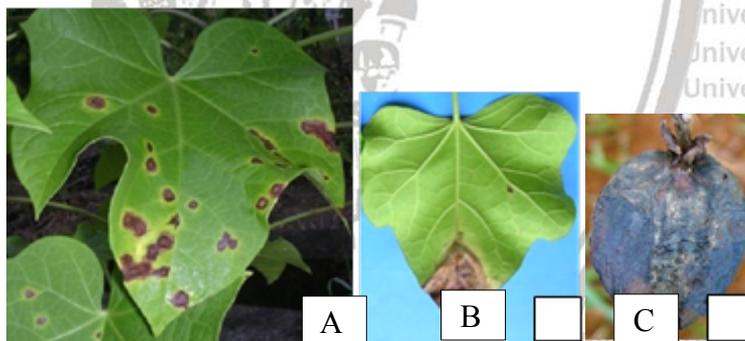
1 Antraknosa

a. Penyebab

Jamur *Colletotrichum gloeosporioides* (Penz.) Sacc.; *Colletotrichum capsici* (Syd.) Butle Bispy [wawancara].

b. Gejala

- Daun memiliki bercak-bercak cokelat kehitaman bulat yang dibatasi halo berwarna kuning.
- Jika menyerang pinggir daun bercak berbentuk tidak beraturan.
- Pada buah juga terdapat bercak cokelat kehitaman pada permukaannya.
- Pada kondisi lembab terlihat pustul-pustul hitam



Gambar 0.1 Gejala antraknosa pada jarak pagar.

A dan B: pada daun, serta C: pada buah jarak pagar

Sumber: (Yulianti & Hidayah, 2012)

2 Bercak Daun

a. Penyebab

Ada dua macam jenis bercak daun, yaitu bercak daun *Cercospora* dan bercak daun *Alternaria*. Bercak daun *Cercospora* disebabkan oleh jamur *Cercospora jatrophiicola* Speg. Bercak daun *Alternaria* disebabkan oleh *Alternaria* [wawancara].

b. Gejala

- Serangan *Cercospora* menyebabkan nekrotik tidak beraturan berwarna kecokelatan pada daun.
- Serangan *Alternaria* menyebabkan bercak daun bulat bercincin (melingkar-lingkar)



Gambar 0.2 Gejala bercak daun cercospora pada daun

Sumber: (Yulianti & Hidayah, 2012)

3 Busuk Arang

Penyakit ini disebabkan oleh jamur yang mana cukup berbahaya karena memiliki kisaran inang yang cukup luas, yakni sekitar 400 spesies tanaman. Tanaman yang sakit mampu melakukan *recovery* dengan pembentukan akar sekunder pada akar – akar yang mengalami pembusukan [wawancara].

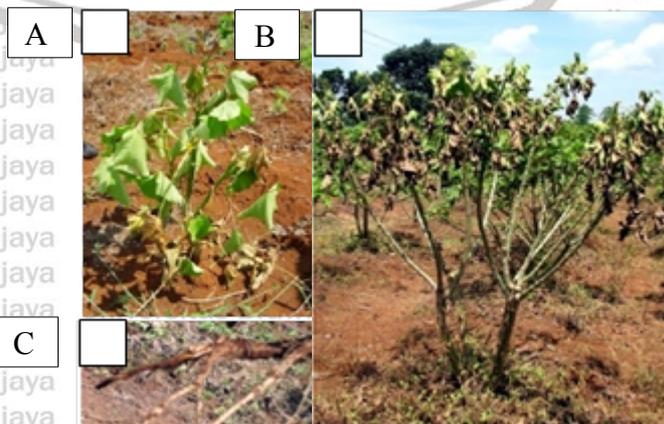
a. Penyebab

Jamur *Rhizoctonia bataticola* (Taub)

b. Gejala

Gejala awal :

- Pada hipokotil pada batas permukaan tanah sebagai bercak cokelat terang
- Berkembang menjadi cokelat gelap di sekeliling hipokotil.
- Jaringan kulit kayu bagian dalam berwarna hitam.
- Kulit luar pangkal batang tersobek – sobek dan terlihat pustule hitam.
- Daun-daun layu pada seluruh bagian tanaman secara tiba-tiba.
- Dalam waktu kurang dari 1 minggu tanaman muda mati
- Jika tanaman dicabut perakaran busuk kering bercak – bercak hitam.
- Gejala pada kondisi yg kurang mendukung pathogen
- Pada kondisi yang kurang mendukung : daun bagian bawah layu
- Kemudian daun menguning terlebih dahulu sebelum akhirnya rontok.
- Jika penyakit berlanjut, tanaman akan mati.
- Jika tanaman yang sakit dicabut maka pada akar akan terlihat busuk kehitaman disertai dengan adanya sklerosia jamur berwarna hitam.



Gambar 0.3 Gejala penyakit busuk arang.

A: Tanaman dan seluruh daun layu tiba-tiba;

- B: dalam waktu kurang dari satu minggu tanaman mati;
 C: jika tanaman dicabut perakaran busuk kering berbecak-becak hitam

Sumber: (Yulianti & Hidayah, 2012)

4 Busuk Pucuk

Penyakit yang masih belum diketahui penyebabnya. Penyebab kanker batang pada tanaman jarak pagar yang terjadi di Malaysia [wawancara].

a. Penyebab

Penyebabnya yaitu *Lasiodiplodia theobromae* (Pat.) Griffon dan Maubl.

b. Gejala

- Bercak cekung yang berkembang menjadi bercak nekrotik pada batang
- Busuk mulai dari pucuk
- Daun/tunas pucuk rontok
- Batang terlihat gundul hingga berkembang ke bagian bawah tanaman.
- Bagian yang busuk biasanya berair
- Bagian pucuk mengering.



Gambar 0.4 Gejala busuk pucuk
 Sumber: (Yulianti & Hidayah, 2012)

5 Embun Tepung

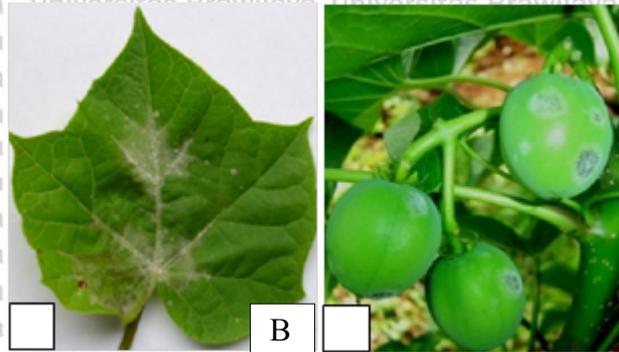
Embun tepung adalah cendawan obligat. Ciri menonjol yang ditunjukkan yaitu koloni berwarna putih agak abu-abu, miselia bersekat dan bercabang; konidiofor tegak, silindris, hialin, bersekat dengan konidium tunggal yang besar (dibandingkan konidioforanya), licin, sedikit lonjong, satu sel, dan hialin [wawancara].

a. Penyebab

Jamur *Pseudoidium jatrophae*

b. Gejala

- Adanya embun tepung berwarna putih pada permukaan daun serta buah.
- Daun gugur atau pucuk tidak berkembang dan mati
- Buah-buah muda yang terserang biasanya berubah bentuk dan gugur.
- Jika serangan terjadi pada daun yang baru saja berkembang akan menyebabkan perubahan warna menjadi kusam, daun lemas dan tepi-tepinya agak mengeriting.



Gambar 0.5 Gejala penyakit embun tepung pada tanaman jarak pagar.

A: pada daun; dan B: pada buah jarak pagar

Sumber: (Yulianti & Hidayah, 2012)

6 Hawar Bakteri

a. Penyebab

Bakteri *Xanthomonas campestris*.

b. Gejala

- Bercak berair berbatas pertulangan daun membentuk bercak bersudut
- Bercak menjadi agak kehitaman dan permukaan bawah daun tampak mengkilat



Gambar 0.6 Gejala hawar daun *Xanthomonas* sp. pada daun jarak pagar

Sumber: (Yulianti & Hidayah, 2012)

7 Layu Bakteri

Penyakit layu bakteri disebabkan oleh bakteri yang mana penyebarannya melalui tanah. Tanah yang lembab merupakan lingkungan yang sangat disukai bakteri ini. Selain tanaman jarak pagar dewasa penyakit ini juga menyerang tanaman bibit [wawancara].

Bakteri ini memiliki kisaran inang yang sangat luas yaitu tembakau, serta tanaman hortikultura seperti cabai, tomat, terong, serta pisang. Tingkat keparahan dan perkembangan penyakit ini tergantung pada tingkat ketahanan tanaman terhadap penyakit selain itu juga tergantung kondisi lingkungan [wawancara].

a. Penyebab

Bakteri *Ralstonia solanacearum*

b. Gejala

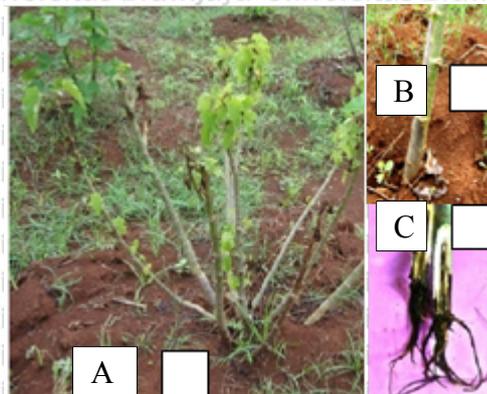
Gejala awal :

- Busuk dimulai dari akar menjalar ke pangkal batang dan satu sisi tanaman.
- Perakaran tanaman yang sakit busuk basah, jika dibelah terlihat pembuluh maupun empulur berubah coklat.

- Jika bagian yang sakit dimasukkan ke dalam air akan terlihat eksudat (cair, merupakan massa bakteri) keluar dari pembuluh batang.

Gejala berat :

- Tanaman mati tiba – tiba



Gambar 0.7 Gejala layu bakteri pada tanaman jarak pagar.

A: pada tanaman dewasa; B: pembusukan dimulai dari perakaran menjalar ke pangkal batang dan satu sisi; C: perakaran tanaman yang sakit busuk basah, jika dibelah terlihat pembuluh maupun empulur berubah cokelat

Sumber: (Yulianti & Hidayah, 2012)

8 Layu Fusarium

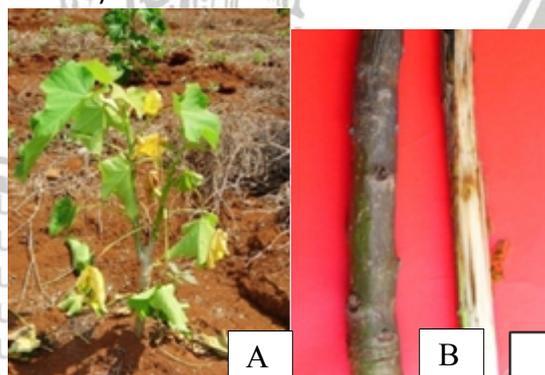
Penyakit ini cukup serius ditemukan pada tahun 2006. Di Indonesia tidak banyak ditemukan penyakit ini. *F. moniliforme* merupakan penyebab utama kematian jarak pagar di Haryana, India dengan tingkat kematian hingga 25 % [wawancara].

a. Penyebab

Penyakit ini disebabkan oleh jamur *Fusarium* spp. .

b. Gejala

- Tanaman akan layu disertai daun kekuningan
- Batang dibelah akan terlihat bagian yang berkayu bergaris-garis kecokelatan serta perakaran
- Pangkal batang (leher akar) busuk hitam.



Gambar 0.8 Gejala layu Fusarium pada tanaman jarak pagar

A: tanaman layu; B: jika batang dibelah, bagian berkayu bergaris-garis cokelat

Sumber: (Yulianti & Hidayah, 2012)

2.2 Penelitian Sebelumnya

Beberapa metode sudah dilakukan oleh peneliti untuk mendapatkan pendekatan akurasi terbaik, yaitu *Fuzzy Neural Network*, *Simulated Annealing-Fuzzy Neural Network*, *Multi Layer Perceptron* dan *Extreme Learning Machine*.

2.2.1 Fuzzy Neural Network (FNN)

Fuzzy Neural Network merupakan suatu algoritme yang mencari parameter himpunan *fuzzy* menggunakan pendekatan jaringan syaraf tiruan (Lin *et al.*, 2013; J. W. Lin, Hwang, & Becker, 2003). Parameter himpunan *fuzzy* yang didapat dari pendekatan jaringan syaraf tiruan kemudian digunakan sebagai parameter untuk melakukan penentu keputusan menggunakan metode *fuzzy* yang digunakan (Utomo, Mahmudy, & Anam, 2017).

Penelitian menggunakan FNN untuk klasifikasi penyakit tanaman jarak pagar pernah dilakukan menghasilkan rata-rata akurasi sebesar 11,2% dan akurasi terbaik sebesar 30% (Saragih *et al.*, 2017). Penelitian ini menggunakan fungsi keanggotaan yang didapat dari pakar dan menggunakan 135 data latih dan 31 data uji. Kelemahan pada penelitian ini yaitu fungsi keanggotaan yang ditetapkan oleh pakar baik itu derajat keanggotaan *input* dan *output* tidak tepat sehingga akurasi yang didapat rendah dan perlu adanya optimasi fungsi keanggotaan untuk penelitian selanjutnya.

2.2.2 Simulated Annealing-Fuzzy Neural Network

Simulated Annealing (SA) merupakan sebuah algoritme yang bekerja untuk optimasi (Yadav, *et al.*, 2011). Tujuan dari optimasi pada algoritme ini yaitu untuk mencari solusi terbaik agar memberikan hasil yang lebih baik dari sebelumnya (Bayram & Sahin, 2013). SA bekerja mencari tetangga terdekat yang terbaik dan memberikan waktu komputasi yang lebih cepat (Rere *et al.*, 2015).

Pada penelitian ini SA sebagai metode optimasi fungsi keanggotaan yang didapat dari pakar. Untuk mengetahui seberapa baik hasil optimasi yang dilakukan SA maka klasifikasi menggunakan fungsi keanggotaan yang dioptimasi oleh SA menggunakan FNN. Berdasarkan hasil penelitian ini diperoleh rata-rata akurasi sebesar 19,5% dan akurasi terbaik sebesar 32,5% untuk 135 data latih dan 31 data uji (Fajri *et al.*, 2017). Kesimpulan dari penelitian ini yaitu data yang diperoleh dari pakar tidak cocok untuk kasus *fuzzy* sehingga sangat sulit untuk menentukan fungsi keanggotaan yang tepat terutama fungsi keanggotaan pada *output*.

2.2.3 Multi Layer Perceptron

Multi Layer Perceptron (MLP) biasa dikenal sebagai pembelajaran *backpropagation* pada algoritme jaringan syaraf tiruan (Chang & Aw, 1996). MLP sering digunakan sebagai metode untuk klasifikasi, klusterisasi hingga regresi pada penelitian sebelumnya (Hailesilassie, 2016). Pada metode ini terdapat tiga layer yaitu *input layer*, *hidden layer*, dan *output layer* (Hurtado *et al.*, 2013).

Pada penelitian sebelumnya menggunakan metode MLP didapatkan hasil akurasi rata-rata terbaik 9,1% dan akurasi terbaik sebesar 12,12% untuk kasus yang menggunakan data uji

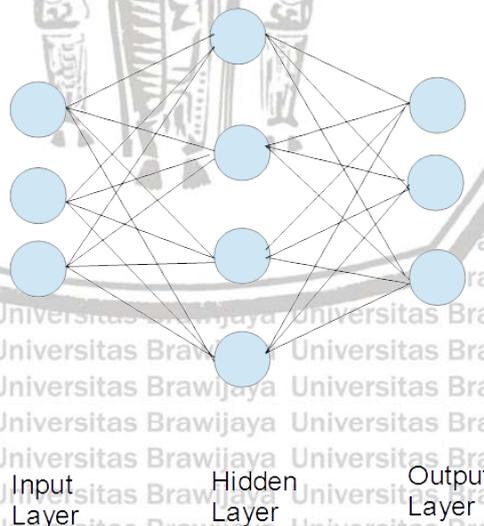
sebanyak 135 kasus dan data latih sebesar 31 kasus. Pada penelitian ini dapat disimpulkan bahwa penggunaan MLP untuk kasus identifikasi penyakit tanaman jarak pagar masih kurang bagus dan perlu pemakaian metode jaringan syaraf tiruan lainnya untuk bisa memberikan akurasi yang lebih baik lagi.

2.2.4 Extreme Learning Machine

Extreme Learning Machine (ELM) yaitu salah satu metode jaringan syaraf tiruan yang memiliki keunggulan waktu komputasi yang sangat cepat dikarenakan memiliki konsep satu kali iterasi dan dapat memberikan hasil yang baik (Huang, Zhu, & Siew, 2006). Pada penelitian sebelumnya untuk identifikasi penyakit jarak pagar menggunakan metode ELM didapatkan hasil akurasi rata-rata terbaik sebesar 60,61% dan akurasi terbaik sebesar 66,67%. Hal ini membuktikan bahwa metode ELM bekerja lebih baik dibandingkan metode pembandingan sebelumnya. Kesimpulan dari penelitian ini yaitu pentingnya pengelompokkan kelas penyakit yang memiliki kemiripan gejala agar dapat mengurangi tingkat kesalahan pada sistem dan dapat memberikan akurasi yang lebih baik lagi dari sebelumnya.

2.3 Extreme Learning Machine

Extreme Learning Machine (ELM) merupakan sebuah algoritme jaringan syaraf tiruan yang sering digunakan untuk klasifikasi, regresi, klasterisasi, dan pembelajaran fitur yang memiliki konsep satu atau lebih *hidden layer* yang bekerja dalam satu kali iterasi (Huang, Zhu, & Siew, 2006; Tang, Deng, & Huang, 2015). Keunggulan pada metode ELM ini yaitu seribu kali lebih cepat dibandingkan dengan metode *Neural Network* lainnya yang menggunakan konsep pembelajaran *backpropogation* (Huang, Zhu, & Siew, 2006). Arsitektur dari ELM dapat dilihat pada Gambar 2.9.



Gambar 0.9 Arsitektur Extreme Learning Machine

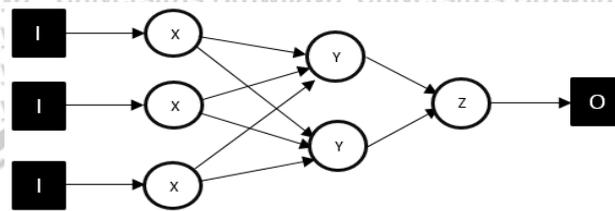
Sumber (Bueno-Crespo, García-Laencina, & Sancho-Gómez, 2013)

2.3.1 Arsitektur Neural Network

Arsitektur pada *Neural Network (NN)* memiliki hubungan antar *neuron* yang saling terkait yang kemudian lebih sering dikenal dengan lapisan-lapisan jaringan syaraf tiruan yang terdiri

dari lapisan *input*, lapisan tersembunyi atau lebih dikenal *hidden layer* dan lapisan *output* (Sutojo, Mulyanto, & Suhartono, 2011). Beberapa arsitektur jaringan yang banyak digunakan adalah lapisan tunggal, lapisan banyak, dan lapisan kompetitif. Fungsi *sigmoid* sebagai salah satu fungsi aktivasi dalam melatih NN menggunakan *backpropogation* memiliki peran penting di dalamnya (Dessy & Irawan 2012).

Pada arsitektur jaringan ada lapisan tunggal yang biasa disebut dengan *single layer* yang hanya terdiri dari lapisan *input* dan langsung terhubung dengan lapisan *output*. Lapisan banyak atau dikenal dengan *multi layer* terdiri dari lapisan *input* kemudian melalui lapisan tersembunyi dan terhubung ke lapisan *output*, yang mana *multi layer* ini sangat terkenal untuk menyelesaikan masalah yang kompleks dan mudah dalam pelatihan dan untuk lapisan kompetitif yang terdapat dalam LVQ (*Learning Vector Quantization*) (Lippmann, 1987).



Gambar 0.10 Arsitektur Jaringan Neural Network
 Sumber (Mcculloch & Pitts 1943)

2.3.2 Model Neural Network

Pada proses permodelan NN diketahui terbagi menjadi dua bagian yang terdiri dari *training* dan *testing* dimana untuk *training* yaitu proses pembelajaran dan pelatihan dari suatu sistem NN untuk mengelola nilai *input* dan bagaimana melakukan pemetaan hingga *output* sehingga memperoleh model yang sesuai dengan harapan. Saat pengaturan yang terjadi pada bobot dan bias terjadi itulah prosesnya dan proses *testing* yaitu dimana proses untuk menguji yang diteliti dari model yang sudah diperoleh berdasarkan proses *training*.

2.3.3 Proses Belajar Neural Network

Proses belajar NN jika berdasarkan pada cara memodifikasi bobotnya terbagi dua yaitu *supervised learning* dan *unsupervised learning*. Proses belajar ini dimana pada proses *training*, *input* dimasukkan kedalam sistem jaringan kemudian mengeluarkan *output* hingga *output* tersebut dibandingkan dengan target. Jika hasil *output* tidak sesuai dengan target yang diinginkan, maka akan dilakukan modifikasi bobot lagi bertujuan untuk memperoleh hasil *output* yang sesuai dengan target yang diharapkan. *Supervised learning* (terbimbing) akan dipandu sedangkan *unsupervised learning* tidak memiliki pembimbing sehingga tidak ada yang memandu dalam proses pelatihan artinya jaringan hanya akan diberi *input* tanpa ada target yang diharapkan (Sutojo, Mulyanto, & Suhartono 2011).

2.3.4 Fungsi Aktivasi

Fungsi yang diaktifkan adalah fungsi-fungsi yang akan mentransformasikan hasil-hasil inputan yang menghasilkan output tertentu. Pada jaringan saraf tiruan berarti informasi akan diterima oleh inputan. Inputan ini akan diperingkat untuk melakukan fungsi perambatan.

Fungsi ini akan menjumlahkan sejumlah inputan, hasil dari penjumlahan ini kemudian akan dibandingkan dengan nilai ambang (threshold) fungsi tertentu melalui setiap neuron. Jika nilai yang dihasilkan melewati batas maka neuron tersebut akan dikonversi jika tidak maka neuron tidak akan diaktifkan. Neuron akan menghasilkan nilai output jika ambang dilewati (Nababan & Zarlis, 2015).

Fungsi aktivasi pada penelitian ini menggunakan fungsi aktivasi sigmoid bipolar. Output fungsi aktivasi sigmoid bipolar dari fungsi ini memiliki range antara 1 sampai -1. Rumus dapat dilihat dari Gambar 2.11 dibawah.

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

Gambar 0.11 Persamaan Fungsi Aktivasi Sigmoid

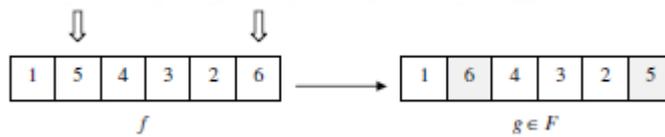
2.4 Simulated Annealing

Simulated Annealing (SA) yaitu sebuah algoritme yang bekerja untuk mengoptimasi menggunakan prinsip termodinamika statistik (Kirkpatrick, Gelatt, & Vecchi, 1983). SA berkerja berdasar pada sebuah analogi proses pendinginan dan pembekuan sebuah logam menjadi struktur kristal dengan menggunakan energi minimum yang disebut proses *annealing* (Örkcü, 2013).

Algoritme SA memiliki kelebihan dibandingkan algoritme metaheuristik yang lain seperti Algoritma Genetika dan *Evolution Strategies*. Kelebihan SA yaitu waktu yang dibutuhkan lebih sedikit dalam komputasi dan memori yang digunakan lebih sedikit (Sousa *et al.*, 2016). Algoritme SA memungkinkan solusi saat ini digantikan oleh solusi tetangga terburuk dengan kemungkinan yang sangat kecil untuk menghindari terjebak pada solusi optimum lokal (S. W. Lin & Yu, 2015). Kelemahan pada SA yaitu tidak adanya jaminan untuk mendapatkan solusi optimum, solusi yang didapat tiap iterasi tidak dapat direproduksi dan berbeda setiap proses karena bersifat stokastik dan tergolong lambat dalam perhitungan disbanding dengan algoritme *direct search* (Selim & Alsultan, 1991). *Direct search* merupakan metode yang digunakan untuk memecahkan masalah optimasi yang tidak membutuhkan informasi tentang gradien dari fungsi tujuan.

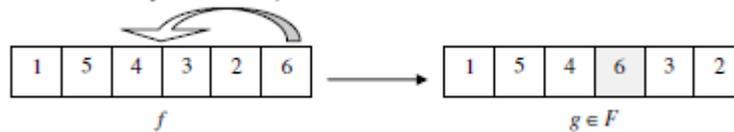
Pada optimasi nilai kepercayaan, SA akan melakukan optimasi dengan representasi solusi sesuai dengan banyak jenis penyakit. SA akan melakukan pencarian berdasarkan parameter-parameter yang ditentukan sebelumnya hingga mendapatkan solusi yang terbaik. Proses pencarian tetangga terdekat menggunakan representasi permutasi.

Ada 4 operator yang sering digunakan dalam pencarian tetangga terdekat dalam representasi permutasi, yaitu *swap mutation*, *insert mutation*, *scramble mutation* dan *inversion mutation*. *Swap mutation* merupakan menukar 2 posisi titik potong yang kita pilih untuk ditukar (Oliveira *et al.*, 2007).



Gambar 0.12 Swap Mutation

Operator kedua yaitu *insert mutation*. *Insert mutation* yaitu melakukan proses pemindahan satu titik yang kita pilih menuju posisi yang ingin dituju (Oliveira et al., 2007).



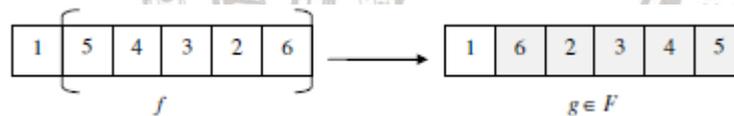
Gambar 0.13 Insert Mutation

Operator selanjutnya yaitu *scramble mutation*, dimana melakukan acak posisi di satu titik potong yang panjangnya bisa kita tentukan sendiri ataupun random (Oliveira et al., 2007).



Gambar 0.14 Scramble Mutation

Operator terakhir yaitu *inversion mutation* yang dimana melakukan perubahan posisi secara *inverse* untuk menghasilkan tetangga baru (Oliveira et al., 2007).



Gambar 0.15 Inversion Mutation

Setelah mendapatkan tetangga terdekat, pada proses *Simulated Annealing* akan memilih satu terbaik untuk dicari tetangganya pada iterasi selanjutnya dan berhenti jika sudah memenuhi jumlah iterasi yang sudah ditentukan.

Tahapan pada *Simulated Annealing* yaitu sebagai berikut (Bayram & Sahin, 2013):

1. Membangkitkan solusi awal (S), suhu awal T_0 , suhu akhir T_t , iterasi maksimum, laju penurunan suhu α (biasanya bernilai 0.9 atau 0.95), energi awal E , solusi terbaik $S_{best} = S$, dan energi terbaik $E_{best} = E$.
2. Membangkitkan solusi tetangga (S_n) dan menghitung energi E_n .
3. Apabila nilai $E_n < E$, menuju langkah 7. Apabila tidak, menuju langkah 4.
4. Menghitung perubahan energi $\Delta E = E_n - E$.
5. Membangkitkan nilai acak *uniformu* antara 0 hingga 1.
6. Apabila nilai $u < e^{\left(\frac{-\Delta E}{T}\right)}$, menuju ke langkah 7. Apabila tidak, menuju langkah 10.

7. Menerima solusi S_n sebagai S dan energi E_n sebagai E .
8. Apabila nilai $E_n < E_{best}$, menuju langkah 9. Apabila tidak, menuju langkah 10.
9. Menerima solusi baru sebagai solusi terbaik, $S_{best} = S_n$ dan $E_{best} = E_n$.
10. Apabila jumlah iterasi maksimum untuk suhu saat ini tercapai, menuju langkah 11. Apabila tidak, menuju langkah 2.
11. Mengurangi suhu $T = T_0 \times \alpha$ dimana α adalah laju penurunan suhu.
12. Apabila suhu minimal sudah tercapai, menuju langkah 13. Apabila tidak, menuju langkah 2.
13. Proses selesai, solusi ditemukan

2.5 Perumusan Hasil Pengujian

Tolak ukur untuk mengetahui seberapa baik hasil optimasi untuk identifikasi penyakit jarak pagar dilakukan menghitung hasil akurasi dari algoritme. Semakin banyak hasil identifikasi dari algoritme yang sama dengan data uji maka semakin bagus akurasi dari algoritme.

$$\text{akurasi} = \frac{\text{jumlah hasil sistem yang sama dengan data uji}}{\text{jumlah data uji}} \times 100\% \dots \dots \dots (1)$$

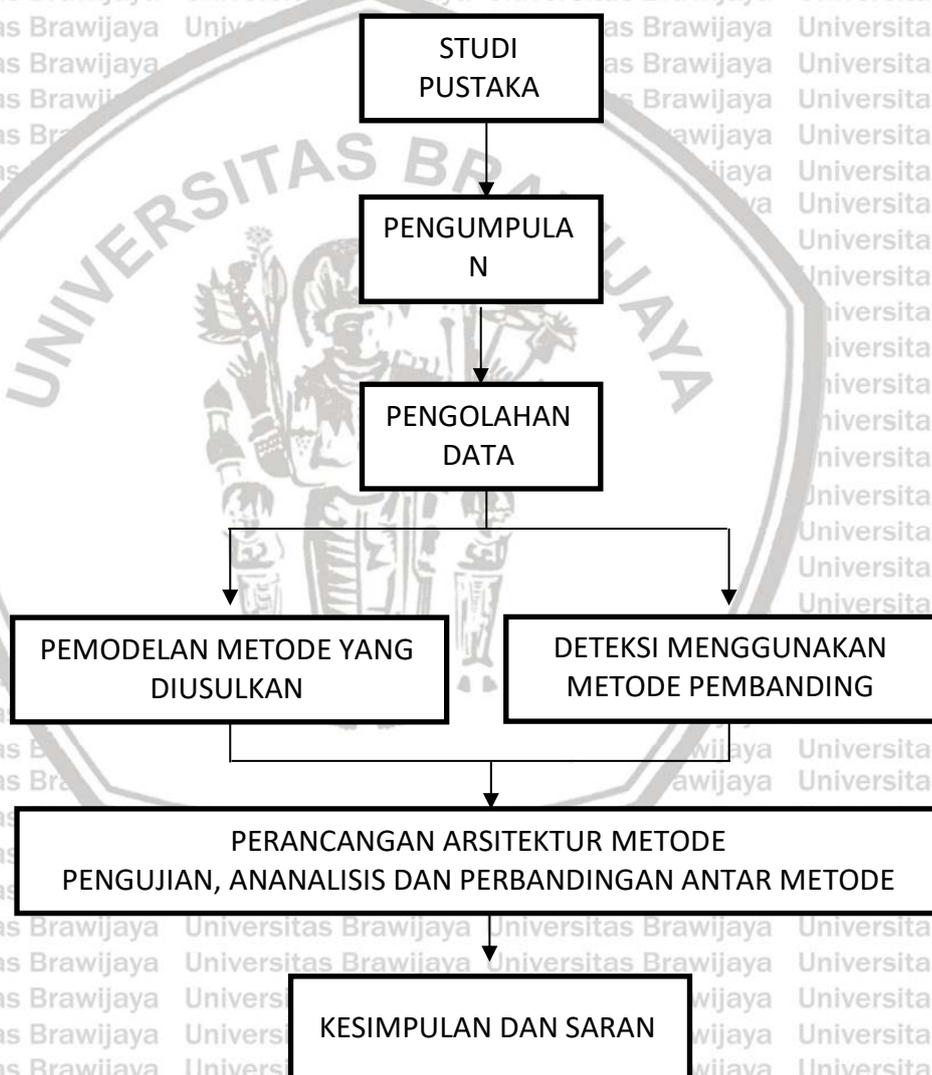


BAB 3 METODOLOGI

Pada bab ini membahas tentang metode penelitian yang digunakan pada modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar menggunakan.

3.1 Metode Penelitian

Pada penelitian ini dilakukan untuk melakukan perbandingan metode yang diusulkan dengan metode pembanding berdasarkan penelitian yang telah dilakukan. Berikut ini adalah alur dari metode penelitian yang dilakukan



Gambar 0.1 Diagram Tahapan Penelitian

3.2 Studi Pustaka

Studi pustaka yang dikumpulkan berasal dari buku, publikasi yang berasal dari jurnal ilmiah serta informasi yang didapat dari *internet* melalui *website* yang menyediakan informasi tersebut. Studi pustaka mencakup tentang jarak pagar, *Extreme Learning Machine*, *Simulated Annealing* dan penelitian sebelumnya yang menggunakan *Extreme Learning Machine* ataupun *Simulated Annealing*.

3.3 Pengumpulan Data

Pengumpulan data pada penelitian ini didapat dari penelitian langsung di lapangan. Lokasi pengumpulan data penyakit diambil di daerah Situbundo dan Malang. Data yang diambil kemudian diolah agar bisa digunakan dalam sistem ini.

3.4 Pengolahan Data

Terdapat 30 *input* dan 9 *output*, dimana *input* merupakan gejala yang dialami tanaman jarak pagar dan *output* merupakan jenis penyakit. Data yang didapat sebanyak 166 data yang kemudian dibagi menjadi 80% data latih dan 20% data uji. Tujuan dari pembagian ini agar melatih metode yang digunakan sehingga bisa memberikan akurasi yang lebih baik. Berikut ini adalah data nilai kepercayaan yang digunakan.

Tabel 0.1 Tabel Nilai Kepercayaan Penyakit Tanaman Jarak Pagar

No.	Gejala	Nilai Densitas Gejala tiap Penyakit								
		P1	P2	P3	P4	P5	P6	P7	P8	P9
1	Bercak coklat kehitaman bulat yang dibatasi oleh halo kuning pada daun	0,9	0	0	0	0	0	0	0	0
2.	Bercak tidak beraturan pada daun	0,75	0,9	0	0	0	0	0,25	0	0
3.	Bercak/pustul hitam pada daun	0,25	0	0	0	0	0	0,25	0	0
4.	Bagian permukaan bawah daun mengkilat	0	0	0	0	0	0	0,25	0	0
5.	Bercak bulat bercincin / melingkar pada daun	0	0	0,9	0	0	0	0	0	0
6.	Daun tanaman layu tiba – tiba	0	0	0	0,9	0	0	0	0,9	0,9
7.	Daun menguning	0	0	0	0,25	0	0	0,25	0,75	0,5
8.	Daun rontok	0	0	0	0,25	0,75	0	0	0,75	0
9.	Empun tepung berwarna putih pada daun	0	0	0	0	0	0,9	0	0	0
10.	Bercak berair berbatas pertulangan	0	0	0	0	0	0	0,9	0	0
11.	Pada hipokotil terdapat bercak coklat	0	0	0	0,25	0	0	0	0	0
12.	Tunas pucuk rontok	0	0	0	0	0,75	0	0	0	0
13.	Pucuk tidak berkembang	0	0	0	0	0,5	0,5	0	0	0
14.	Pucuk mati	0	0	0	0	0,5	0,5	0	0	0
No.	Gejala	Nilai Densitas Gejala tiap Penyakit								

		P1	P2	P3	P4	P5	P6	P7	P8	P9
15	Pucuk kering	0	0	0	0	0	0	0	0	0,25
16	Pucuk / ujung batang busuk berair kemudian menyebar ke seluruh bagian tanaman	0	0	0	0	0,9	0	0	0	0
17	Batang gundul	0	0	0	0	0,5	0	0	0,5	0
18	Bagian busuk dibelah akan terlihat pembuluh dan empelur coklat.	0	0	0	0	0,75	0	0	0	0
19	Jaringan kulit batang bagian dalam berwarna hitam	0	0	0	0,25	0	0	0	0	0
20	Kulit luar pangkal batang tersobek-sobek dan terdapat pustul/ bercak hitam	0	0	0	0,5	0	0	0	0	0
21	Jika batang dibelah, bagian berkayu bergaris – garis kecokelatan	0	0	0	0	0	0	0	0	0,75
22	Busuk dimulai dari akar hingga ke pangkal batang dan satu sisi	0	0	0	0,5	0	0	0	0,75	0,5
23	Akar busuk kering dan terdapat pustul/bercak hitam	0	0	0	0,75	0	0	0	0	0,5
24	Akar tanaman yang sakit mengeluarkan eksudat	0	0	0	0	0	0	0	0,75	0
25	Akar tanaman busuk basah, jika dibelah akan terlihat pembuluh maupun empelur berubah coklat	0	0	0	0	0	0	0	0,5	0
26	Buah berubah bentuk dan gugur	0	0	0	0	0	0,75	0	0	0
27	Bercak kecokelatan pada permukaan buah	0,25	0	0	0	0	0	0	0	0
28	Empun tepung berwarna putih pada buah	0	0	0	0	0	0,9	0	0	0
29	Tanaman layu	0	0	0	0,9	0,5	0	0	0,9	0,75
30	Tanaman mati	0	0	0	0,75	0	0	0	0,5	0,5

Keterangan:

P1 : Antraknosa

P2 : Bercak Daun Cercospora

P3 : Bercak Daun Altenaria

P4 : Busuk Arang

P5 : Busuk Pucuk

P6 : Embun Tepung

P7 : Hawar Bakteri

P8 : Layu Bakteri

P9 : Layu Fusarium

Berikut ini adalah daftar nama patologi beserta penyakit yang disebabkan pada tanaman jarak pagar(Machado & Pereira, 2012).

- Antraknosa diserang *Colletotrichum gloeosporioides* dan *Colletotrichum capsici*.
- Bercak daun cercospora diserang *Cercospora jatrophiicola* dan *Cercospora jatrophiigena*.
- Embun tepung disebabkan oleh *Pseudoidium jatrophae*.
- Busuk pucuk disebabkan oleh *Lasiodiplodia theobromae*.



Berikut ini merupakan contoh data latih penyakit yang didapat dari hasil penelitian yang sudah dilakukan.

Tabel 0.2 Contoh Data Latih Penyakit

Data latih ke-	Penyakit	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	G24	G25	G26	G27	G28	G29	G30		
1	Antraknosa	0.9	0.75	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	Bercak Daun	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	Bercak Alter	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	Busuk Arang	0	0	0	0	0	0.9	0.25	0.25	0.75	0.75	0.25	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
5	Busuk Arang	0	0	0	0	0	0.9	0.25	0.25	0.75	0.75	0.25	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
6	Busuk Arang	0	0	0	0	0	0.9	0.25	0.25	0.75	0.75	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
7	Busuk Arang	0	0	0	0	0	0.9	0.25	0.25	0.75	0	0	0.25	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
8	Busuk Arang	0	0	0	0	0	0.9	0.25	0.25	0.75	0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
9	Busuk Arang	0	0	0	0	0	0.9	0.25	0	0	0	0.25	0.25	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
10	Busuk Arang	0	0	0	0	0	0.9	0.25	0	0	0.75	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
11	Busuk Arang	0	0	0	0	0	0.9	0	0	0	0	0.25	0.25	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.5	0	0	
12	Busuk Pucuk	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0.75	0	0.5	0.5	0	0	0	0	0	0.5	0	0	0	0	0	
13	Busuk Pucuk	0	0	0	0	0	0	0	0.75	0	0	0	0	0	0.75	0	0.9	0.75	0	0.5	0.5	0	0	0	0	0	0	0	0	0	0	0	
14	Busuk Pucuk	0	0	0	0	0	0	0	0.75	0	0	0	0	0	0.75	0.5	0.9	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	
15	Embun Tepung	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0.9	0.5	0.5	0	0.75	0	0	0	0	0	0	0	0	
16	Embun Tepung	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0.9	0.5	0.5	0	0	0	0	0	0	0	0	0	0	
17	Embun Tepung	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0.9	0	0	0	0	0	0	0	0	0	
18	Hawar Bakteri	0	0.25	0.25	0	0	0	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	
19	Hawar Bakteri	0	0	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0.25	0	0	0	0	0	
20	Layu Bakteri	0	0	0	0	0	0.9	0.75	0.75	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.75	0.75	0	0
21	Layu Bakteri	0	0	0	0	0	0.75	0.75	0.5	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.75	0.75	0	0
22	Layu Bakteri	0	0	0	0	0	0.9	0	0	0.5	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.75	0.75	0	0
23	Layu Bakteri	0	0	0	0	0	0.9	0.75	0.75	0.5	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0
24	Layu Fusarium	0	0	0	0	0	0.9	0.5	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0.25	0	0.5	0.25	0.75
25	Layu Fusarium	0	0	0	0	0	0.9	0.5	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0.25	0	0.5	0.25	0.75
26	Layu Fusarium	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.25	0	0	0.25	0
27	Layu Fusarium	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0.25	0	0.5	0.25	0.75

3.5 Pemodelan Metode

Mekanisme dari metode yang ingin dikerjakan yaitu pertama mulai dari proses *Simulated Annealing* (SA). Pada proses SA akan dilakukan optimasi untuk pemberian nilai bobot pada *neuron Neural Network*. Pada SA terdapat modifikasi proses pada representasi permutasinya. Pada proses representasi permutasi menggunakan metode baru yang memiliki proses pertama melakukan pemotongan dua titik yang panjang kedua titik ditentukan secara random. Setelah melakukan pemotongan kemudian melakukan pertukaran posisi kedua titik tersebut, kemudian salah satu titik potong akan dilakukan pengacakan posisi baru pada titik potong itu.

Untuk mengetahui seberapa baik hasil optimasi dari SA maka digunakan *Extreme Learning Machine* (ELM) untuk melakukan identifikasi penyakit. Jumlah data yang digunakan yaitu sebanyak data latih sebanyak 80% dan data uji 20% dari total 166 data kasus penyakit tanaman jarak pagar. Setelah didapat nilai kepercayaan terbaik maka kemudian akan disimpan sebagai generasi terbaru untuk proses iterasi selanjutnya. Proses SA dan ELM ini terus berlanjut hingga sejumlah iterasi yang telah ditentukan dari awal.

3.6 Pengujian

Tahapan pada pengujian penelitian ini yaitu:

1. Pengujian arsitektur *Extreme Learning Machine*

Pada pengujian ini dilakukan untuk melihat parameter *Extreme Learning Machine* terbaik yang didapat dalam melakukan identifikasi penyakit tanaman jarak pagar. Pada pengujian ini dilihat berdasarkan akurasi sistem yang didapat.

2. Pengujian parameter *Simulated Annealing* dengan modifikasi. Pada pengujian ini dilakukan pengujian menggunakan operator baru pada *Simulated Annealing* yang diusulkan pada penelitian ini untuk mengelompokkan penyakit berdasarkan kemiripan gejala. Setelah itu kemudian membandingkan dengan metode pembandingan yang sudah dilakukan.

3.7 Metode Pembandingan

Metode pembandingan yang digunakan yaitu *Fuzzy Neural Network*, *Simulated Annealing-Fuzzy Neural Network*, *Multi Layer Perceptron*, *Simulated Annealing-Multi Layer Perceptron* dan *Simulated Annealing-Multi Layer Perceptron* dengan operator baru yang diusulkan. Untuk mengetahui seberapa baik dari tiap-tiap metode pembandingan yang digunakan akan diukur berdasarkan pengujian akurasi yang dilakukan pada tiap-tiap metode pembandingan. Untuk keunggulan dan penjelesan masing-masing metode dapat dilihat pada bab 2.

BAB 4 PERANCANGAN

4.1 Deskripsi Perancangan Simulated Annealing-Extreme Learning Machine

Penelitian ini menggunakan modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit jarak pagar dengan perancangan yang sudah disusun. *Input* variabel berupa gejala penyakit sebagai pendukung untuk identifikasi penyakit tanaman jarak pagar kemudian diproses untuk menghasilkan solusi menggunakan metode modifikasi *Simulated Annealing-Extreme Learning Machine*. Hasil yang akan diberikan merupakan nilai akurasi tertinggi yang mana semakin tinggi akurasi yang dihasilkan maka semakin bagus metode yang digunakan untuk identifikasi penyakit jarak pagar.

4.2 Contoh Solusi Manual

Identifikasi penyakit jarak pagar terlebih dahulu harus diketahui gejala apa saja yang muncul pada tanaman jarak pagar. Contoh data penyakit gejala jarak pagar dapat dilihat pada Lampiran B.

Contoh solusi manual untuk identifikasi penyakit jarak pagar berdasarkan gejala dapat dilihat pada Gambar 4.1.

Kasus 1	
1.	Bercak coklat kehitaman bulat yang dibatasi oleh halo kuning pada daun
2.	Bercak tidak beraturan pada daun
3.	Bercak/pustul hitam pada daun
Identifikasi dini : Antraknosa	

Gambar 0.1 Contoh data kasus 1

Misalkan identifikasi penyakit tanaman jarak pagar yaitu antraknosa dengan gejala yaitu bercak coklat kehitaman bulat yang dibatasi oleh halo kuning pada daun, bercak tidak beraturan pada daun, dan bercak/pustul hitam pada daun. Nilai fakta gejala yang terjadi pada kasus itu sangat mempengaruhi ke metode pada penentuan penyakit jarak pagar. Semakin tinggi nilai fakta gejala yang ada, maka semakin tinggi pengaruh gejala tersebut.

Berikut ini adalah contoh kasus penyakit tanaman jarak pagar.

1. Bercak coklat kehitaman bulat yang dibatasi oleh halo kuning pada daun = 0,9
2. Bercak tidak beraturan pada daun = 0,75
3. Bercak/pustul hitam pada daun = 0,25

Identifikasi dini : antraknosa

Identifikasi penyakit tanaman jarak pagar pada kasus 1 berdasarkan nilai gejala yang dipilih yaitu masuk dalam jenis penyakit antraknosa. Nilai-nilai ini didapat dari pakar berdasarkan fakta lapangan yang ada.

Optimasi nilai bobot *neuron* pada *Extreme Learning Machine* menggunakan *Simulated Annealing* bertujuan untuk memberikan nilai bobot terbaik pada proses *Extreme Learning Machine*. Pada *Simulated Annealing* terdapat sebuah representasi solusi yang mana merupakan solusi permasalahan ini. Representasi solusi yang dibentuk dapat dilihat pada Tabel 4.1.

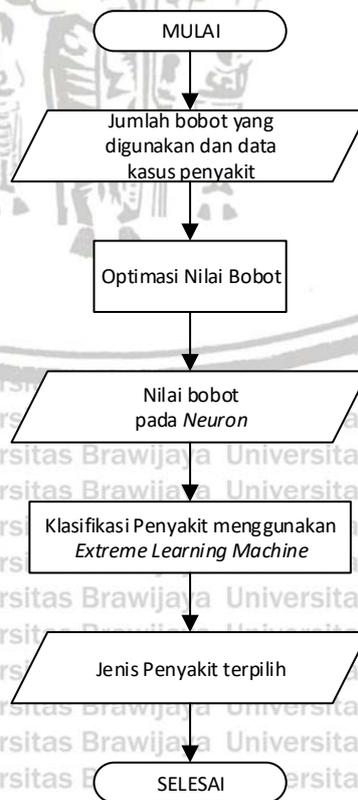
Panjang nilai untuk sebuah representasi solusi sebanyak jumlah *neuron* yang digunakan. Setiap nilai pada setiap representasi solusi merupakan perwakilan dari tiap bobot sebuah *neuron*. Pada Tabel 4.1 merupakan contoh dari representasi solusi jika terdapat 9 *neuron*. Nilai pertama hingga kesembilan mewakili bobot pertama hingga kesembilan dari sebuah *neuron*.

Tabel 0.1 Contoh representasi solusi

(x_1)	(x_2)	(x_3)	(x_4)	(x_5)	(x_6)	(x_7)	(x_8)	(x_9)
0,5	0,4	0,1	-0,1	0	-0,3	0,2	0,45	-0,3

4.3 Modifikasi *Simulated Annealing-Extreme Learning Machine*

Penerapan modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar secara umum dapat dilihat pada diagram alir yang ada pada Gambar 4.2.



Gambar 0.2 Proses Modifikasi *Simulated Annealing-Extreme Learning Machine*

Penjelasan diagram alir pada Gambar 4.2 yaitu:

1. Data berupa jumlah bobot *neuron* yang digunakan dan data kasus penyakit sebagai masukan awal untuk diproses.
2. Melakukan optimasi nilai bobot menggunakan *Simulated Annealing*.
3. Hasil dari *Simulated Annealing* yaitu nilai bobot *neuron*.
4. Melakukan proses klasifikasi penyakit menggunakan *Extreme Learning Machine* berdasarkan data kasus penyakit.

5. Mendapatkan jenis penyakit berdasarkan hasil proses klasifikasi

Proses pada Gambar 4.2 merupakan proses gambaran secara umum modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar. Pada Gambar 4.3 merupakan proses secara rinci gambaran tentang diagram alir mengenai tahapan demi tahapan modifikasi *Simulated Annealing-Extreme Learning Machine* untuk identifikasi penyakit tanaman jarak pagar

1. Melakukan inisialisasi representasi solusi menggunakan representasi permutasi yang berisi sebanyak jumlah *neuron* yang digunakan sebagai solusi awal (S)
2. Melakukan perhitungan nilai akurasi berdasarkan solusi awal sebagai nilai *fitness* untuk energi awal (E)
3. Menyimpan solusi S sebagai solusi terbaik (S_{best}) dan energi E sebagai energi terbaik (E_{best}).
4. Melakukan inisialisasi suhu saat ini (T) bernilai sama dengan suhu awal (T_0) yang telah ditetapkan sebelumnya dan iterasi bernilai 0.
5. Membentuk solusi tetangga (S_n) dengan cara melakukan perubahan satu atau beberapa nilai pada solusi S . Pencarian solusi tetangga menggunakan representasi permutasi.
6. Melakukan perhitungan nilai energi pada solusi tetangga yang sudah dibentuk (E_n) dengan menggunakan nilai akurasi yang didapat pada proses *Extreme Learning Machine*.
7. Melakukan pengecekan terhadap nilai E_n dan nilai E . Apabila nilai E_n lebih dari nilai E , maka menuju ke langkah 10. Sedangkan, apabila nilai E_n kurang dari nilai E , maka menuju ke langkah 7.
8. Melakukan perhitungan terhadap perubahan energi (ΔE) dengan menggunakan Persamaan 4.2.

$$\Delta E = E_n - E \quad (4.1)$$
9. Memilih nilai desimal secara acak (u) pada interval 0 sampai 1 untuk digunakan sebagai nilai batas dalam menentukan diterima atau tidaknya S_n sebagai solusi (S).

10. Melakukan pengecekan terhadap nilai u . Apabila nilai u kurang dari $e^{\left(\frac{-\Delta E}{T}\right)}$, maka menuju ke langkah 10. Apabila nilai u lebih dari $e^{\left(\frac{-\Delta E}{T}\right)}$, maka menuju ke langkah 14.
11. Menyimpan S_n sebagai solusi (S) dan energi E_n sebagai energi (E).
12. Melakukan pengecekan terhadap nilai E_n dan nilai E_{best} . Apabila nilai E_n lebih dari nilai E_{best} , maka menuju ke langkah 12. Apabila nilai E_n kurang dari E_{best} , maka menuju ke langkah 14.
13. Menyimpan S_n sebagai sebagai solusi terbaik (S_{best}) dan E_n sebagai energi terbaik (E_n).
14. Mengeset nilai iterasi = iterasi + 1.
15. Melakukan pengecekan terhadap nilai iterasi. Apabila nilai iterasi kurang dari atau sama dengan iterasi maksimum, maka menuju ke langkah 4. Apabila tidak, menuju ke langkah 15.
16. Melakukan pengurangan terhadap suhu T dengan menggunakan Persamaan 4.3, dimana α adalah laju penurunan suhu.

$$T = T \times \alpha \tag{4.2}$$
17. Melakukan pengecekan terhadap nilai suhu akhir (T_t). Apabila nilai T_t kurang dari atau sama dengan suhu saat ini (T), maka menuju ke langkah 17. Apabila tidak, menuju ke langkah 4.
18. Keluaran dari proses ini adalah nilai bobot *neuron* dengan energi terbaik.

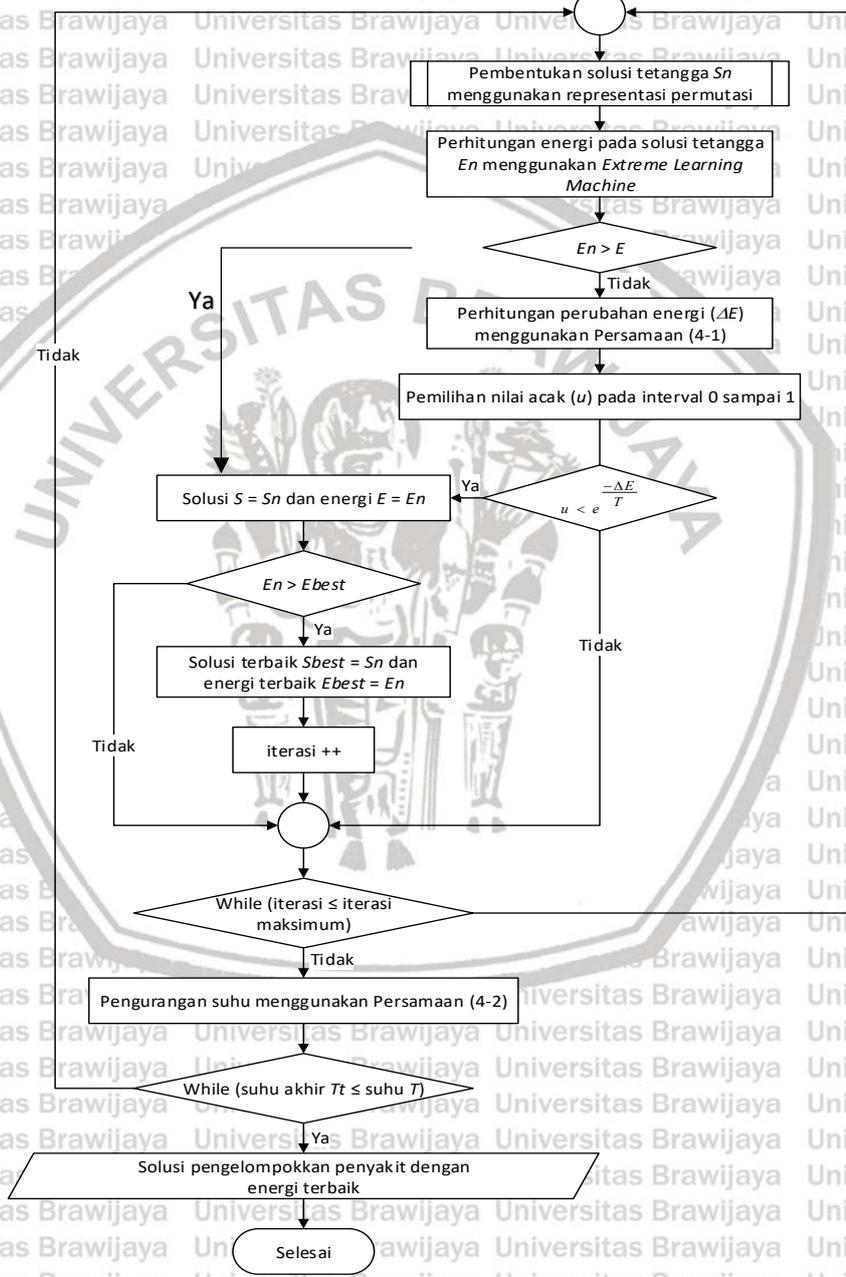
4.3.1 Representasi Solusi

Untuk mendapatkan solusi dari permasalahan yang ingin diinginkan menggunakan representasi solusi. Representasi solusi yang digunakan yaitu representasi permutasi. Setiap nilai di dalam sebuah representasi solusi mewakili bobot yang dimiliki tiap-tiap *neuron*.

Nilai-nilai nilai di dalam representasi solusi tadi harus mewakili bobot masing-masing *neuron*. Nilai pertama hingga terakhir mewakili nilai bobot dari neuron pertama hingga akhir. Tabel 4.2 merupakan contoh gambaran representasi solusi yang mewakili sembilan nilai bobot dari sembilan *neuron*.

Tabel 0.2 Contoh representasi solusi

Representasi solusi	(x_1)	(x_2)	(x_3)	(x_4)	(x_5)	(x_6)	(x_7)	(x_8)	(x_9)
P1	0,5	0,4	0,6	0,8	0,7	0,3	0,2	0,45	0,3

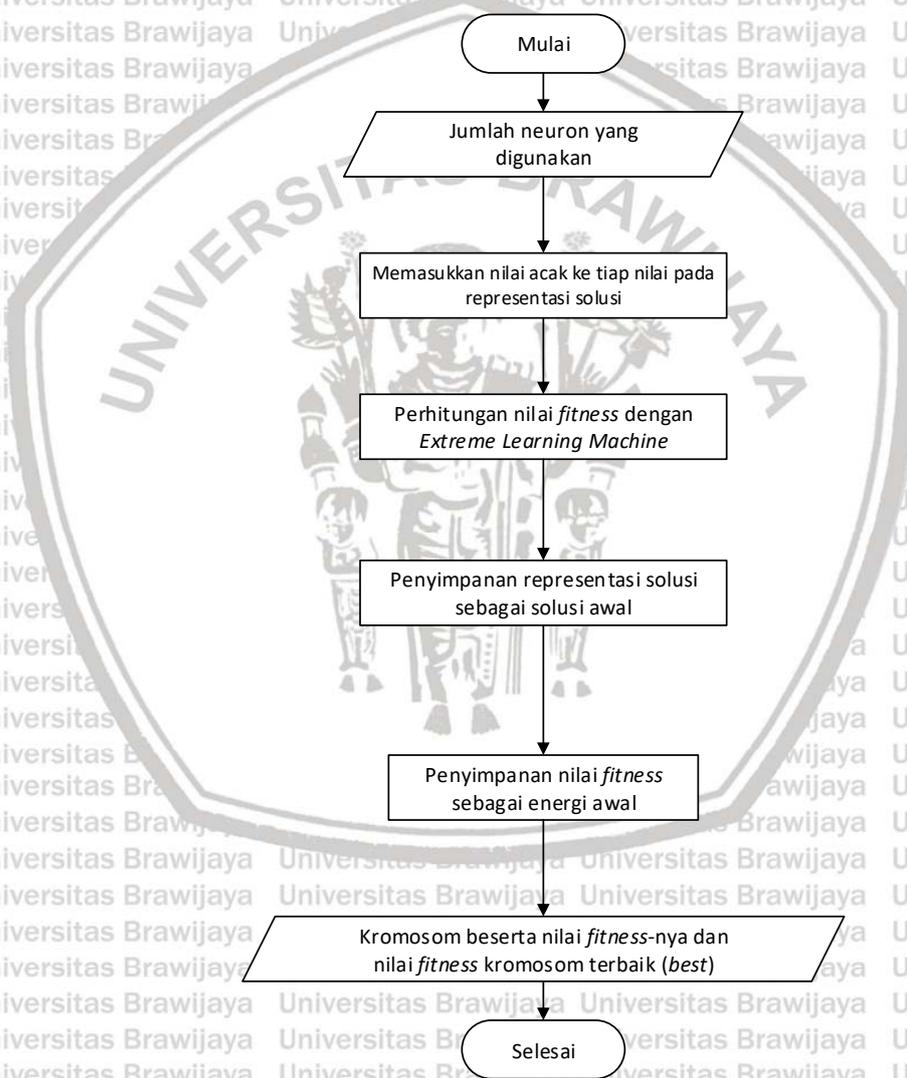


Gambar 0.3 Diagram alir modifikasi *Simulated Annealing-Extreme Learning Machine*

4.3.2 Inisialisasi representasi solusi

Proses inisialisasi representasi solusi digambarkan pada sebuah diagram alir yang ditunjukkan pada Gambar 4.4 dengan tahapan sebagai berikut:

1. Masukan pada proses ini adalah jumlah *neuron* yang digunakan;
2. Memasukkan nilai acak ke tiap nilai sebagai nilai bobot.
3. Melakukan perhitungan nilai akurasi menggunakan metode *Extreme Learning Machine* sebagai nilai *fitness* pada representasi solusi yang sudah dibentuk.
4. Menyimpan representasi solusi pertama sebagai solusi awal
5. Menyimpan nilai *fitness* representasi solusi terbaik sebagai energi awal.
6. Keluaran dari proses ini adalah sebuah representasi solusi beserta nilai *fitness*.



Gambar 0.4 Diagram alir inisialisasi representasi solusi

4.3.3 Perhitungan nilai *fitness*

Nilai *fitness* bertujuan untuk menjelaskan seberapa baik kualitas dari representasi solusi yang dihasilkan. Semakin tinggi nilai *fitness* dari representasi solusi, maka semakin baik solusi yang diberikan dari representasi solusi tersebut, begitu juga sebaliknya.

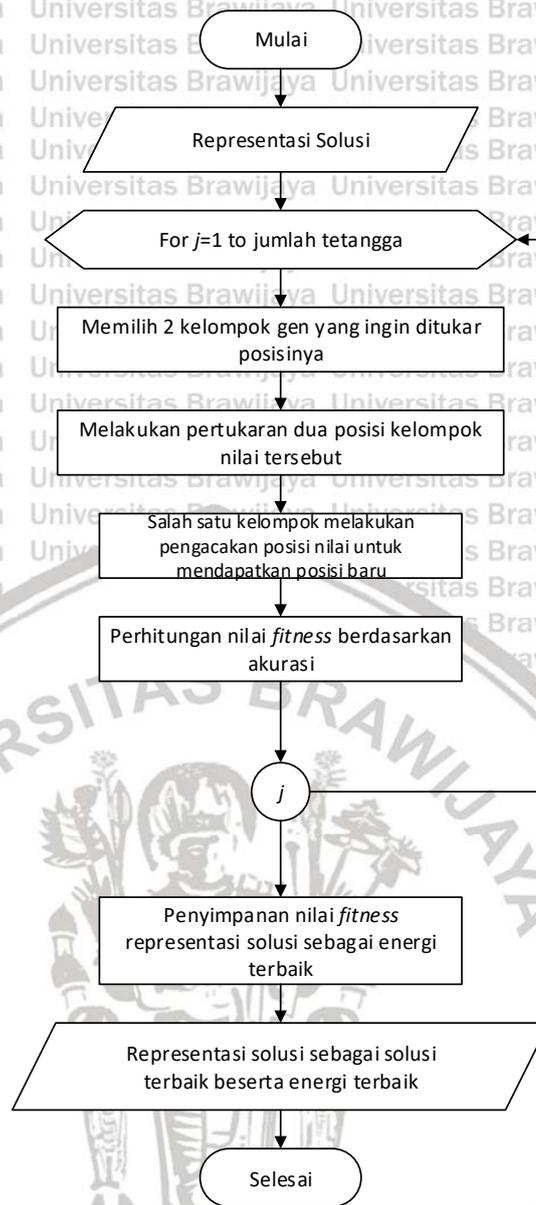
Fungsi *fitness* ditentukan berdasarkan permasalahan yang ingin diselesaikan. Fungsi *fitness* yang digunakan untuk permasalahan ini bertujuan untuk melihat seberapa baik optimasi nilai bobot untuk identifikasi penyakit jarak pagar dengan metode *Extreme Learning Machine*. Semakin tinggi akurasi yang didapat maka semakin tinggi nilai *fitness* yang diperoleh yang berarti semakin baik solusi yang diberikan oleh representasi solusi tersebut.

$$\text{akurasi} = \frac{\text{jumlah hasil sistem yang sama dengan data uji}}{\text{jumlah data uji}} \times 100\% \dots \dots \dots (2)$$

4.3.4 Mutasi

Metode mutasi yang digunakan pada penelitian ini adalah *modified mutation* yang diusulkan pada penelitian ini. Metode ini bekerja dengan cara melakukan pertukaran antar titik nilai dan melakukan perpindahan posisi tiap kelompok nilai yang terpilih. Proses mutasi digambarkan pada diagram alir yang ditunjukkan pada Gambar 4.5 dengan tahapan sebagai berikut:

1. Masukan pada proses ini adalah representasi solusi utama.
2. Membangkitkan perulangan dengan mengeset $j = j + 1$.
3. Memilih 2 kelompok nilai yang ingin ditukar posisinya. Panjang masing-masing kelompok nilai harus sama.
4. Melakukan pertukaran dua posisi kelompok nilai tersebut.
5. Salah satu kelompok melakukan pengacakan posisi nilai untuk mendapatkan posisi baru.
6. Melakukan perhitungan nilai *fitness* pada representasi solusi yang sudah dibentuk dengan mencari nilai akurasi menggunakan metode *Extreme Learning Machine*.
7. Mengulangi langkah 2 sampai 7 sebanyak jumlah tetangga yang ingin dicari.
8. Menyimpan satu nilai *fitness* representasi solusi terbaik sebagai energi terbaik.
9. Keluaran dari proses ini adalah solusi terbaik beserta energi terbaik.



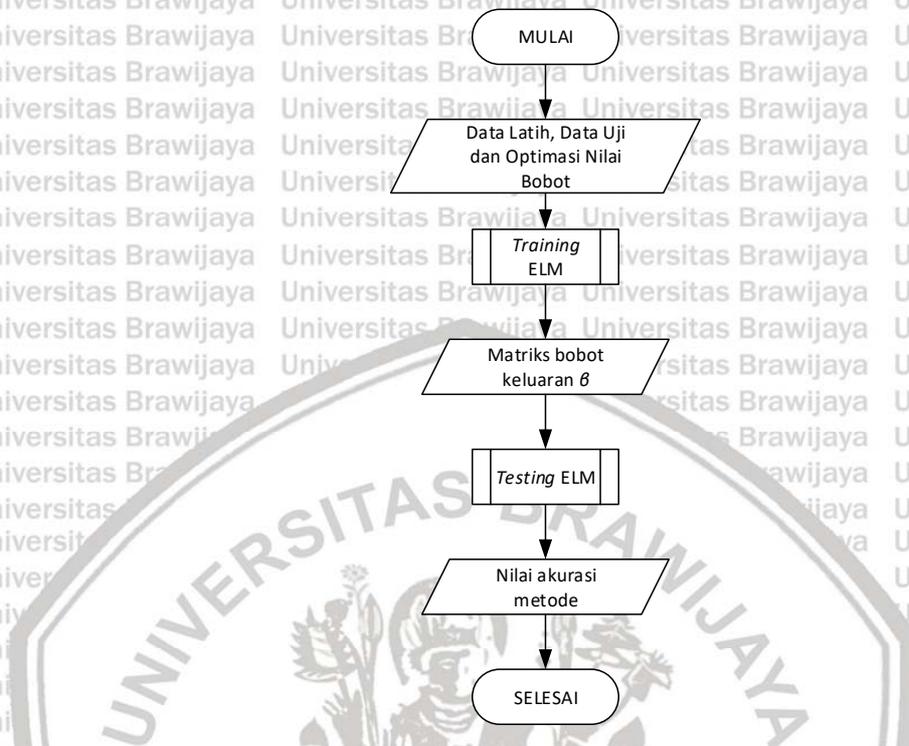
Gambar 0.5 Diagram alir mutasi

4.3.5 Extreme Learning Machine (ELM)

Extreme Learning Machine banyak digunakan pada kasus identifikasi. Metode ini sering digunakan karena waktu komputasi yang sangat cepat juga memberikan hasil yang baik. Pada Gambar 4.6 merupakan diagram alir dari metode *Extreme Learning Machine* secara umum. Berikut ini penjelasan dari alur metode *Extreme Learning Machine*.

1. Masukan pada proses ini adalah data latih, data uji, dan hasil optimasi nilai bobot dari *Simulated Annealing*.
2. Melakukan *training* ELM pada data latih.
3. Mendapatkan matriks bobot keluaran θ hasil dari *training* ELM untuk diproses ke pengujian.

4. Melakukan *testing* ELM berdasarkan matriks bobot yang sudah didapat dari *training* ELM untuk data uji.
5. Hasil dari metode ini yaitu berupa nilai akurasi pada sistem ini.



Gambar 0.6 Diagram Alir Metode Extreme Learning Machine

4.4 Proses Pengujian

Pada penelitian ini dilakukan proses pengujian untuk beberapa parameter yang berpengaruh yaitu:

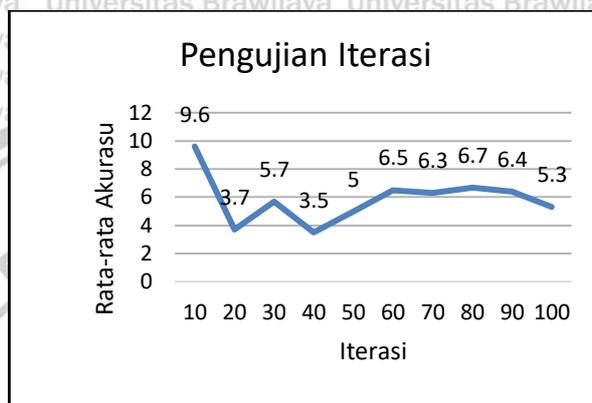
- 1) *Neuron*, untuk melihat seberapa banyak *hidden neuron* yang diperlukan pada metode *Extreme Learning Machine*. Jumlah neuron yang digunakan sebanyak jumlah neuron yang memberikan solusi terbaik pada pengujian metode *Extreme Learning Machine*.
- 2) *Temperature*, untuk melihat temperatur awal dan temperatur akhir pada proses *Simulated Annealing*.
- 3) *Cooling Factor*, semakin tinggi nilai *cooling factor* maka semakin banyak iterasi yang dilakukan pada proses *Simulated Annealing*.

BAB 5 PENGUJIAN ALGORITME PEMBANDING

Beberapa pengujian metode algoritme sebagai metode pembandingan untuk mengetahui hasil akurasi metode yang diusulkan lebih baik dari metode pembandingan pada penelitian ini sebagai berikut:

5.1 Pengujian *Fuzzy Neural Network* (FNN)

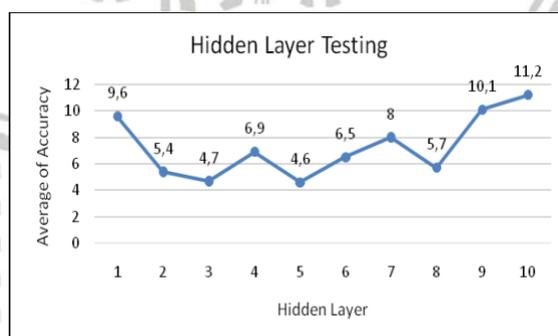
Pada subbab ini menjelaskan penerapan FNN untuk identifikasi penyakit tanaman jarak pagar. Data yang digunakan yaitu sebanyak 166 data sebagai data uji. Parameter pada pengujian FNN yaitu iterasi, *hidden layer* dan *learning rate*. Pada Gambar 5.1 merupakan grafik dari hasil pengujian parameter iterasi.



Gambar 0.1 Grafik Pengujian Iterasi

Pada Gambar 5.1 merupakan hasil dari pengujian iterasi pada FNN. Pada pengujian ini menggunakan parameter *hidden layer* sebanyak 1 dan *learning rate* sebesar 0,5. Akurasi rata-rata terbaik pada pengujian ini yaitu sebesar 9,6% pada iterasi sebanyak 10.

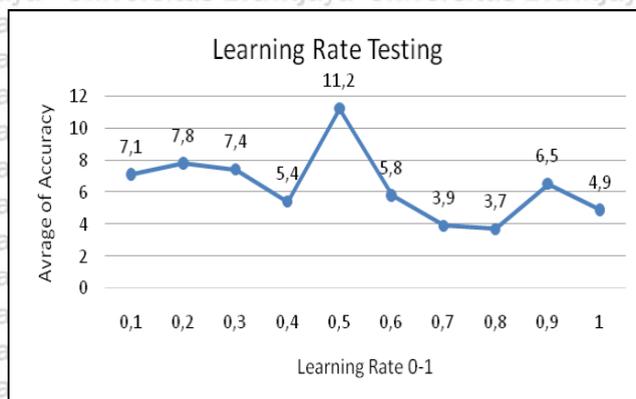
Pada pengujian selanjutnya yaitu pengujian *hidden layer*. Pengujian ini bertujuan untuk melihat seberapa banyak *hidden layer* yang digunakan untuk memberikan hasil terbaik. Hasil dari pengujian *hidden layer* dapat dilihat pada Gambar 5.2.



Gambar 0.2 Grafik Pengujian *Hidden Layer*

Pada Gambar 5.2 merupakan hasil dari pengujian *hidden layer*. Pada pengujian ini menggunakan parameter iterasi sebanyak 10 berdasarkan hasil terbaik di pengujian sebelumnya dan *learning rate* sebesar 0,5. Akurasi rata-rata terbaik yaitu sebesar 11,2% pada *hidden layer* sebanyak 10.

Pengujian terakhir yaitu pengujian *learning rate*. Pengujian *learning rate* bertujuan untuk melihat *learning rate* terbaik yang memberikan hasil maksimal. Hasil dari pengujian *learning rate* dapat dilihat pada Gambar 5.3.



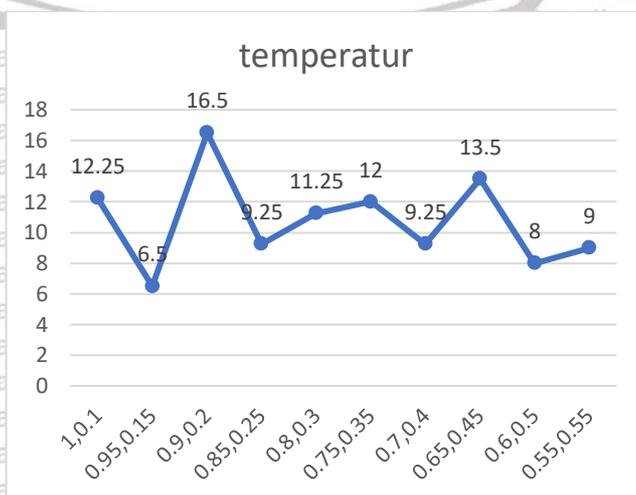
Gambar 0.3 Grafik Pengujian *Learning Rate*

Pada Gambar 5.3 merupakan hasil dari pengujian *learning rate*. Pada pengujian ini menggunakan parameter iterasi sebanyak 10 dan *hidden layer* sebanyak 10. Hasil akurasi rata-rata terbaik yaitu 11,2% dengan parameter *learning rate* sebesar 0,5.

Berdasarkan hasil pengujian FNN maka dapat disimpulkan bahwa akurasi rata-rata terbaik sebesar 11,2% dengan parameter iterasi sebanyak 10, *hidden layer* sebanyak 10 dan *learning rate* sebesar 0,5. Akurasi tertinggi FNN yaitu sebesar 30% pada pengujian ini. Kesimpulan pada pengujian ini yaitu perlunya optimasi derajat keanggotaan untuk meningkatkan hasil akurasi identifikasi penyakit jarak pagar.

5.2 Pengujian *Simulated Annealing-Fuzzy Neural Network*

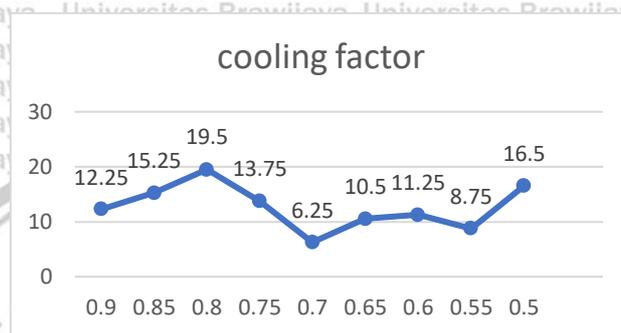
Pada subbab ini menjelaskan penerapan metode sebelumnya yaitu *Fuzzy Neural Network* (FNN) dengan *Simulated Annealing* (SA). SA berfungsi sebagai optimasi nilai derajat keanggotaan pada FNN. Pada pengujian ini akan menggunakan parameter terbaik dari hasil pengujian FNN dan melakukan pengujian parameter SA yaitu parameter temperatur, *cooling factor* dan nilai K. Data yang digunakan yaitu 166 data kasus penyakit. Pada Gambar 5.4 dapat kita lihat hasil dari pengujian temperatur.



Gambar 0.4 Grafik Pengujian Temperatur

Pada Gambar 5.4 merupakan grafik hasil pengujian temperatur. Pada pengujian temperatur akan melihat kombinasi temperatur awal dan terakhir yang terbaik untuk memberikan hasil akurasi maksimal. Parameter pengujian yang digunakan yaitu *cooling factor* sebesar 0,5 dan nilai K sebesar 1000. Parameter terbaik yaitu temperatur awal 0,9 dan temperatur akhir sebesar 0,2 dengan akurasi rata-rata terbaik sebesar 16,5%.

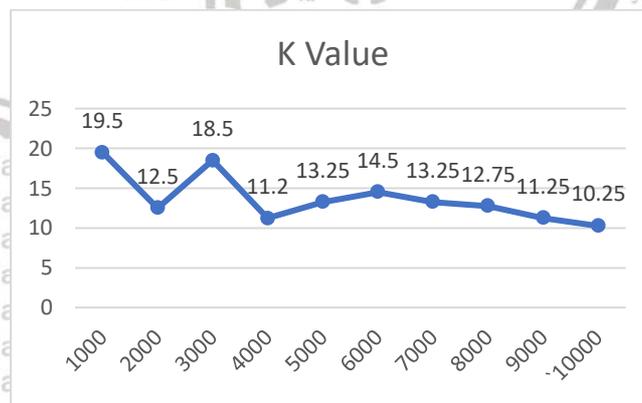
Pengujian selanjutnya yaitu pengujian yaitu pengujian *cooling factor*. Pada pengujian ini untuk melihat laju pendinginan pada proses SA. Semakin besar nilai *cooling factor* semakin banyak proses iterasi yang dilewati. Hasil dari pengujian *cooling factor* dapat dilihat pada Gambar 5.5.



Gambar 0.5 Grafik Pengujian Cooling Factor

Pada Gambar 5.5 merupakan grafik hasil pengujian *cooling factor*. Pada pengujian *cooling factor* melihat seberapa besar nilai *cooling factor* yang memberikan akurasi terbaik. Pada pengujian ini menggunakan parameter kombinasi temperatur awal 0,9 dan akhir 0,2 berdasarkan parameter terbaik sebelumnya dan nilai K sebesar 1000. Pada pengujian ini didapat akurasi rata-rata terbaik yaitu sebesar 19,5% dengan parameter *cooling factor* sebesar 0,8.

Pada pengujian terakhir yaitu pengujian nilai K. Pengujian nilai K bertujuan untuk melihat seberapa banyak pencarian tetangga yang memberikan akurasi terbaik. Hasil dari pengujian nilai K dapat dilihat pada Gambar 5.6.



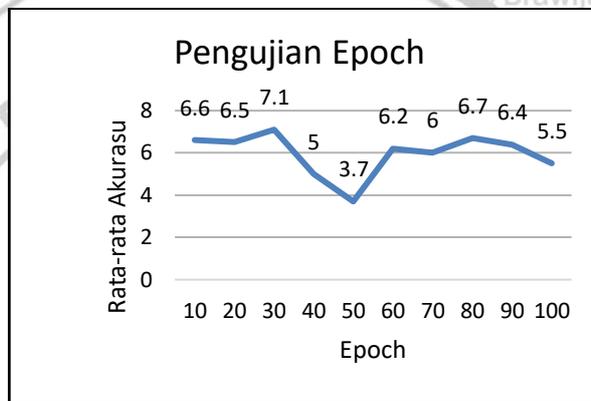
Gambar 0.6 Grafik Pengujian Nilai K

Pada Gambar 5.6 merupakan grafik hasil pengujian nilai K. Pada pengujian ini menggunakan parameter terbaik pengujian sebelumnya yaitu kombinasi temperatur awal 0,9 dan akhir 0,2 serta *cooling factor* sebesar 0,8. Pada pengujian ini didapatkan akurasi rata-rata terbaik 19,5% dengan parameter nilai K sebesar 1000.

Berdasarkan pengujian SA-FNN dapat dilihat bahwa akurasi rata-rata terbaik sebesar 19,5% dengan parameter kombinasi temperatur awal 0,9 dan akhir 0,2, cooling factor sebesar 0,8 dan nilai K sebanyak 1000. Pada pengujian ini dapat dilihat bahwa pengaruh nilai K tidak ada dan hanya membuat proses identifikasi berjalan lebih lama. Akurasi terbaik pada metode SA-FNN yaitu sebesar 32,5%.

5.3 Pengujian *Multi Layer Perceptron* (MLP)

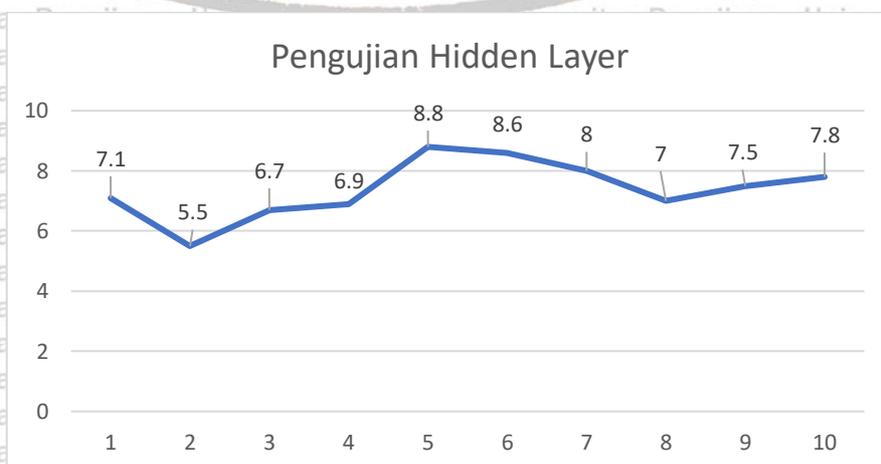
Pada subbab ini menjelaskan proses pengujian MLP. MLP merupakan salah satu metode dalam neural network dengan proses pembelajaran backpropogation. MLP digunakan untuk identifikasi penyakit tanaman jarak pagar. Data yang digunakan yaitu sebanyak 166 data penyakit. Pengujian yang dilakukan meliputi parameter *epoch*, *hidden layer* dan *learning rate*. Pengujian pertama yaitu pengujian *epoch* dengan tujuan untuk melihat seberapa banyak iterasi yang dilakukan untuk memberikan hasil terbaik. Hasil dari pengujian epoch dapat dilihat pada Gambar 5.7.



Gambar 0.7 Grafik Pengujian Epoch

Pada Gambar 5.7 merupakan hasil dari pengujian epoch pada MLP. Pada pengujian ini menggunakan parameter *hidden layer* sebanyak 1 dan *learning rate* sebesar 0,5. Akurasi rata-rata terbaik pada pengujian ini yaitu sebesar 7,1% pada iterasi sebanyak 30.

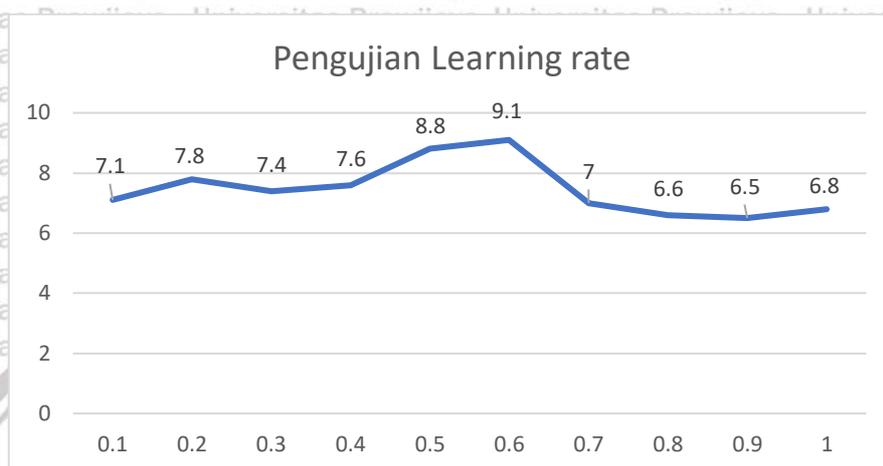
Pada pengujian selanjutnya yaitu pengujian *hidden layer*. Pengujian ini bertujuan untuk melihat seberapa banyak *hidden layer* yang digunakan untuk memberikan hasil terbaik. Hasil dari pengujian *hidden layer* dapat dilihat pada Gambar 5.8.



Gambar 0.8 Grafik Pengujian *Hidden Layer*

Pada Gambar 5.8 merupakan hasil dari pengujian *hidden layer*. Pada pengujian ini menggunakan parameter *epoch* sebanyak 30 berdasarkan hasil terbaik di pengujian sebelumnya dan *learning rate* sebesar 0,5. Akurasi rata-rata terbaik yaitu sebesar 8,8% pada *hidden layer* sebanyak 5.

Pengujian terakhir yaitu pengujian *learning rate*. Pengujian *learning rate* bertujuan untuk melihat *learning rate* terbaik yang memberikan hasil maksimal. Hasil dari pengujian *learning rate* dapat dilihat pada Gambar 5.9.



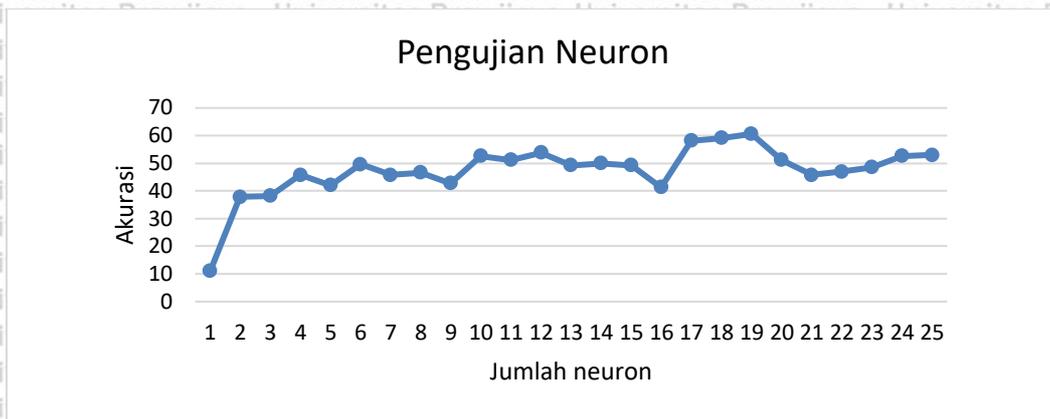
Gambar 0.9 Grafik Pengujian *Learning Rate*

Pada Gambar 5.9 merupakan hasil dari pengujian *learning rate*. Pada pengujian ini menggunakan parameter *epoch* sebanyak 30 dan *hidden layer* sebanyak 5. Hasil akurasi rata-rata terbaik yaitu 9,1% dengan parameter *learning rate* sebesar 0,6.

Berdasarkan hasil pengujian MLP maka dapat disimpulkan bahwa akurasi rata-rata terbaik sebesar 9,1% dengan parameter *epoch* sebanyak 30, *hidden layer* sebanyak 5 dan *learning rate* sebesar 0,6. Akurasi tertinggi MLP yaitu sebesar 12,12% pada pengujian ini. Kesimpulan pada pengujian ini yaitu tidak cocoknya data yang digunakan karena banyaknya kasus penyakit yang berbeda dengan gejala yang hampir sama.

5.4 Pengujian *Extreme Learning Machine (ELM)*

Pada subbab ini menjelaskan hasil pengujian dari ELM. Pada pengujian ini yaitu hanya memerlukan parameter *neuron*. Jumlah data yang digunakan sebanyak 166 data kasus penyakit dengan pembagian 80% data latih dan 20% data uji. Pengujian parameter *neuron* bertujuan untuk melihat seberapa banyak neuron di dalam proses ELM yang dapat memberikan akurasi terbaik. Pada Gambar 5.10 dapat dilihat hasil pengujian *neuron*.



Gambar 0.10 Grafik Pengujian Neuron

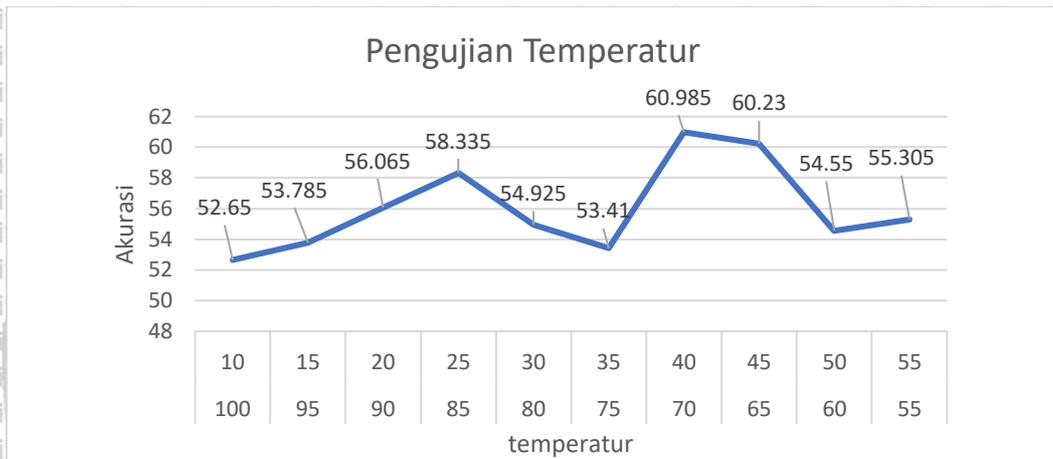
Pada Gambar 5.10 merupakan hasil dari pengujian neuron. Akurasi rata-rata terbaik yaitu sebesar 60,61% dengan jumlah neuron sebanyak 19. Parameter jumlah neuron ini selanjutnya akan digunakan untuk pengujian SA-ELM. Akurasi terbaik pada pengujian ELM yaitu sebesar 66,67%. Hal ini membuktikan bahwa ELM sudah cukup baik untuk identifikasi penyakit tanaman jarak pagar. Optimasi nilai bobot pada neuron sangat bermanfaat agar dapat menaikkan akurasi identifikasi penyakit tanaman jarak pagar.



BAB 6 HASIL DAN DISKUSI UNTUK ALGORITME UTAMA

6.1 Pengujian *Extreme Learning Machine* dan Modifikasi *Simulated Annealing*

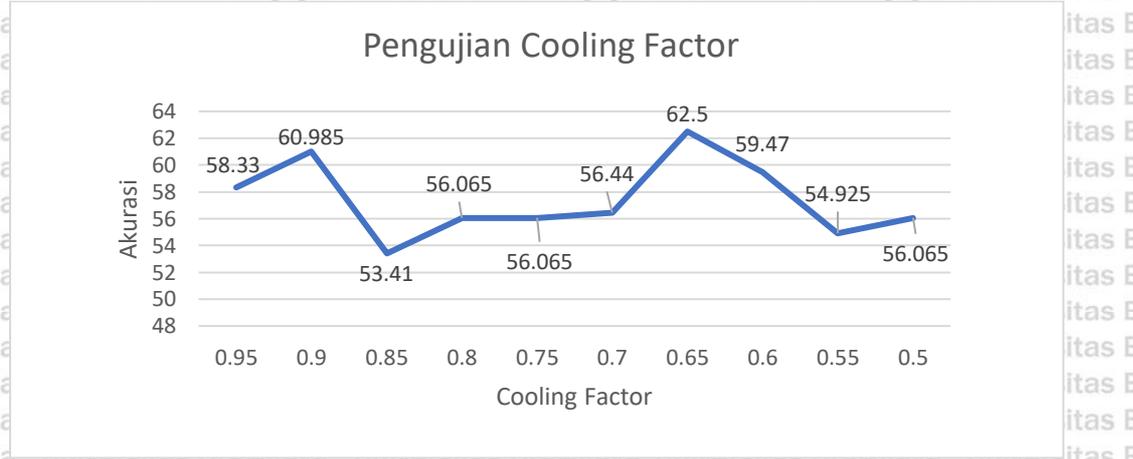
Pada pengujian ini dilakukan untuk melihat seberapa baik *Extreme Learning Machine* (ELM) yang nilai bobot pada neuron sudah dipotimasi dengan *Simulated Annealing* (SA) yang sudah dimodifikasi. Proses modifikasi pada SA yaitu pada representasi permutasi baru yang sudah diusulkan untuk melakukan pencarian tetangga terdekat untuk mendapatkan solusi terbaik. Pengujian ini menggunakan parameter temperatur dan *cooling factor*. Pada Gambar 6.1 merupakan hasil dari pengujian temperatur.



Gambar 0.1 Grafik Pengujian Temperatur

Pada Gambar 6.1 merupakan grafik hasil pengujian temperatur. Pengujian ini menggunakan parameter *neuron* sebanyak 19 berdasarkan parameter hasil pengujian terbaik ELM dan *cooling factor* sebesar 0,9. Pada pengujian temperatur didapatkan kombinasi temperatur awal sebesar 70 dan akhir sebesar 40 dengan akurasi rata-rata terbaik sebesar 60,985%.

Pengujian terakhir yaitu pengujian *cooling factor*. Pengujian ini bertujuan untuk melihat besar *cooling factor* yang memberikan solusi terbaik. Hasil dari pengujian *cooling factor* dapat dilihat pada Gambar 6.2.



Gambar 0.2 Grafik Pengujian *Cooling Factor*

Pada Gambar 6.2 merupakan hasil pengujian *cooling factor*. Parameter yang digunakan pada pengujian ini yaitu *neuron* sebanyak 19 berdasarkan parameter terbaik di pengujian ELM dan kombinasi temperatur awal 70 dan akhir 40. Akurasi rata-rata terbaik didapat sebesar 62,5% dengan parameter *cooling factor* sebesar 0,65. Akurasi maksimal yang didapat yaitu sebesar 69,7% yang berarti mengalami peningkatan dibandingkan tanpa optimasi nilai bobot.

Setelah itu kemudian ditambahkan pohon keputusan untuk penentuan penyakit tanaman jarak pagar. Pohon keputusan bertujuan untuk mempermudah sistem untuk mengambil keputusan berdasarkan pengetahuan pakar. Aturan pada Tabel 6.1 didapatkan berdasarkan pengetahuan pakar yang mana terdapat gejala yang hanya dimiliki satu penyakit, sehingga dapat mempermudah pengambilan keputusan identifikasi penyakit jika salah satu gejala khusus muncul. Pada Tabel 6.1 merupakan contoh aturan dari penyakit tanaman jarak pagar.

Tabel 0.1 Tabel Pohon Keputusan Gejala Terhadap Penyakit

Nomor	Aturan
R-1	IF G01 OR G27 THEN P1
R-2	IF G05 THEN P3
R-3	IF G11 OR G19 OR G20 THEN P4
R-4	IF G12 OR G16 OR G18 THEN P5
R-5	IF G09 OR G26 OR G28 THEN P6
R-6	IF G04 OR G10 THEN P7
R-7	IF G24 THEN P8
R-8	IF G15 OR G21 THEN P9

Berdasarkan aturan tersebut kemudian dilakukan pengujian untuk identifikasi penyakit tanaman jarak pagar berdasarkan parameter terbaik sebelumnya. Hasil pengujian dapat dilihat pada Tabel 6.2

Tabel 0.2 Hasil Pengujian Menggunakan Penambahan Pohon Keputusan

Hasil	Percobaan									
	1	2	3	4	5	6	7	8	9	10
Data latih	87,22	94,74	88,72	84,21	90,23	87,97	88,72	84,21	87,97	87,22
Data Uji	84,85	84,85	78,79	84,85	81,82	90,91	84,85	84,85	93,94	84,85
Rata-rata	86,035	89,795	83,755	84,53	86,025	89,44	86,785	84,53	90,955	86,035

Setelah semua pengujian metode dilakukan, maka dapat dilihat pada Tabel 6.3 hasil perbandingan antar metode.

Tabel 0.3 Perbandingan Hasil Antar Metode

Metode	Akurasi Rata-Rata Terbaik	Akurasi Terbaik
FNN	11,2%	30%

SA-FNN	19,5%	32,5%
MLP	9,1%	12,12%
ELM	60,61%	66,67%
Modifikasi SA dan ELM	62,5%	69,7%
Modifikasi SA dan ELM + Pohon Keputusan	90,955%	94,74%

Berdasarkan hasil perbandingan antar metode yang ditunjukkan pada Tabel 6.3 menunjukkan bahwa metode yang diusulkan yaitu modifikasi SA dan ELM memberikan hasil terbaik dalam identifikasi penyakit jarak pagar. Hal ini dibuktikan dengan memiliki hasil terbaik pada akurasi rata-rata terbaik yaitu sebesar 90,955% dan akurasi maksimal terbaik sebesar 94,74%.

6.2 Validasi Hasil Akurasi Dengan Pakar

Validasi hasil akurasi dengan pakar bertujuan agar hasil akurasi dari identifikasi yang didapat bernilai benar dan dapat digunakan untuk melakukan identifikasi penyakit tanaman jarak pagar. 166 data yang digunakan merupakan hasil observasi langsung berdasarkan fakta lapangan sehingga setiap kasus penyakit yang memiliki masing-masing gejala bernilai benar dan metode yang digunakan dapat digunakan untuk identifikasi selanjutnya terhadap kasus penyakit yang berbeda. Berikut ini adalah hasil validasi dari setiap kasus penyakit dengan melakukan perbandingan hasil dari sistem dapat dilihat di Tabel 6.4 dan selengkapnya di Lampiran.

Tabel 0.4 Hasil Validasi Antara Keputusan Pakar Dengan Sistem

Jenis Data Kasus	Benar	Salah
Data Latih	117 data	16 data
Data Uji	31 data	2 data

BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan pada penelitian yang didapat adalah sebagai berikut.

1. Parameter pada *Extreme Learning Machine* dan modifikasi *Simulated Annealing* yang memberikan akurasi terbaik yaitu *neuron* sebanyak 19, kombinasi temperatur awal 70 dan akhir 40 serta *cooling factor* sebesar 0,65.
2. Penambahan pohon keputusan sangat berpengaruh pada sistem ini. Sistem tanpa pohon keputusan hanya mendapatkan akurasi rata-rata terbaik sebesar 62,5% dan setelah ditambah pohon keputusan mengalami peningkatan sebesar 90,955%.
3. Akurasi rata-rata terbaik pada algoritme modifikasi *Simulated Annealing* dan *Extreme Learning Machine* sebesar 90,955% dengan akurasi terbaik sebesar 94,74% dengan menambahkan pohon keputusan di dalamnya. Hal ini membuktikan metode yang diusulkan memberikan hasil identifikasi yang cukup baik dan menjadi yang terbaik dibanding metode perbandingan yang lainnya berdasarkan Tabel 6.3.

7.2 Saran

Target penelitian ini mendapatkan hasil akurasi terbaik untuk identifikasi penyakit tanaman jarak pagar. Semakin tinggi hasil akurasi maka semakin baik hasil identifikasi dari metode yang digunakan. Untuk penelitian selanjutnya mungkin bisa melakukan representasi *real coded* untuk optimasi nilai bobot pada *neuron* menggunakan *Simulated Annealing* untuk memberikan hasil akurasi yang lebih baik. Penggunaan modifikasi *Extreme Learning Machine* diharapkan juga dapat memberikan hasil yang lebih baik daripada sebelumnya. Penambahan otomatisasi pohon keputusan untuk identifikasi tanaman jarak pagar diperlukan sehingga jika muncul gejala yang menjadi khas dari satu penyakit bisa memberikan keputusan yang lebih baik.

DAFTAR PUSTAKA

- Alfarisy, G. A. F., Sihananto, A. N., Fatyanosa, T. N., Burhan, M. S., & Mahmudy, W. F. (2017). Hybrid Genetic Algorithm and Simulated Annealing for Function Optimization. *Journal of Information Technology and Computer Science*, 1(2), 82–97.
- Bayram, H., & Sahin, R. (2013). A New Simulated Annealing Approach for Travelling Salesman Problem. *Mathematical and Computational Applications*, 18(3), 313–322.
- Bueno-Crespo, A., García-Laencina, P. J., & Sancho-Gómez, J.-L. (2013). Neural architecture design based on extreme learning machine. *Neural Networks*, 48, 19–24. <https://doi.org/10.1016/j.neunet.2013.06.010>
- Chang, S. I., & Aw, C. A. (1996). A neural fuzzy control chart for detecting and classifying process mean shifts. *International Journal of Production Research*, 34(December 2014), 2265. <https://doi.org/10.1080/00207549608905024>
- Dessy, W. M., & Irawan, A. (2012). Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah. *Jurnal Komputer Dan Informatika (KOMPUTA)*, 1(1), 45–51.
- Fajri, D. M. N., Saragih, T. H., Hamdianah, A., Mahmudy, W. F., & Anggodo, Y. P. (2017). Optimized Fuzzy Neural Network for Jatropha Curcas Plant Disease Identification. In *International Conference on Sustainable Information Engineering and Technology (SIET)*, Batu, Indonesia, 25-25 November.
- Fernández, C. M., Fiori, L., Ramos, M. J., Pérez, Á., & Rodríguez, J. F. (2015). Supercritical extraction and fractionation of Jatropha curcas L. oil for biodiesel production. *The Journal of Supercritical Fluids*, 97, 100–106. <https://doi.org/10.1016/j.supflu.2014.11.010>
- Gerlein, E. A., McGinnity, M., Belatreche, A., & Coleman, S. (2016). Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, 54, 193–207. <https://doi.org/10.1016/j.eswa.2016.01.018>
- Hailesilassie, T. (2016). Rule Extraction Algorithm for Deep Neural Networks:, 14(7), 376–381.
- Huang, G. Bin, Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3), 489–501. <https://doi.org/10.1016/j.neucom.2005.12.126>
- Hurtado, C., Luis, J., Fregoso, C., & Hector, J. (2013). Forecasting Mexican inflation using neural networks. In *Electronics, Communications and Computing (CONIELECOMP)* (pp. 32–35). IEEE.
- Jenhani, I., Amor, N. Ben, & Elouedi, Z. (2008). Decision trees as possibilistic classifiers. *International Journal of Approximate Reasoning*, 48(3), 784–807. <https://doi.org/10.1016/j.ijar.2007.12.002>
- Kaewthai, R., Thammaboosadee, S., & Kiattisin, S. (2015). Diabetes dose titration identification model. In *2015 8th Biomedical Engineering International Conference (BMEiCON)* (pp. 1–5). IEEE. <https://doi.org/10.1109/BMEiCON.2015.7399557>

- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- Lin, C.-L., Hsieh, S.-T., & Hu, Y.-J. (2013). Fuzzy Neural Network-Based Influenza Diagnostic System. *2013 First International Symposium on Computing and Networking*, 115(1), 633–635. <https://doi.org/10.1109/CANDAR.2013.115>
- Lin, J. W., Hwang, M. I., & Becker, J. D. (2003). A fuzzy neural network for assessing the risk of fraudulent financial reporting. *Managerial Auditing Journal*, 18(8), 657–665. <https://doi.org/10.1108/02686900310495151>
- Lin, S. W., & Yu, V. F. (2015). A Simulated Annealing Heuristic for the Multiconstraint Team Orienteering Problem with Multiple Time Windows. *Applied Soft Computing*, 37, 632–642. <https://doi.org/10.1016/j.asoc.2015.08.058>
- Lippmann, R. P. (1987). An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, 4(2), 4–22. <https://doi.org/10.1109/MASSP.1987.1165576>
- Machado, A. R., & Pereira, O. L. (2012). Major Diseases of the Biofuel Plant , Physic Nut (*Jatropha curcas*). *Biodiesel - Feedstocks, Production and Applications*, 18. <https://doi.org/10.5772/52336>
- Masaleno, A., & Hasan, M. M. (2012). Skin Diseases Expert System using Dempster-Shafer Theory. *International Journal of Intelligent Systems and Applications*, 5, 38–44.
- Mateo, F., Carrasco, J. J., Sellami, A., Millán-Giraldo, M., Domínguez, M., & Soria-Olivas, E. (2013). Machine learning methods to forecast temperature in buildings. *Expert Systems with Applications*, 40(4), 1061–1068. <https://doi.org/10.1016/j.eswa.2012.08.030>
- Mcculloch, W. S., & Pitts, W. (1943). A logical calculus nervous activity. *Bulletin of Mathematical Biology*, 52(1), 99–115. <https://doi.org/10.1007/BF02478259>
- Nababan, E. B., & Zarlis, M. (2015). Bipolar Dalam Algoritma Backpropagation Pada. *Jurnal Teknovasi*, 02, 103–116.
- Oliveira, H. C. B. De, Vasconcelos, G. C., Alvarenga, G. B., Mesquita, R. V., & Souza, M. M. De. (2007). A Robust Method for the VRPTW with Multi-Start Simulated Annealing and Statistical Analysis. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling*, 7, 198–205.
- Örkcü, H. H. (2013). Subset Selection in Multiple Linear Regression Models: A Hybrid of Genetic and Simulated Annealing Algorithms. *Applied Mathematics and Computation*, 219(23), 11018–11028. <https://doi.org/10.1016/j.amc.2013.05.016>
- Podolak, I., Roman, A., Szykuła, M., & Zieliński, B. (2018). A machine learning approach to synchronization of automata. *Expert Systems with Applications*, 97, 357–371. <https://doi.org/10.1016/j.eswa.2017.12.043>
- Precup, R., Radac, M., Dragos, C., & Preitl, S. (2013). Simulated Annealing Approach to Fuzzy Modeling of Servo Systems, 267–272.
- Rere, L. M. R., Fanany, M. I., & Arymurthy, A. M. (2015). Simulated Annealing Algorithm for Deep



Learning. *Procedia Computer Science*, 72, 137–144. <https://doi.org/10.1016/j.procs.2015.12.114>

Rodrigues, J., Miranda, I., Furquim, L., Gominho, J., Vasconcelos, M., Barradas, G., ... Ferreira-Dias, S. (2015). Storage stability of *Jatropha curcas* L. oil naturally rich in gamma-tocopherol. *Industrial Crops and Products*, 64, 188–193. <https://doi.org/10.1016/j.indcrop.2014.10.048>

Saragih, T. H., Fajri, D. M. N., Hamdianah, A., Mahmudy, W. F., & Anggodo, Y. P. (2017). *Jatropha Curcas* Disease Identification Using Fuzzy Neural Network. In *International Conference on Sustainable Information Engineering and Technology (SIET), Batu, Indonesia, 25-25 November*.

Selim, S. Z., & Alsultan, K. (1991). A Simulated Annealing Algorithm for the Clustering Problem. *Pattern Recognition*, 24(10), 1003–1008. [https://doi.org/10.1016/0031-3203\(91\)90097-0](https://doi.org/10.1016/0031-3203(91)90097-0)

Sousa, T., Soares, T., Morais, H., Castro, R., & Vale, Z. (2016). Simulated Annealing to Handle Energy and Ancillary Services Joint Management Considering Electric Vehicles. *Electric Power Systems Research*, 136, 383–397. <https://doi.org/10.1016/j.eprsr.2016.03.031>

Sutojo, Mulyanto, E., & Suhartono, V. (2011). *Kecerdasan Buatan*. (B. R. W, Ed.). Yogyakarta, Semarang: ANDI, UDINUS.

T.Sutojo, Mulyanto, E., & Suhartono, V. (2011). *Kecerdasan Buatan*. (B. R. W, Ed.). Yogyakarta, Semarang: ANDI, UDINUS.

Tang, J., Deng, C., & Huang, G.-B. (2015). Extreme Learning Machine for Multilayer Perceptron. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13. <https://doi.org/10.1109/TNNLS.2015.2424995>

Utomo, M. C. C., Mahmudy, W. F., & Anam, S. (2017). Determining the Neuron Weights of Fuzzy Neural Networks Using Multi-Populations Particle Swarm Optimization for Rainfall Forecasting. *Journal of Telecommunication, Electronic and Computer Engineering*, 9(2), 37–42.

Xu, S., & Wang, J. (2016). A fast incremental extreme learning machine algorithm for data streams classification. *Expert Systems with Applications*, 65, 332–344. <https://doi.org/10.1016/j.eswa.2016.08.052>

Yadav, A. K., Rahi, O. P., Malik, H., & Azeem, A. (2011). Design Optimization of High-Frequency Power Transformer by Genetic Algorithm and Simulated Annealing. *International Journal of Electrical and Computer Engineering*, 1(2), 102–109.

Yoon, H., Park, C. S., Kim, J. S., & Baek, J. G. (2013). Algorithm learning based neural network integrating feature selection and classification. *Expert Systems with Applications*, 40(1), 231–241. <https://doi.org/10.1016/j.eswa.2012.07.018>

Yulianti, T., & Hidayah, N. (2012). Penyakit Tanaman Jarak Pagar. In *Inovasi Teknologi Jarak Pagar Penghasil Bioenergi Masa Depan* (pp. 217–232). Balai Penelitian Tanaman Pemanis dan Serat.

Yulianti, T., & Hidayah, N. (2015). *Jatropha Curcas Disease*. Malang: Balai Penelitian Tanaman

Pemanis dan Serat.

Zamani, A. M., Amaliah, B., & Munif, A. (2012). Implementation of Genetic Algorithm on Backpropagation Neural Network for Breast Cancer Classification. *Jurnal Teknik POMITS*, 1(1), 1–6.

Zurada, J. (2013). A Fuzzy Neural Approach to Classifying Low Back Disorders Risks. <https://doi.org/10.1109/HICSS.2013.38>



LAMPIRAN A HASIL VALIDASI KEPUTUSAN SISTEM DAN DATA KASUS

No	Gejala	Hasil Pakar	Hasil Sistem
1	G01 = 0,9 G02 = 0,75 G03 = 0,25 G27 = 0,25	Antraknosa	Antraknosa
2	G06 = 0,9 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G29 = 0,9	Busuk Arang	Busuk Arang
3	G06 = 0,9 G07 = 0,5 G15 = 0,25 G22 = 0,5 G23 = 0,5 G25 = 0,25 G30 = 0,5	Layu Fusarium	Layu Fusarium
4	G06 = 0,9 G17 = 0,5 G22 = 0,75 G24 = 0,75 G25 = 0,5	Layu Bakteri	Layu Bakteri
5	G06 = 0,9 G07 = 0,75 G08 = 0,75 G22 = 0,75 G24 = 0,75 G25 = 0,5	Layu Bakteri	Layu Bakteri

6	G30 = 0,5 G08 = 0,5 G09 = 0,9 G26 = 0,75	Embun Tepung	Embun Tepung
7	G06 = 0,9 G07 = 0,5 G15 = 0,25 G21 = 0,75 G22 = 0,5 G23 = 0,5 G25 = 0,25 G30 = 0,5	Layu Fusarium	Layu Fusarium
8	G08 = 0,75 G17 = 0,5 G25 = 0,5	Layu Bakteri	Hawar Bakteri
9	G07 = 0,75 G08 = 0,75 G17 = 0,5	Layu Bakteri	Hawar Bakteri
10	G06 = 0,9 G07 = 0,25 G11 = 0,25 G22 = 0,5 G23 = 0,75 G29 = 0,9	Busuk Arang	Busuk Arang
11	G01 = 0,9 G03 = 0,25 G27 = 0,25	Antraknosa	Antraknosa
12	G06 = 0,9 G20 = 0,5 G22 = 0,5 G23 = 0,75 G30 = 0,75	Busuk Arang	Busuk Arang

13	G08 = 0,5 G09 = 0,9 G28 = 0,9	Embun Tepung	Embun Tepung
14	G06 = 0,9 G07 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9	Busuk Arang	Busuk Arang
15	G06 = 0,9 G07 = 0,25 G08 = 0,25 G11 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
16	G06 = 0,9 G07 = 0,75 G08 = 0,75 G17 = 0,5 G22 = 0,75 G24 = 0,75 G25 = 0,5	Layu Bakteri	Layu Bakteri
17	G02 = 0,9	Bercak Daun Cercospora	Antraknosa
18	G06 = 0,9 G07 = 0,25 G08 = 0,25 G22 = 0,5	Busuk Arang	Busuk Arang
19	G08 = 0,75	Busuk Pucuk	Busuk Pucuk

	G12 = 0,75 G18 = 0,75		
20	G06 = 0,9 G07 = 0,75 G08 = 0,75 G23 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
21	G06 = 0,9 G15 = 0,25 G25 = 0,25	Layu Fusarium	Layu Fusarium
22	G06 = 0,9 G07 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
23	G01 = 0,9	Antraknosa	Antraknosa
24	G06 = 0,9 G19 = 0,25 G22 = 0,5 G23 = 0,75 G29 = 0,9	Busuk Arang	Busuk Arang
25	G06 = 0,9 G07 = 0,25 G08 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang

26	$G06 = 0,9$ $G07 = 0,25$ $G20 = 0,5$ $G22 = 0,5$ $G23 = 0,75$ $G29 = 0,9$	Busuk Arang	Busuk Arang
27	$G06 = 0,9$ $G07 = 0,5$ $G15 = 0,25$ $G21 = 0,75$ $G22 = 0,5$ $G23 = 0,5$ $G25 = 0,25$ $G29 = 0,75$ $G30 = 0,5$	Layu Fusarium	Layu Fusarium
28	$G13 = 0,5$ $G14 = 0,5$ $G26 = 0,75$	Embun Tepung	Embun Tepung
29	$G06 = 0,9$ $G07 = 0,25$ $G08 = 0,25$ $G11 = 0,25$ $G19 = 0,25$ $G22 = 0,5$ $G23 = 0,75$ $G29 = 0,9$ $G30 = 0,75$	Busuk Arang	Busuk Arang
30	$G12 = 0,75$ $G16 = 0,9$ $G17 = 0,5$ $G18 = 0,75$	Busuk Pucuk	Busuk Pucuk
31	$G22 = 0,75$	Layu Bakteri	Hawar Bakteri

	G25 = 0,5		
32	G08 = 0,75 G13 = 0,5 G14 = 0,5 G16 = 0,9 G17 = 0,5 G29 = 0,5	Busuk Pucuk	Busuk Pucuk
33	G06 = 0,9 G07 = 0,75 G08 = 0,75 G17 = 0,5 G22 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Hawar Bakteri
34	G02 = 0,25 G03 = 0,25 G07 = 0,25 G10 = 0,9	Hawar Bakteri	Hawar Bakteri
35	G30 = 0,25	Bercak Daun Altenaria	Bercak Daun Altenaria
36	G08 = 0,5 G09 = 0,9 G14 = 0,5 G26 = 0,75 G28 = 0,9	Embun Tepung	Embun Tepung
37	G06 = 0,9 G22 = 0,5	Busuk Arang	Busuk Arang
38	G06 = 0,9 G07 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5	Busuk Arang	Busuk Arang

	G22 = 0,5 G29 = 0,9		
39	G01 = 0,9 G02 = 0,75 G03 = 0,25	Antraknosa	Antraknosa
40	G06 = 0,9 G07 = 0,5 G15 = 0,25 G21 = 0,75 G22 = 0,5 G23 = 0,5 G29 = 0,75 G30 = 0,5	Layu Fusarium	Layu Fusarium
41	G06 = 0,9 G07 = 0,25 G08 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9	Busuk Arang	Busuk Arang
42	G06 = 0,9 G07 = 0,75 G08 = 0,75 G17 = 0,5 G24 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
43	G01 = 0,9 G08 = 0,75	Antraknosa	Antraknosa
44	G06 = 0,9	Busuk Arang	Busuk Arang

	G07 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75		
45	G03 = 0,25 G04 = 0,25 G10 = 0,9	Hawar Bakteri	Hawar Bakteri
46	G01 = 0,9 G02 = 0,9 G03 = 0,25	Antraknosa	Antraknosa
47	G02 = 0,9	Bercak Daun Cercospora	Antraknosa
48	G06 = 0,9 G15 = 0,25 G21 = 0,75 G22 = 0,5 G23 = 0,5 G25 = 0,25 G29 = 0,75 G30 = 0,5	Layu Fusarium	Layu Fusarium
49	G08 = 0,75 G17 = 0,5 G22 = 0,75 G24 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
50	G08 = 0,5 G09 = 0,9	Embun Tepung	Embun Tepung

	<p>G13 = 0,5 G14 = 0,5 G26 = 0,75 G28 = 0,9</p>		
51	<p>G06 = 0,9 G29 = 0,9</p>	Busuk Arang	Busuk Arang
52	<p>G08 = 0,75 G14 = 0,5 G17 = 0,5</p>	Busuk Pucuk	Busuk Arang
53	<p>G06 = 0,9 G07 = 0,25 G08 = 0,25 G22 = 0,5 G29 = 0,9 G30 = 0,75</p>	Busuk Arang	Busuk Arang
54	<p>G08 = 0,75 G12 = 0,75 G13 = 0,5 G14 = 0,5 G17 = 0,5 G29 = 0,5</p>	Busuk Pucuk	Busuk Pucuk
55	<p>G06 = 0,9 G07 = 0,75 G08 = 0,75 G22 = 0,75 G24 = 0,75</p>	Layu Bakteri	Layu Bakteri
56	<p>G06 = 0,9 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75</p>	Busuk Arang	Busuk Arang

57	$G06 = 0,9$ $G07 = 0,75$ $G08 = 0,75$ $G17 = 0,5$ $G22 = 0,75$ $G24 = 0,75$ $G30 = 0,5$	Layu Bakteri	Layu Bakteri
58	$G13 = 0,5$ $G14 = 0,5$ $G16 = 0,9$ $G18 = 0,75$ $G29 = 0,5$	Busuk Pucuk	Busuk Pucuk
59	$G06 = 0,9$ $G07 = 0,25$ $G08 = 0,25$ $G11 = 0,25$ $G19 = 0,25$ $G20 = 0,5$ $G22 = 0,5$ $G23 = 0,75$ $G29 = 0,9$ $G30 = 0,75$	Busuk Arang	Busuk Arang
60	$G06 = 0,9$ $G07 = 0,75$ $G08 = 0,75$ $G17 = 0,5$ $G22 = 0,75$ $G24 = 0,75$ $G25 = 0,5$ $G30 = 0,5$	Layu Bakteri	Layu Bakteri
61	$G26 = 0,75$ $G28 = 0,9$	Embun Tepung	Embun Tepung

62	G06 = 0,9 G08 = 0,75 G30 = 0,5	Layu Fusarium	Layu Bakteri
63	G08 = 0,5 G09 = 0,9 G13 = 0,5 G14 = 0,5 G28 = 0,9	Embun Tepung	Embun Tepung
64	G06 = 0,9 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9	Busuk Arang	Busuk Arang
65	G06 = 0,9 G08 = 0,75 G22 = 0,75 G24 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
66	G06 = 0,9 G07 = 0,5 G08 = 0,75 G21 = 0,75 G22 = 0,5 G23 = 0,5 G29 = 0,75 G30 = 0,5	Layu Fusarium	Layu Fusarium
67	G06 = 0,9 G07 = 0,25 G08 = 0,25 G11 = 0,25 G19 = 0,25	Busuk Arang	Busuk Arang

	<p>G20 = 0,5 G22 = 0,5 G29 = 0,9 G30 = 0,75</p>		
68	<p>G08 = 0,75 G12 = 0,75 G13 = 0,5 G14 = 0,5 G16 = 0,9 G18 = 0,75</p>	Busuk Pucuk	Busuk Pucuk
69	<p>G06 = 0,9 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75</p>	Busuk Arang	Busuk Arang
70	<p>G07 = 0,75 G08 = 0,75 G17 = 0,5 G22 = 0,75 G24 = 0,75 G25 = 0,5 G30 = 0,5</p>	Layu Bakteri	Layu Bakteri
71	<p>G13 = 0,5 G14 = 0,5 G17 = 0,5</p>	Busuk Pucuk	Busuk Arang
72	<p>G29 = 0,9 G30 = 0,75</p>	Busuk Arang	Busuk Arang
73	<p>G06 = 0,9 G19 = 0,25 G20 = 0,5 G22 = 0,5</p>	Busuk Arang	Busuk Arang



74	<p>G29 = 0,9</p> <p>G06 = 0,9</p> <p>G07 = 0,5</p> <p>G15 = 0,25</p> <p>G22 = 0,5</p> <p>G23 = 0,5</p> <p>G25 = 0,25</p> <p>G29 = 0,75</p> <p>G30 = 0,5</p>	Layu Fusarium	Layu Fusarium
75	<p>G06 = 0,9</p> <p>G07 = 0,25</p> <p>G08 = 0,25</p> <p>G19 = 0,25</p> <p>G20 = 0,5</p> <p>G22 = 0,5</p> <p>G29 = 0,9</p> <p>G30 = 0,75</p>	Busuk Arang	Busuk Arang
76	<p>G08 = 0,5</p> <p>G13 = 0,5</p> <p>G14 = 0,5</p>	Embun Tepung	Embun Tepung
77	<p>G05 = 0,9</p>	Bercak Daun Altenaria	Bercak Daun Altenaria
78	<p>G01 = 0,9</p> <p>G02 = 0,75</p>	Antraknosa	Antraknosa
79	<p>G08 = 0,75</p> <p>G12 = 0,75</p> <p>G14 = 0,5</p> <p>G16 = 0,9</p> <p>G17 = 0,5</p>	Busuk Pucuk	Busuk Pucuk
80	<p>G15 = 0,25</p> <p>G21 = 0,75</p> <p>G22 = 0,5</p>	Layu Fusarium	Layu Fusarium

	G25 = 0,25 G30 = 0,5		
81	G13 = 0,5 G14 = 0,5 G18 = 0,75	Busuk Pucuk	Busuk Pucuk
82	G06 = 0,9 G20 = 0,5 G22 = 0,5 G23 = 0,75 G30 = 0,75	Busuk Arang	Busuk Arang
83	G02 = 0,25 G03 = 0,25 G07 = 0,25	Hawar Bakteri	Hawar Bakteri
84	G06 = 0,9 G07 = 0,5 G21 = 0,75 G29 = 0,75	Layu Fusarium	Layu Fusarium
85	G02 = 0,9	Bercak Daun Cercospora	Antraknosa
86	G07 = 0,25 G08 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
87	G08 = 0,75 G12 = 0,75 G13 = 0,5 G14 = 0,5	Busuk Pucuk	Busuk Pucuk

88	<p>G29 = 0,5</p> <p>G06 = 0,9</p> <p>G08 = 0,25</p> <p>G11 = 0,25</p> <p>G20 = 0,5</p> <p>G22 = 0,5</p> <p>G23 = 0,75</p> <p>G29 = 0,9</p> <p>G30 = 0,75</p>	Busuk Arang	Busuk Arang
89	<p>G06 = 0,9</p> <p>G08 = 0,75</p> <p>G15 = 0,25</p> <p>G21 = 0,75</p> <p>G25 = 0,25</p> <p>G29 = 0,75</p>	Layu Fusarium	Layu Fusarium
90	<p>G02 = 0,25</p> <p>G03 = 0,25</p> <p>G04 = 0,25</p> <p>G07 = 0,25</p>	Hawar Bakteri	Hawar Bakteri
91	<p>G06 = 0,9</p> <p>G17 = 0,5</p> <p>G22 = 0,75</p> <p>G24 = 0,75</p> <p>G25 = 0,5</p> <p>G30 = 0,5</p>	Layu Bakteri	Layu Bakteri
92	<p>G01 = 0,9</p> <p>G03 = 0,25</p>	Antraknosa	Antraknosa
93	<p>G06 = 0,9</p> <p>G07 = 0,75</p> <p>G23 = 0,75</p> <p>G25 = 0,5</p> <p>G29 = 0,9</p>	Busuk Arang	Busuk Arang



94	<p>G30 = 0,75</p> <p>G02 = 0,25</p> <p>G04 = 0,25</p> <p>G10 = 0,9</p>	Hawar Bakteri	Hawar Bakteri
95	<p>G06 = 0,9</p> <p>G07 = 0,25</p> <p>G08 = 0,25</p> <p>G20 = 0,5</p> <p>G22 = 0,5</p> <p>G23 = 0,75</p> <p>G29 = 0,9</p>	Busuk Arang	Busuk Arang
96	<p>G06 = 0,9</p> <p>G08 = 0,25</p> <p>G11 = 0,25</p> <p>G19 = 0,25</p> <p>G20 = 0,5</p> <p>G22 = 0,5</p> <p>G23 = 0,75</p> <p>G29 = 0,9</p> <p>G30 = 0,75</p>	Busuk Arang	Busuk Arang
97	<p>G06 = 0,9</p> <p>G07 = 0,5</p> <p>G15 = 0,25</p> <p>G21 = 0,75</p> <p>G22 = 0,5</p> <p>G25 = 0,25</p> <p>G29 = 0,75</p> <p>G30 = 0,5</p>	Layu Fusarium	Layu Fusarium
98	<p>G13 = 0,5</p> <p>G14 = 0,5</p> <p>G26 = 0,75</p>	Embun Tepung	Embun Tepung
99	<p>G06 = 0,9</p>	Busuk Arang	Busuk Arang

	<p>G08 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75</p>		
100	<p>G06 = 0,9 G07 = 0,75 G17 = 0,5</p>	Layu Bakteri	Hawar Bakteri
101	<p>G06 = 0,9 G08 = 0,75 G21 = 0,75 G29 = 0,75</p>	Layu Fusarium	Layu Fusarium
102	<p>G06 = 0,9 G07 = 0,25 G08 = 0,25 G22 = 0,5 G29 = 0,9</p>	Busuk Arang	Busuk Arang
103	<p>G09 = 0,9 G28 = 0,9</p>	Embun Tepung	Embun Tepung
104	<p>G17 = 0,5 G25 = 0,5 G30 = 0,5</p>	Layu Bakteri	Hawar Bakteri
105	<p>G08 = 0,75 G13 = 0,5 G18 = 0,75</p>	Busuk Pucuk	Busuk Pucuk
106	<p>G08 = 0,75 G13 = 0,5 G14 = 0,5 G29 = 0,9</p>	Busuk Pucuk	Busuk Pucuk

107	G06 = 0,9 G07 = 0,25 G08 = 0,25 G22 = 0,5 G29 = 0,9	Busuk Arang	Busuk Arang
108	G08 = 0,75 G13 = 0,5 G14 = 0,5 G18 = 0,75	Busuk Pucuk	Busuk Pucuk
109	G06 = 0,9 G23 = 0,75 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
110	G17 = 0,5 G22 = 0,75 G25 = 0,5	Layu Bakteri	Hawar Bakteri
111	G06 = 0,9 G07 = 0,25 G08 = 0,25 G23 = 0,75 G30 = 0,75	Busuk Arang	Busuk Arang
112	G05 = 0,9 G30 = 0,25	Bercak Daun Altenaria	Bercak Daun Altenaria
113	G06 = 0,9 G07 = 0,25 G11 = 0,25 G20 = 0,5 G22 = 0,5 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
114	G08 = 0,5 G09 = 0,9	Embun Tepung	Embun Tepung

	G13 = 0,5 G14 = 0,5		
115	G10 = 0,9	Hawar Bakteri	Hawar Bakteri
116	G06 = 0,9 G17 = 0,5 G22 = 0,75 G24 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
117	G06 = 0,9 G19 = 0,25 G20 = 0,5 G22 = 0,5 G29 = 0,9	Busuk Arang	Busuk Arang
118	G05 = 0,9	Bercak Daun Altenaria	Bercak Daun Altenaria
119	G08 = 0,5 G09 = 0,9 G13 = 0,5 G14 = 0,5 G28 = 0,9	Embun Tepung	Embun Tepung
120	G08 = 0,5 G09 = 0,9 G13 = 0,5 G26 = 0,75 G28 = 0,9	Embun Tepung	Embun Tepung
121	G07 = 0,75 G08 = 0,75 G22 = 0,75 G24 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri

122	G06 = 0,9 G07 = 0,5 G22 = 0,5 G29 = 0,75	Layu Fusarium	Layu Bakteri
123	G05 = 0,9 G30 = 0,75	Bercak Daun Altenaria	Bercak Daun Altenaria
124	G08 = 0,75 G12 = 0,75 G13 = 0,5 G14 = 0,5 G16 = 0,9 G17 = 0,5 G18 = 0,75 G29 = 0,5	Busuk Pucuk	Busuk Pucuk
125	G06 = 0,9 G07 = 0,75 G23 = 0,75 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
126	G06 = 0,9 G23 = 0,75 G30 = 0,75	Busuk Arang	Busuk Arang
127	G06 = 0,9 G07 = 0,5 G15 = 0,25 G21 = 0,75 G22 = 0,5 G23 = 0,5 G25 = 0,25 G29 = 0,75	Layu Fusarium	Layu Fusarium
128	G04 = 0,25 G07 = 0,25	Hawar Bakteri	Hawar Bakteri

	G10 = 0,9		
129	G06 = 0,9 G07 = 0,75 G17 = 0,5 G22 = 0,75 G24 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
130	G07 = 0,75 G08 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
131	G08 = 0,5 G09 = 0,9 G26 = 0,75 G28 = 0,9	Embun Tepung	Embun Tepung
132	G08 = 0,75 G13 = 0,5 G14 = 0,5 G17 = 0,5	Busuk Pucuk	Busuk Arang
133	G06 = 0,9 G07 = 0,75 G08 = 0,75 G17 = 0,5 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
134	G05 = 0,9 G30 = 0,25	Bercak Daun Altenaria	Bercak Daun Altenaria
135	G08 = 0,75 G13 = 0,5 G14 = 0,5 G17 = 0,5	Busuk Pucuk	Busuk Pucuk

136	G06 = 0,9 G07 = 0,25 G08 = 0,25 G11 = 0,25 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
137	G02 = 0,9 G03 = 0,25 G10 = 0,9	Hawar Bakteri	Hawar Bakteri
138	G06 = 0,9 G07 = 0,5 G21 = 0,75 G22 = 0,5 G23 = 0,5 G25 = 0,25 G29 = 0,75 G30 = 0,5	Layu Fusarium	Layu Fusarium
139	G04 = 0,25 G07 = 0,75 G10 = 0,9	Hawar Bakteri	Hawar Bakteri
140	G02 = 0,9	Bercak Daun Cercospora	Bercak Daun Cercospora
141	G06 = 0,9 G07 = 0,75 G17 = 0,5 G22 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
142	G04 = 0,25 G10 = 0,9	Hawar Bakteri	Hawar Bakteri

143	G13 = 0,5 G14 = 0,5 G29 = 0,5	Busuk Pucuk	Busuk Pucuk
144	G22 = 0,5 G29 = 0,9 G30 = 0,75	Busuk Arang	Bercak Daun Altenaria
145	G09 = 0,9 G13 = 0,5 G14 = 0,5	Embun Tepung	Embun Tepung
146	G06 = 0,9 G22 = 0,75 G24 = 0,75 G25 = 0,5 G30 = 0,5	Layu Bakteri	Layu Bakteri
147	G06 = 0,9 G07 = 0,25 G08 = 0,25 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75	Busuk Arang	Busuk Arang
148	G08 = 0,75 G14 = 0,5 G16 = 0,9 G17 = 0,5 G18 = 0,75 G29 = 0,5	Busuk Pucuk	Busuk Pucuk
149	G08 = 0,75 G14 = 0,5 G16 = 0,9 G29 = 0,5	Busuk Pucuk	Busuk Pucuk
150	G07 = 0,5	Layu Fusarium	Layu Fusarium



	<p>G08 = 0,75 G15 = 0,25 G21 = 0,75 G22 = 0,5 G23 = 0,5 G25 = 0,25 G29 = 0,75 G30 = 0,5</p>		
151	<p>G08 = 0,5 G13 = 0,5 G14 = 0,5 G26 = 0,75 G28 = 0,9</p>	Embun Tepung	Embun Tepung
152	<p>G06 = 0,9 G19 = 0,25 G20 = 0,5 G22 = 0,5 G29 = 0,9 G30 = 0,75</p>	Busuk Arang	Busuk Arang
153	<p>G06 = 0,9 G07 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9</p>	Busuk Arang	Busuk Arang
154	<p>G06 = 0,9 G07 = 0,5 G15 = 0,25 G21 = 0,75 G23 = 0,5</p>	Layu Fusarium	Layu Fusarium

	G25 = 0,25 G29 = 0,75 G30 = 0,5		
155	G08 = 0,5 G09 = 0,9 G13 = 0,5 G14 = 0,5 G26 = 0,75	Embun Tepung	Embun Tepung
156	G06 = 0,9 G07 = 0,75 G08 = 0,75 G30 = 0,75	Layu Bakteri	Layu Bakteri
157	G06 = 0,9 G15 = 0,25 G21 = 0,75 G22 = 0,5 G25 = 0,25 G29 = 0,75	Layu Fusarium	Layu Fusarium
158	G02 = 0,25 G03 = 0,25 G04 = 0,25 G07 = 0,25 G10 = 0,9	Hawar Bakteri	Hawar Bakteri
159	G02 = 0,9	Bercak Daun Cercospora	Bercak Daun Cercospora
160	G06 = 0,9 G07 = 0,75 G08 = 0,75 G17 = 0,5 G22 = 0,75 G30 = 0,5	Layu Bakteri	Hawar Bakteri
161	G06 = 0,9	Busuk Arang	Busuk Arang

	<p>G07 = 0,25 G08 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5 G23 = 0,75 G29 = 0,9 G30 = 0,75</p>		
162	<p>G13 = 0,5 G14 = 0,5 G16 = 0,9 G17 = 0,5 G18 = 0,75</p>	Busuk Pucuk	Busuk Pucuk
163	<p>G02 = 0,25 G04 = 0,25 G10 = 0,9</p>	Hawar Bakteri	Hawar Bakteri
164	<p>G15 = 0,25 G21 = 0,75 G29 = 0,75 G30 = 0,5</p>	Layu Fusarium	Layu Fusarium
165	<p>G06 = 0,9 G07 = 0,25 G08 = 0,25 G11 = 0,25 G19 = 0,25 G20 = 0,5 G22 = 0,5</p>	Busuk Arang	Busuk Arang
166	<p>G03 = 0,25 G04 = 0,25 G07 = 0,25 G10 = 0,9</p>	Hawar Bakteri	Hawar Bakteri

LAMPIRAN B KODE PROGRAM REPRESENTASI PERMUTASI YANG DIUSULKAN

```
1 public chromosom swap_scramble(int neuron) {
2     double[][] w_bobot = new double[neuron][input[0].length];
3     double[] bias = new double[neuron];
4     chromosom c = null;
5     // w_bobot
6     double[] new_bobot = new double[neuron * input[0].length];
7     int limit = (int) ((int) new_bobot.length * 0.35);
8     int batas1 = (int) ((int) new_bobot.length * 0.15);
9     int batas2 = (int) ((int) new_bobot.length * 0.75);
10    int panjang = new_bobot.length;
11    int r1 = (int) Math.random() * (batas1);
12    int r2 = (int) ((int) Math.random() * (batas2 - (panjang *
13    0.5)) + (panjang * 0.5));
14    int x = r1;
15    int y = r2;
16    int ii = 0;
17    for (int i = 0; i < w_bobot.length; i++) {
18        for (int j = 0; j < w_bobot[i].length; j++) {
19            new_bobot[ii++] = w_bobot[i][j];
20        }
21    }
22    for (int i = 0; i < new_bobot.length; i++) {
23        if (i >= r1 && i <= r1 + limit) {
24            new_bobot[y++] = new_bobot[i];
25        } else if (i >= r2 && i <= r2 + limit) {
26            new_bobot[x++] = new_bobot[i];
27        }
28    }
29    // - scramble
30    double rand = Math.random();
31    // r1
32    if (rand > 0.5) {
33        new_bobot = scramble(new_bobot, r1, r1 + limit);
34    } else { // r2
35        new_bobot = scramble(new_bobot, r2, r2 + limit);
36    }
37    int r = 0;
38    int k = 0;
39    for (int i = 0; i < new_bobot.length; i++) {
40        w_bobot[r][k++] = new_bobot[i];
41        if (k == w_bobot[0].length) {
42            r++;
43            k = 0;
44        }
45    }
46    // bias
47    limit = (int) ((int) bias.length * 0.35);
48    batas1 = (int) ((int) bias.length * 0.15);
49    batas2 = (int) ((int) bias.length * 0.75);
50    panjang = bias.length;
51    r1 = (int) Math.random() * (batas1);
52    r2 = (int) ((int) Math.random() * (batas2 - (panjang * 0.5)) +
53    (panjang * 0.5));
54    x = r1;
55    y = r2;
56    for (int i = 0; i < bias.length; i++) {
57        if (i >= r1 && i <= (r1 + limit)) {
58            bias[y++] = bias[i];
59        } else if (i >= r2 && i <= (r2 + limit)) {
60            bias[x++] = bias[i];
```

```
61     }  
62 }  
63 // - scramble  
64 rand = Math.random();  
65 // r1  
66 if (rand > 0.5) {  
67     bias = scramble(bias, r1, r1 + limit);  
68 } else { // r2  
69     bias = scramble(bias, r2, r2 + limit);  
70 }  
71 c = new chromosom(w_bobot, bias);  
72 return c;  
73 }
```

