

SKRIPSI

APLIKASI JARINGAN SARAF TIRUAN BERBASIS *WEB* UNTUK DIAGNOSA PENYAKIT PADA TANAMAN TOMAT



Disusun Oleh

ROSEVINE HUTABARAT

07. 12. 569

MILIK
PERPUSTAKAAN
ITN MALANG

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011**

3011

UNIVERSITY MICROFILMS
SERIALS ACQUISITION
300 N ZEEB RD
ANN ARBOR MI 48106-1500

APR 13 1988
UNIVERSITY MICROFILMS
ANN ARBOR MI



UNIVERSITY MICROFILMS
SERIALS ACQUISITION
300 N ZEEB RD
ANN ARBOR MI 48106-1500

3011-01

SKRIPSI

**APLIKASI JARINGAN SARAF TIRUAN BERBASIS *WEB* UNTUK
DIAGNOSA PENYAKIT PADA TANAMAN TOMAT**



Disusun Oleh
ROSEVINE HUTABARAT
07. 12. 569

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2011

LEMBAR PERSETUJUAN

APLIKASI JARINGAN SARAF TIRUAN BERBASIS *WEB* UNTUK
DIAGNOSA PENYAKIT PADA TANAMAN TOMAT

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)*

Disusun oleh :

ROSEVINE HUTABARAT

07. 12. 569



Mengetahui,

Ketua Jurusan Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y:1018800189

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y.1018800189

Dr.Eng.Aryuanto, ST, MT
NIP.P. 1030800417

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2011

ABSTRAK

APLIKASI JARINGAN SARAF TIRUAN BERBASIS *WEB* UNTUK DIAGNOSA PENYAKIT PADA TANAMAN TOMAT

Rosevina Hutabarat, NIM 07.12.569

Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT dan Dr. Eng. Aryuanto, ST, MT

Tanaman tomat dapat diserang oleh berbagai macam penyakit, penyakit tersebut dapat diketahui dari gejala-gejala yang ditimbulkan. Akan tetapi, untuk mengetahui secara tepat jenis penyakit yang menyerang tanaman tomat, diperlukan tenaga pakar pertanian dalam waktu yang bersamaan, sedangkan jumlah pakar juga tidak terlalu banyak. Sehingga diperlukan suatu sistem yang memiliki kemampuan seperti seorang pakar.

Sistem dibangun berbasis komputer dengan menggunakan jaringan saraf tiruan backpropagation untuk dapat mendiagnosa jenis penyakit pada tanaman tomat. Sistem juga dibangun berbasis web, agar dapat diakses dengan mudah oleh siapa saja, terutama para petani.

Kata Kunci: jaringan saraf tiruan, backpropagation, web, penyakit tanaman tomat

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, yang telah memberikan berkat-Nya, sehingga penulis dapat menyelesaikan laporan Skripsi ini dengan baik dan lancar.

Laporan Skripsi ini merupakan salah satu persyaratan akademik dalam menyelesaikan program Strata 1 Jurusan Teknik Elektro, Konsentrasi Komputer & Informatika, Institut Teknologi Nasional Malang. Adapun judul laporan Skripsi ini adalah:

JARINGAN SYARAF TIRUAN BERBASIS *WEB* UNTUK DIAGNOSA PENYAKIT PADA TANAMAN TOMAT

Selanjutnya pada kesempatan ini penulis juga menyampaikan rasa terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan tugas akhir, diantaranya :

1. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
2. Bapak Dr. Aryunto Soetedjo, ST, MT selaku Sekertaris Jurusan Teknik Elektro S-1 ITN Malang dan pengusul serta penyedia ruang Skripsi.
3. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Dosen Pembimbing I
4. Bapak Dr. Aryunto Soetedjo, ST, MT selaku Dosen Pembimbing II
5. Bapak Ahmad Faisol, selaku Dosen Wali.
6. Mama, dan kedua kakakku yang telah memberikan dukungan untuk selalu berdoa, berusaha dan nasehat yang telah diberikan sampai saat ini.
7. Seluruh dosen dan pegawai ITN Kampus 2 Malang.

8. Semua teman-teman Laboratorium Jaringan Telekomunikasi, Himpunan Mahasiswa Elektro 2010-2011, Teman-teman Laskar Terasi
9. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis berharap agar buku laporan Skripsi ini dapat memberikan banyak manfaat bagi semua pihak yang membutuhkan, khususnya bagi rekan-rekan mahasiswa. Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, oleh karena itu mohon maaf apabila dalam buku ini terdapat hal-hal yang kurang berkenan dihati para pembaca.

Penulis juga mengharap koreksi, kritik serta saran-saran yang bermanfaat demi kesempurnaan buku Laporan Skripsi ini.

Malang, Agustus 2011

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	6
2.1. Tanaman Tomat.....	6
2.1.1. Definisi	6
2.1.2. Jenis – jenis Penyakit pada Tumbuhan Tomat	7
2.2. Jaringan Saraf Tiruan	14
2.2.1. Definisi	14

2.2.2. Arsitektur Jaringan Saraf Tiruan	14
2.2.3. Proses Pembelajaran	16
2.3. <i>Backpropagation</i> (Propagasi Balik)	17
2.3.1. Arsitektur <i>Backpropagation</i>	18
2.3.2. Fungsi Aktivasi	19
2.3.3. Algoritma <i>Backpropagation</i>	19
2.4. Optimalisasi Arsitektur <i>Backpropagation</i>	24
2.4.1. Pemilihan Bobot dan Bias	24
2.4.2. Jumlah Unit Tersembunyi	25
2.4.3. Lama Iterasi.....	25
2.5. Borland Delphi	26
2.5.1. Pengenalan Delphi	26
2.5.2. Lingkungan Kerja Delphi (IDE)	27
2.5.3. Delphi dan Aplikasi <i>Database</i>	27
2.6 MySQL	27
2.7 PHP	28
2.8 Apache	29
BAB III PERANCANGAN SISTEM.....	30
3.1. Deskripsi Sistem.....	30
3.2. Analisis Sistem.....	30
3.2.1. Analisis Kebutuhan Perangkat Lunak.....	30

3.2.2. Pegolahan Data.....	31
3.3. Perancangan Sistem.....	31
3.4. Desain Sistem	32
3.5. Gambaran Umum Perangkat Lunak	32
3.6. Proses Jaringan Syaraf Tiruan <i>Backpropagation</i>	34
3.6.1. Arsitektur.....	34
3.6.2. Penetapan Masukan (<i>Input</i>)	35
3.6.3. Proses Pelatihan	35
3.6.3.1 Inisialisasi Bobot dan Bias Awal Nyugen Widrow	36
3.6.3.2 <i>Feedforward</i>	37
3.6.3.3 <i>Backpropagation</i>	39
3.6.3.4 <i>Update</i> Bobot	40
3.6.4 Proses Pengujian	41
3.7. Perancangan <i>Form</i> Utama	44
3.8. Struktur Basis Pengetahuan (<i>Knowledge Base</i>)	47
BAB IV IMPLEMENTASI DAN PEMBAHASAN	53
4.1. Implementasi	53
4.1.1. Perangkat Keras	53
4.1.2. Perangkat Lunak.....	53
4.2. Implementasi Program	54
4.2.1. Struktur Data	54
4.2.2. Prosedur Menghitung Faktor Skala (β)	56
4.2.3. Prosedur Inisialisasi Bobot Awal Nguyen Widrow	57

4.2.4. Prosedur Inisialisasi Bobot.....	58
4.2.5. Prosedur <i>Feedforward</i>	59
4.2.6. Prosedur Hitung <i>Error</i>	60
4.2.7. Prosedur <i>Backward</i>	61
4.2.8. Prosedur <i>Update</i> Bobot	63
4.3. Implementasi Program	64
4.4. Penetapan Keluaran (<i>Output</i>)	69
4.5. Hasil dan Pembahasan	69
4.6 Proses Pelatihan	69
4.7 Analisa Hasil	71
BAB V KESIMPULAN DAN SARAN	74
5.1. Kesimpulan.....	74
5.2. Saran.....	75
DAFTAR PUSTAKA	76
LAMPIRAN – LAMPIRAN	

DAFTAR TABEL

BAB III PERANCANGAN SISTEM

Tabel 3.1. Pembentukan Aturan Penyakit Tanaman Tomat.....	47
Tabel 3.2. Pengurutan Gejala untuk Penyakit Tanaman Tomat.....	50

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Tabel 4.1. Hasil Pelatihan Data Penyakit dan Gejala.....	71
Tabel 4.2. Hasil Pengujian pada Data yang telah dilakukan Pembelajaran	72
Tabel 4.3. Data Penyakit yang Belum Diuji Datanya	73

DAFTAR GAMBAR

BAB II TINJAUAN PUSTAKA

Gambar 2.1. Daun Tanaman Tomat yang Terserang Virus Kuning	7
Gambar 2.2. Tanaman Tomat yang Terserang Penyakit Layu Fusarium.....	8
Gambar 2.3. Daun Tanaman Tomat yang Terserang Penyakit Layu Lendir ..	8
Gambar 2.4. Daun dan Buah Tomat yang Terserang Penyakit Hawar Daun..	9
Gambar 2.5. Daun Tanaman Tomat yang Terserang Penyakit Kapang Daun	9
Gambar 2.6. Daun Tanaman Tomat yang Terserang Penyakit Busuk Daun ..	10
Gambar 2.7. Daun yang Terserang Penyakit Bercak Daun Septoria	11
Gambar 2.8. Daun Tanaman Tomat yang Terserang Penyakit Bercak Coklat	11
Gambar 2.9. Buah dari Tanaman Tomat yang Terserang Penyakit Busuk Buah Rhizoctonia	12
Gambar 2.10 Buah dari Tanaman Tomat yang Terserang Penyerang Penyakit Busuk Buah Antraknosa	13
Gambar 2.11 Jaringan Saraf dengan Lapisan Tunggal	15
Gambar 2.12 Jaringan Saraf dengan Banyak Lapisan	16
Gambar 2.13 Jaringan Saraf dengan Lapisan Kompetitif	16
Gambar 2.14 Arsitektur <i>Backpropagation</i>	18

BAB III PERANCANGAN SISTEM

Gambar 3.1. Desain Sistem Jaringan Saraf Tiruan <i>Backpropagation</i> untuk Mendiagnosa Penyakit pada Tanaman Tomat	32
Gambar 3.2. Diagram Alir Sistem Secara Umum	33

Gambar 3.3. Arsitektur Jaringan Saraf Tiruan <i>Backpropagation</i>	34
Gambar 3.4. Diagram Alir Proses Pelatihan Jaringan Saraf Tiruan <i>Backpropagation</i>	36
Gambar 3.5. Diagram Alir Proses Inisialisasi Bobot dan Bias Awal Nguyen Widrow.....	37
Gambar 3.6. Diagram Alir Proses <i>Feedforward</i>	38
Gambar 3.7. Diagram Alir Proses <i>Backpropagation</i>	40
Gambar 3.8. Diagram Alir Proses Perubahan Bobot	41
Gambar 3.9. Diagram Alir Proses Pengujian Jaringan Saraf Tiruan	43
Gambar 3.10. <i>Form</i> Menu untuk Diagnosa Penyakit Tanaman Tomat	44
Gambar 3.11. <i>Form</i> Menu <i>Web</i> untuk Diagnosa Penyakit Tanaman Tomat ...	45
Gambar 3.12. Rancangan <i>Form</i> Pelatihan.....	45
Gambar 3.13. Rancangan <i>Form</i> Diagnosa Penyakit pada Tanaman <i>Web</i>	46
Gambar 3.14. Rancangan <i>Form</i> Hasil Diagnosa Penyakit pada Tampilan <i>Web</i>	47

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Gambar 4.1. <i>Source Code</i> Struktur Data Jaringan Saraf Tiruan	55
Gambar 4.2. <i>Source Code</i> Hitung Faktor Skala.....	57
Gambar 4.3. Prosedur Inisialisasi Bobot Awal Nguyen Widrow	58
Gambar 4.4. Prosedur Inisialisasi Bobot dan Bias awal	59
Gambar 4.5. Prosedur <i>Feedforward</i>	60
Gambar 4.6. Prosedur Hitung <i>Error</i>	61

Gambar 4.7. Prosedur <i>Backward</i>	63
Gambar 4.8. Prosedur Perbaharui Bobot.....	64
Gambar 4.9. Tampilan <i>Form</i> Admin.....	64
Gambar 4.10. Tampilan <i>Form</i> Utama	65
Gambar 4.11. Tampilan <i>Form</i> Utama pada Halaman <i>Web</i>	65
Gambar 4.12. Tampilan <i>Form</i> Pengisian Data Training	66
Gambar 4.13. Tampilan Halaman <i>Web</i> untuk Mendiagnosa Penyakit Tomat	66
Gambar 4.14 Tampilan <i>Form</i> Proses Pelatihan pada Halaman Admin	67
Gambar 4.15. Tampilan Hasil Diagnosa pada Halamn <i>Web</i>	68
Gambar 4.16. Proses Memasukkan Data Penyakit dan Gejala	69
Gambar 4.17. Proses Melatih Data Penyakit dan Gejalanya.....	70
Gambar 4.18 Tampilan Hasil Data Training	70

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pertanian mempunyai arti yang sangat penting bagi kehidupan manusia, selama manusia hidup, selama itu juga pertanian akan tetap ada. Hal itu disebabkan karena makanan merupakan kebutuhan manusia paling pokok, selain udara dan air. Makanan merupakan hasil dari pertanian yang mana setiap tahun kebutuhan akan makanan semakin meningkat karena jumlah populasi manusia yang semakin bertambah. Secara khusus tomat merupakan salah satu hasil dari pertanian yang digunakan sebagai sumber vitamin, yang sangat penting bagi kesehatan manusia. Tanaman tomat sangat rentan terhadap serangan penyakit, seperti virus dan jamur. Dengan begitu pentingnya peranan tomat bagi kebutuhan manusia, maka diperlukan cara untuk dapat mendiagnosa penyakit pada tanaman tomat. Salah satu caranya adalah dengan menggunakan jaringan saraf tiruan.

Penyakit pada tanaman tomat banyak dijumpai di Indonesia. Bagi sebagian besar petani banyak yang tidak mengetahui gejala-gejala penyakit pada tanaman tomat. Kebutuhan informasi yang cepat dan tepat dari seorang ahli atau pakar penyakit tanaman tomat sangatlah dibutuhkan. Hal inilah yang mendorong untuk mengembangkan suatu sistem berbasis komputer sebagai alternatif dalam mendiagnosa penyakit tanaman tomat menggunakan jaringan saraf tiruan. Dengan menggunakan jaringan saraf tiruan ini kita bisa mendiagnosa penyakit tanaman tomat dengan melihat ciri-ciri penyakit yang dapat menggambarkan tanaman tomat terserang penyakit atau tidak.

Menurut Sri Kusumadewi, salah satu cabang dari *Artificial Intelligence* adalah apa yang dikenal dengan Jaringan Saraf Tiruan (*Artificial Neural Network*), hal ini diterangkan dalam bukunya yang berjudul *Artificial Intelligence (Teknik dan Aplikasinya)*. Dengan menggunakan jaringan saraf tiruan maka identifikasi pola data dari sistem diagnosis penyakit tanaman tomat dapat dilakukan dengan metode pendekatan pembelajaran atau pelatihan. Yaitu untuk

menentukan bobot penghubung antar simpul yang optimum. Keunggulan utama jaringan saraf tiruan adalah kemampuan komputasi yang paralel dengan cara belajar dari pola-pola yang diajarkan. Berdasarkan kemampuan belajar yang dimilikinya, maka jaringan saraf tiruan dapat dilatih untuk mempelajari dan menganalisa pola data masa lalu dan berusaha mencari suatu formula yang akan menghubungkan pola data masa lalu dengan keluaran yang diinginkan pada saat ini. Terdapat berbagai macam algoritma pada jaringan saraf tiruan, yaitu *Hebb Rule*, *Bidirectional Associative Memory*, *Learning Vector Quantization*, *Perceptron*, *Kohonen*, dan *Backpropagation*.

Dalam skripsi ini akan diterapkan algoritma pembelajaran *Backpropagation* untuk memprediksi penyakit pada tanaman tomat. *Backpropagation* merupakan suatu metode untuk melakukan pelatihan terhadap lapisan-lapisan kompetitif yang terawasi, dimana diperlukan data pelatihan dalam prosesnya. Jaringan dalam *Backpropagation* bersifat *multilayer*. Jadi, minimal terdapat sebuah *input layer*, *hidden layer*, dan *output layer*.

Penggunaan algoritma *Backpropagation* dalam sistem ini ditujukan untuk meminimalkan error pada keluaran yang dihasilkan jaringan dengan targetnya. Apabila keluaran jaringan memberikan hasil yang salah, maka penimbang (*weight*) dikoreksi agar error dapat diperkecil sehingga tanggapan jaringan saraf tiruan selanjutnya diharapkan dapat mendekati nilai yang benar.

1.2 Rumusan Masalah

Dalam skripsi ini, ruang lingkup permasalahan dibatasi pada bagaimana membuat program aplikasi dalam bentuk perangkat lunak (*software*) yang dipergunakan untuk mengidentifikasi hama dan penyakit tomat beserta solusi penanganannya dalam usaha memberikan masukan bagi petani maupun pihak-pihak yang terkait.

Karena permasalahan hama dan penyakit tomat ini cukup kompleks maka permasalahan pada skripsi ini dapat dirumuskan sebagai berikut :

1. Bagaimana mengidentifikasi hama dan penyakit tomat serta penanggulangannya?

2. Bagaimana membangun perangkat lunak program aplikasi dalam menanggulangi hama dan penyakit tomat?

1.3 Batasan Masalah

Permasalahan yang akan dibahas dalam penulisan ini dibatasi pada:

1. Membahas bagaimana perancangan pembuatan program aplikasi untuk mengidentifikasi hama dan penyakit tomat dengan jaringan saraf tiruan *Backpropagation*.
2. Program aplikasi yang dibuat hanya berfungsi untuk mengidentifikasi hama dan penyakit tomat setelah terjadi serangan hama atau penyakit untuk penanganan hama dan penyakit yang menyerang tomat.
3. Solusi yang dihasilkan hanya bersifat rekomendasi, tidak merupakan solusi yang mutlak.
4. Tidak membahas tentang sistem keamanan *web*.

1.4 Tujuan Penelitian

Tujuan penulisan skripsi ini adalah :

1. Mengidentifikasi hama dan penyakit tomat dengan menggunakan aplikasi jaringan saraf tiruan, beserta solusi penanggulangannya.
2. Membuat program aplikasi sebagai pengganti pakar dengan mensubstitusikan pengetahuan manusia ke dalam bentuk sistem sehingga dapat dipakai orang banyak untuk menangani hama dan penyakit tomat.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari pelaksanaan penelitian ini adalah:

1. Bagi para Ahli Pertanian dan Masyarakat
 - a. Membantu tugas para ahli pertanian untuk memudahkan pengenalan jenis penyakit serta tepat dalam mengambil tindakan.
 - b. Membantu masyarakat, khususnya petani untuk mengetahui penyakit yang menyerang tanaman tomat, sehingga upaya pengobatan penyakit dapat secara cepat dan tepat.

- c. Sistem yang dibuat diharapkan dapat membantu meringankan beban tenaga dan meningkatkan produktifitas kerja ahli pertanian.
2. Bagi Penulis
 - a. Untuk menambah wawasan dan ilmu pengetahuan di luar lingkungan kampus yang berhubungan dengan program studi yang dipilih.
 - b. Untuk menambah pengetahuan sebelum terjun langsung ke dunia kerja dan ke tengah masyarakat untuk mengabdikan ilmu yang dipelajari di bangku kuliah.
 3. Bagi Ilmu Pengetahuan
 - a. Hasil penelitian ini diharapkan dapat memberikan sumbangan bagi pengembangan ilmu di bidang informatika.
 - b. Menggali solusi alternatif dan menambah kepustakaan di bidang ilmu komputer.
 - c. Menambah kepustakaan di bidang ilmu pertanian.

1.6 Metodologi Penelitian

1. Studi literatur dan survei

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan, referensi, dan survei lapangan dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem di mana nantinya akan digunakan sebagai acuan perancangan sistem.

3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka

global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan kedalam sistem.

4. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.7 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

Bab III : Perancangan dan Analisa Sistem

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik *software* maupun *hardware* yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

Bab IV : Pembuatan dan Pengujian Sistem

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Tanaman Tomat

2.1.1 Definisi

Tanaman tomat adalah tumbuhan dari keluarga *Solanaceae*, tumbuhan asli dari Amerika Selatan, Amerika Tengah, Peru, dan Meksiko. Tanaman tomat mulai ditanam setelah kedatangan orang Belanda. Tanaman tomat menghasilkan buah tomat, yang banyak dimanfaatkan untuk kebutuhan konsumsi manusia. Buah tomat bermacam-macam warnanya, seperti merah, hijau, atau kekuningan. Bentuk buah tomat juga bermacam, seperti:

1. *Granola* yang bentuknya bulat dengan pangkal buah mendatar dan mencakup yang biasanya dikenal sebagai tomat buah (karena dapat dimakan langsung).
2. *Gondol* yang biasa dibuat saus dengan bentuk lonjong oval (biasanya yang ditanam di Indonesia adalah kultivar 'Gondol Hijau' dan 'Gondol Putih', dan keturunan dari kultivar impor 'Roma').
3. Sayur adalah tomat dengan buah biasanya padat dan dipakai untuk diolah dalam masakan.
4. Ceri (tomat ranti) yang berukuran kecil dan tersusun berangkai pada tangkai buah yang panjang.

Klasifikasi tanaman tomat adalah sebagai berikut:

Kingdom	: <i>Plantae</i>
Subkingdom	: <i>Tracheobionia</i>
Divisi	: <i>Magnoliophyta</i>
Class	: <i>Magnoliopsida</i>
Subclass	: <i>Asteridae</i>
Order	: <i>Solanales</i>

Family : *Solaneceae*
Genus : *Solanum*
Species : *Solanum lycopersicum*
Binomial name: *Solanum lycopersicum*
Sinonim : *Lycopersicon esculentum*

2.1.2 Jenis-jenis Penyakit pada Tanaman Tomat

Tanaman tomat rentan terhadap serangan penyakit, diantaranya diakibatkan oleh virus dan jamur. Adapun jenis-jenis penyakit yang menyerang tanaman tomat, yaitu:

1. Virus Kuning, yaitu penyakit yang disebabkan oleh virus *Geminivirus* yang ditularkan melalui serangga. Gejala-gejalanya adalah:
 - a) Daun menguning.
 - b) Daun menggulung ke atas.



Gambar 2.1 Daun Tanaman Tomat yang Terserang Virus Kuning

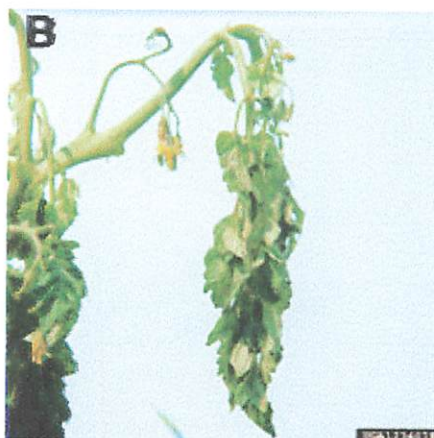
2. Virus Mosaik Keriting, yaitu penyakit yang disebabkan oleh beberapa virus, seperti *Cucumber Mozaic Virus*, *Potato Mozaic Virus*, dan *Tobacco Mozaic Virus*. Penularannya melalui serangga. Gejala-gejalanya adalah:
 - a) Daun berwarna belang (antara hijau tua dan hijau muda).
 - b) Daun berubah bentuk (keriting).
 - c) Daun berpola bercak-bercak seperti mosaik.

3. Layu Fusarium, yaitu penyakit yang disebabkan oleh cendawan *Fusarium oxysporum*. Penyakit menyerang jaringan pembuluh pada tanaman. Gejala-gejalanya adalah:
- Tanaman layu.
 - Daun berubah warna menjadi kuning.
 - Akar akan mengalami diskolorasi (berwarna coklat).



Gambar 2.2 Tanaman Tomat yang Terserang Penyakit Layu Fusarium

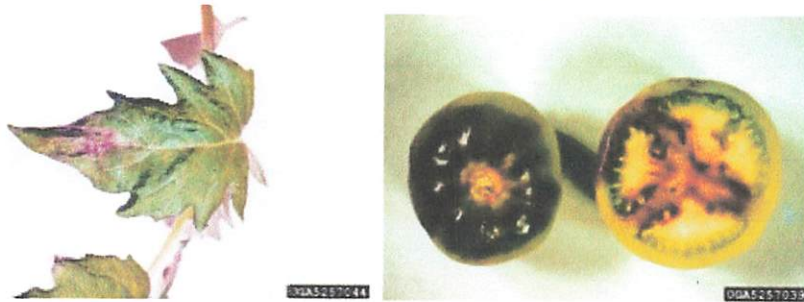
4. Layu Lendir, yaitu penyakit yang disebabkan oleh bakteri *Pseudomonas solanacearum* (E.F. Sm) E.F.Sm. Penyakit menyerang bagian daun dan batang tanaman tomat. Gejala-gejalanya adalah:
- Daun layu.
 - Bagian bawah daun menguning.
 - Daun menggulung ke bawah.



Gambar 2.3 Daun Tanaman Tomat yang Terserang Penyakit Layu Lendir

5. Hawar Daun, yaitu penyakit yang disebabkan oleh cendawan *Phytophthora infestans*. Menyerang bagian daun dan buah dari tanaman. Gejala-gejalanya adalah:

- Bercak berwarna hijau muda sampai kecoklatan pada daun.
- Pada permukaan bawah daun terdapat gejala busuk yaitu berwarna abu-abu keputihan, sampai menjadi bercak besar berwarna coklat.
- Daun berubah warna menjadi coklat.
- Terlihat bercak gelap seperti berminyak pada buah.



Gambar 2.4 Daun dan Buah Tomat yang Terserang Penyakit Hawar Daun

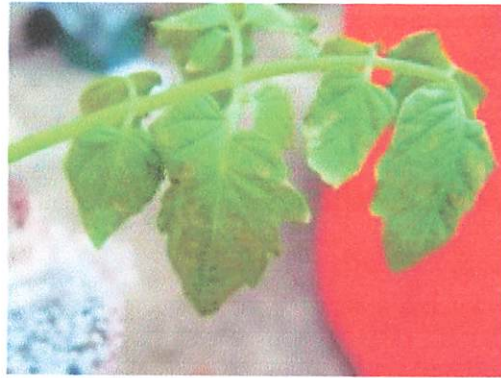
6. Kapang Daun, yaitu penyakit yang disebabkan oleh cendawan *Fulvia fulva* (*Cke*) *Cif*. Penyakit ini menyerang bagian daun. Gejala-gejalanya adalah:

- Terdapat bercak pucat pada permukaan daun.
- Pada bagian bawah daun terdapat spora-spora berwarna kelabu, coklat, hingga kekuning-kuningan.
- Menyerang pada daun-daun bagian bawah, hingga ke seluruh tanaman.



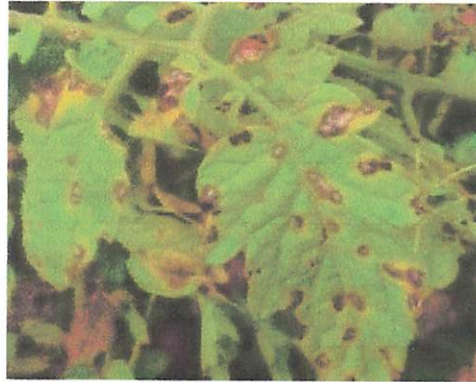
Gambar 2.5 Daun Tanaman Tomat yang Terserang Penyakit Kapang Daun

7. Busuk Daun, yaitu penyakit yang disebabkan oleh cendawan *Phytophthora infestans* (Mont.) de bary. Penyakit ini menyerang bagian daun dan juga buah. Gejala-gejalanya adalah:
- Bercak coklat kehitaman pada daun hingga tangkai daun.
 - Pada bagian bawah daun terdapat jamur berwarna putih keunguan seperti beludru.
 - Bercak berair pada pangkal buah yang berwarna hijau hingga kecoklatan.
 - Pada buah terdapat bercak-bercak berbentuk cincin yang berwarna coklat tua.
 - Bercak pada buah agak keras dan berkerut.



Gambar 2.6 Daun Tanaman Tomat yang Terserang Penyakit Busuk Daun

8. Bercak Daun Septoria, yaitu penyakit yang disebabkan oleh cendawan *Septoria lycopersici* sp. Penyakit menyerang bagian daun, dan tanaman tomat yang masih muda maupun sudah tua. Gejala-gejalanya adalah:
- Pada permukaan bawah daun terdapat bercak bulat kecil berair (bercak berwarna coklat muda).
 - Daun menggulung, kering.



Gambar 2.7 Daun yang Terserang Penyakit Bercak Daun Septoria

9. Bercak Coklat, yaitu penyakit yang disebabkan oleh cendawan *Alternaria solani* Sor. Penyakit ini menyerang bagian daun dan juga buah. Gejala-gejalanya adalah:
- a) Daun terdapat bercak berwarna coklat sampai kehitaman.
 - b) Bercak pada daun akan membesar, hingga daun berubah warna menjadi kuning, layu dan mati.
 - c) Menyebabkan keguguran pada bunga.
 - d) Buah tomat yang masih muda atau sudah matang menjadi busuk, berwarna hitam, dan cekung.
 - e) Menyerang pangkal buah (ujung tangkai), yang menyebabkan perubahan warna menjadi coklat, dan tertutup spora hitam seperti beludru.



Gambar 2.8 Daun Tanaman Tomat yang Terserang Penyakit Bercak Coklat

10. Busuk Buah Rhizoctonia, yaitu penyakit yang disebabkan cendawan *Thanatephorus cucumeris* (Frank) Donk. Penyakit ini terutama menyerang pada bagian buah, dan bagian tumbuhan lainnya seperti batang tumbuhan. Gejala-gejalanya adalah:

- a) Bercak cekung kecil berwarna coklat pada buah.
- b) Batang tanaman terpilin.
- c) Pembusukan dengan warna coklat kemerahan pada pangkal batang.



Gambar 2.9 Buah dari Tanaman Tomat yang Terserang Penyakit Busuk Buah Rhizoctonia

11. Busuk Buah Antraknosa, yaitu penyakit yang disebabkan oleh cendawan *Colletotrichum coccodes* (Wallr.) Hughes. Gejala-gejalanya adalah:

- a) Bercak kecil berwarna coklat, bulat, dan berair pada buah.
- b) Bercak ungu dekat tangkai buah.
- c) Daun menjadi layu.
- d) Akar akan mengalami diskolorasi (berwarna coklat).



Gambar 2.10 Buah dari Tanaman Tomat yang Terserang Penyakit Busuk Buah Antraknosa

12. Bercak Bakteri, yaitu penyakit yang disebabkan oleh bakteri yang merugikan, yang menyerang bagian batang dan daun tanaman tomat.

Gejala-gejalanya adalah:

- a) Bercak kecil berair pada batang dan daun.
- b) Daun menggulung (keriting) ke bawah.
- c) Daun kering.
- d) Bercak berair, lalu berubah menjadi bercak seperti gabus pada buah.
- e) Terdapat luka seperti kerak keabu-abuan pada batang.

2.2 Jaringan Saraf Tiruan

2.2.1 Definisi

Jaringan Saraf Tiruan (JST) merupakan representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan saraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran.

Jaringan saraf tiruan disusun sama seperti jaringan pada saraf biologis, yaitu:

1. Pengolahan informasi terjadi pada elemen-elemen pemrosesan (*neuron-neuron*).
2. Sinyal antara dua buah *neuron* diteruskan melalui perantara koneksi.
3. Setiap perantara koneksi memiliki bobot terasosiasi.
4. Setiap *neuron* menerapkan sebuah fungsi aktivasi terhadap masukan jaringan (jumlah sinyal input berbobot). Tujuannya adalah untuk menentukan sinyal keluaran. Fungsi aktivasi yang digunakan biasanya adalah fungsi yang nonlinier.

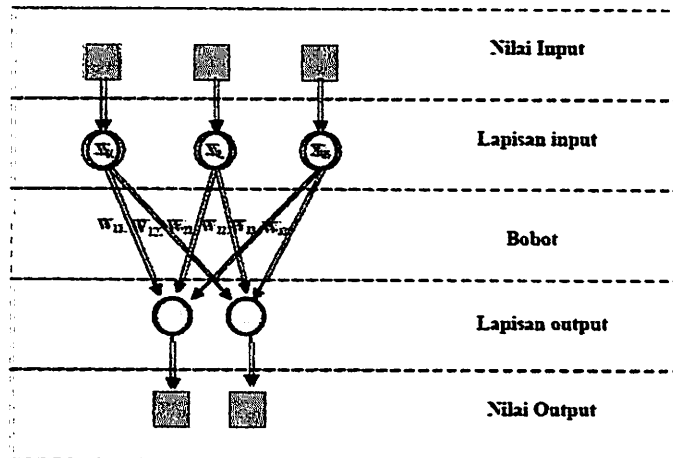
Adapun cara belajar jaringan saraf tiruan adalah, ke dalam jaringan saraf tiruan dimasukkan informasi yang sebelumnya telah diketahui hasil keluarannya. Masukan dilakukan lewat *node* (simpul) atau unit masukan. Bobot-bobot antar koneksi dalam suatu arsitektur diberi nilai awal, lalu jaringan saraf tiruan dijalankan. Bobot-bobot akan digunakan untuk proses belajar dan mengingat informasi. Pengaturan bobot dilakukan secara terus-menerus dengan menggunakan aturan tertentu sampai diperoleh keluaran yang diharapkan.

2.2.2 Arsitektur Jaringan Saraf Tiruan

Hubungan antar *neuron* dalam jaringan saraf mengikuti pola tertentu pada arsitektur jaringan sarafnya. Arsitektur jaringan saraf dibagi menjadi tiga, yaitu:

1. Jaringan saraf dengan lapisan tunggal (*single layer net*)

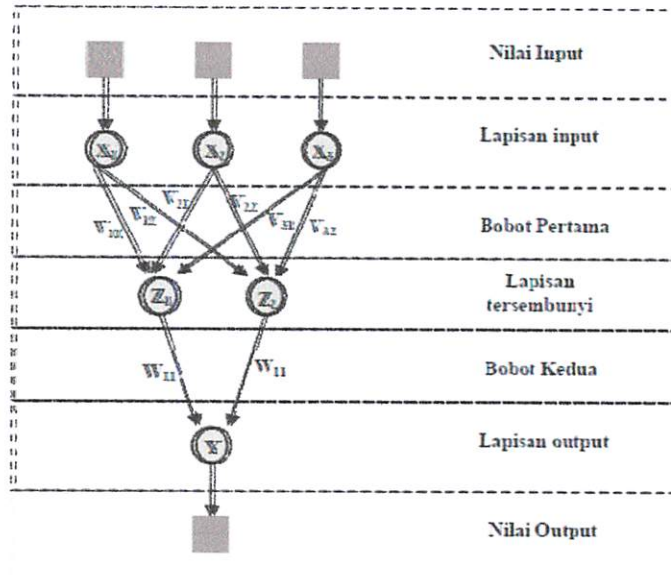
Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima *input*, kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi. Dalam gambar 2.11 menunjukkan arsitektur jaringan saraf dengan lapisan tunggal.



Gambar 2.11 Jaringan Saraf dengan Lapisan Tunggal

2. Jaringan saraf dengan banyak lapisan (*multilayer net*)

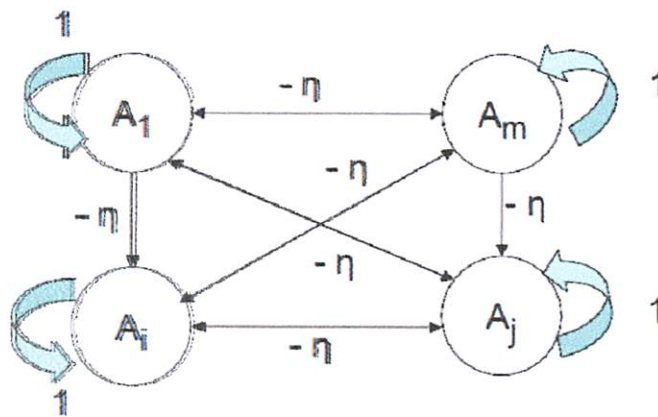
Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output* (memiliki satu atau lebih lapisan tersembunyi). Jaringan saraf dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit, dengan pembelajaran yang lebih rumit pula. Dalam gambar 2.12 menunjukkan arsitektur jaringan saraf dengan banyak lapisan.



Gambar 2.12 Jaringan Saraf dengan Banyak Lapisan

3. Jaringan saraf dengan lapisan kompetitif (*competitive layer net*)

Hubungan antar *neuron* pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Dalam gambar 2.13 menunjukkan contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot $-\eta$.



Gambar 2.13 Jaringan Saraf dengan Lapisan Kompetitif

2.2.3 Proses Pembelajaran

Jaringan saraf tiruan akan mencoba untuk mensimulasikan kemampuan otak manusia untuk belajar. Perubahan yang terjadi selama proses pembelajaran adalah perubahan nilai bobot. Nilai bobot akan bertambah, jika informasi yang diberikan oleh *neuron* tersampaikan, sebaliknya jika informasi tidak disampaikan oleh suatu *neuron* ke *neuron* lain, maka nilai bobot yang menghubungkan

keduanya akan dikurangi. Pembelajaran dilakukan pada masukan yang berbeda, maka nilai bobot akan diubah secara dinamis hingga mencapai suatu nilai yang seimbang. Terdapat dua metode pembelajaran, yaitu:

1. Pembelajaran terawasi (*supervised learning*)

Metode pembelajaran pada jaringan saraf tiruan disebut terawasi jika keluaran yang diharapkan telah diketahui sebelumnya.

2. Pembelajaran tak terawasi (*unsupervised learning*)

Metode ini tidak memerlukan target keluaran. Selain itu tidak dapat ditentukan hasil yang diharapkan selama proses pembelajaran^[1].

Dalam proses pembelajaran, *learning rate* adalah sebuah variabel yang nilainya diantara 0 sampai 1, yang berguna untuk menentukan laju pembelajaran. Sedangkan *max epoch* adalah batas maksimum perulangan untuk setiap pola pelatihan.

2.3 Backpropagation (Propagasi balik)

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyi. Algoritma *Backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*feedforward*) harus dikerjakan terlebih dahulu. Terdapat parameter-parameter penentu dalam pembelajaran Jaringan Saraf Tiruan, yaitu:

1. *Maximum epoch*

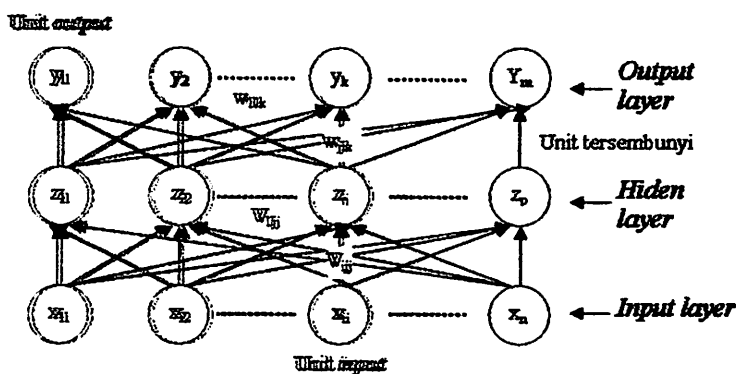
Menentukan berapa iterasi pembelajaran yang akan dilakukan pada Jaringan Saraf Tiruan. Semakin besar *maximum epoch*, maka tingkat kesalahan Jaringan Saraf Tiruan akan semakin menurun. Namun, penentuan *maximum epoch* yang terlalu besar akan menyebabkan Jaringan Saraf Tiruan terlalu mengikuti pola data pelatihan dan meningkatkan kesalahan yang mungkin terjadi ketika Jaringan Saraf Tiruan diberikan data masukan dari permasalahan sebenarnya. Hal ini disebut *overfitting*.

2. Laju pembelajaran (α)

Laju pembelajaran menunjukkan seberapa cepat Jaringan Saraf Tiruan akan menyesuaikan diri dengan data pelatihan yang diterimanya. *Learning rate* Merupakan salah satu parameter pelatihan untuk menghitung nilai koreksi bobot pada waktu proses pelatihan. Nilai α ini berada pada range 0 (nol) sampai 1 (satu). Semakin besar nilai dari *learning rate*, maka proses pelatihan akan berjalan semakin cepat. Namun, apabila nilai *learning rate* terlalu besar, pada umumnya proses pelatihan dapat melampaui keadaan optimal, yaitu pada saat dicapai nilai *error* yang paling minimal. Dengan kata lain, *learning rate* mempengaruhi ketelitian jaringan suatu sistem. Semakin besar *learning rate*, maka ketelitian jaringan akan semakin berkurang. Sebaliknya, apabila *learning rate*-nya semakin kecil, maka ketelitian jaringan akan semakin besar atau bertambah dengan konsekuensi proses pelatihan akan memakan waktu yang lama.

2.3.1 Arsitektur *Backpropagation*

Jaringan ini memiliki satu lapis masukan ditambah dengan sebuah bias, satu atau lebih lapisan tersembunyi (*hidden layer*) ditambah dengan sebuah bias, dan satu lapisan keluaran. Setiap lapis memiliki *neuron-neuron* (unit-unit). Di antara *neuron* pada lapis berikutnya dihubungkan dengan model koneksi yang memiliki bobot-bobot (*weights*).



Gambar 2.14 Arsitektur *Backpropagation*

Berdasarkan gambar 2.14, v_{ij} merupakan bobot dari *neuron input layer* x_i ke *neuron hidden layer* z_i , dimana v_{ij} merupakan bobot yang menghubungkan bias dari *neuron input layer* ke *hidden layer*. Sedangkan w_{jk} merupakan bobot dari *neuron hidden layer* z_j ke *neuron output layer* y_k , dimana w_{jk} merupakan bobot dari bias di *neuron hidden layer* ke *neuron output layer*.

2.3.2 Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering digunakan dalam Jaringan Saraf Tiruan. Penggunaan fungsi aktivasi dalam algoritma *Backpropagation* harus berupa fungsi yang dapat diturunkan. Biasanya menggunakan fungsi aktivasi Sigmoid. Dalam metode *Backpropagation*, fungsi aktivasi yang sering digunakan adalah fungsi Sigmoid Biner (*Binary Sigmoid*), yang mempunyai jangkauan nilai antara 0 (nol) sampai 1 (satu). Nilai dari keluaran dan masukan ada diantara 0 (nol) sampai 1 (satu). Persamaan fungsinya adalah :

$$f(x) = \frac{1}{1+e^{-ax}} \quad (2.1)$$

fungsi turunannya adalah :

$$f(x) = \alpha f(x)[1 - f(x)] \quad (2.2)$$

Learning rate (α) merupakan laju pembelajaran dan e adalah bilangan euler.

2.3.3 Algoritma *Backpropagation*

Pelatihan sebuah jaringan *backpropagation* terdiri dari tiga langkah, yaitu pelatihan pola masukan secara *feedforward*, *backpropagation* dari kumpulan kesalahan, dan penyesuaian bobot. Pelatihan dilakukan berulang-ulang dan berhenti jika telah mencapai batas iterasi maksimum yang ditentukan dan nilai *error* kurang dari *Mean Square Error* (MSE). Ketepatan algoritma *Backpropagation* ditentukan dengan *Mean Square Error* (MSE). Semakin kecil nilai MSE maka dapat dianggap bahwa arsitektur jaringan semakin baik. MSE dihitung dengan persamaan 2.3

$$\text{MSE} = \frac{1}{2} \sum_p (t_p - y_p)^2 \quad (2.3)$$

dimana p adalah jumlah *neuron* (unit) keluaran, t adalah target, dan y adalah keluaran.

Algoritma pelatihan *backpropagation* adalah sebagai berikut :

1. Bobot awal ditentukan secara acak dengan nilai nilai sekecil mungkin, misalnya $[-0,01,0,01]$.
2. Inisialisasi *error target*, jumlah *hidden layer*, *learning rate*, dan jumlah maksimum epoch.
3. Selama (epoch < maksimum epoch) dan (MSE < target error), maka :

a. Lakukan *feedforward* untuk setiap pasangan pelatihan :

- 1) Tiap-tiap unit masukan (x_i , $i=1,2,\dots,n$) menerima sinyal dan menyalurkan sinyal ini ke semua unit lapisan tersembunyi.
- 2) Tiap-tiap unit tersembunyi (z_j , $j=1,2,\dots,p$) menjumlahkan perkalian nilai masukan dan bobot sinyal masukan dengan Persamaan 2.4

$$z_{in_j} = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (2.4)$$

Gunakan fungsi aktivasi untuk menghitung *outputnya* yaitu :

$$z_j = f(z_{in_j}) \quad (2.5)$$

Lalu kirimkan sinyal tersebut ke semua unit di lapisan atasnya.

- 3) Tiap-tiap unit keluaran (y_k ; $k=1,\dots,m$) akan menjumlahkan bobot sinyal keluaran dengan Persamaan 2.6.

$$y_{in_k} = v_{ok} + \sum_l z_l w_{lk} \quad (2.6)$$

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran.

$$y_k = f(y_{in_k}) \quad (2.7)$$

b. Lakukan propagasi balik untuk masing-masing pasangan pelatihan :

- 1) Tiap-tiap unit keluaran (y_k ; $k=1, \dots, m$) menerima target yang berkesesuaian dengan pola pelatihan, hitung informasi kesalahan.

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (2.8)$$

Kemudian hitung koreksi bobot, digunakan untuk memperbaharui w_{jk} (Persamaan 2.9).

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.9)$$

hitung koreksi bias untuk memperbaiki bobot bias (w_{ok}).

$$\Delta w_{ok} = \alpha \delta_k \quad (2.10)$$

Kemudian kirimkan nilai δ_k ke seluruh unit yang berada pada lapisan di bawahnya (*backward*).

- 2) Tiap-tiap unit tersembunyi (Z_j ; $j=1, \dots, p$) menjumlahkan *input delta* dari unit lapisan atasnya.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.11)$$

Kalikan fungsi ini dengan turunan fungsi aktivasi untuk menghitung informasi kesalahan

$$\delta_j = \delta_{in_j} f'(y_{in_j}) \quad (2.12)$$

Hitung koreksi bobot untuk memperbaiki v_{ij} .

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.13)$$

Hitung koreksi bobot bias untuk memperbaiki v_{oj} .

$$\Delta v_{oj} = \alpha \delta_j \quad (2.14)$$

c. Perbaiki bobot dan bias

Tiap-tiap unit keluaran (y_k ; $k=1, \dots, m$) memperbaiki bobot dan bias ($j=0, \dots, p$).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.15)$$

Tiap-tiap unit tersembunyi ($Z ; 1, \dots, p$) memperbaiki bobot dan bias ($i ; 0, \dots, n$).

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.16)$$

1) Jika iterasi mencapai maksimum, maka berhenti.

Setelah proses pelatihan, tahap pengujian jaringan saraf *backpropagation* diaplikasikan dengan hanya menggunakan tahap perambatan maju (*feedforward*) dari algoritma pelatihan. Prosedur pelatihannya adalah sebagai berikut :

- a) Inisialisasi bobot yang diperoleh dari proses pelatihan jaringan.
- b) Tiap-tiap unit masukan ($x_i ; i=1,2,\dots,n$) menerima sinyal dan menyalurkan ke semua unit lapisan tersembunyi.
- c) Tiap-tiap unit tersembunyi ($z_j ; j=1,\dots,p$) menjumlahkan perkalian nilai masukan dan bobot sinyal masukan.

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.17)$$

Gunakan fungsi aktivasi untuk menghitung keluaran, yaitu:

$$z_j = f(z_{in_j}) \quad (2.18)$$

- d) Tiap-tiap unit keluaran ($y_k ; k=1,\dots,m$) akan menjumlahkan bobot sinyal keluaran.

$$y_{in_k} = v_{0k} + \sum_i z_k w_{jk} \quad (2.19)$$

Gunakan fungsi aktivasi untuk menghitung nilai keluaran.

$$y_k = f(y_{in_k}) \quad (2.20)$$

Jika $y_k \geq 0,5$ maka $y_k=1$, *else* $y_k=0$.

Keterangan dari simbol-simbol :

- t = keluaran vector target, $t=(t_1, \dots, x_k, \dots, x_m)$
- δ_k = informasi tentang kesalahan pada unit y_k yang disebarkan kembali ke unit tersembunyi
- δ_j = informasi tentang kesalahan dari lapisan keluaran ke unit tersembunyi z_j
- α = *learning rate*
- x_i = unit masukan i
- v_{0j} = bobot awal bias pada lapisan tersembunyi j
- v_{ij} = bobot menuju *hidden*
- z_j = unit tersembunyi j
- z_in_j = masukan ke jaringan z_j
- w_{0k} = bobot awal bias pada menuju keluaran
- w_{jk} = bobot menuju keluaran
- y_k = unit keluaran i
- y_in_k = masukan jaringan ke y_k
- e = error tiap pasangan

2.4 Optimalitas Arsitektur *Backpropagation*

2.4.1 Pemilihan Bobot dan Bias

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Bobot yang menghasilkan nilai turunan fungsi aktivasi yang terlalu kecil sebaiknya dihindari, demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil.

Nguyen dan Widrow (1990) mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat.

Bila diketahui : n = jumlah unit masukan

p = jumlah unit tersembunyi

β = faktor skala = $0.7^n \sqrt{p}$

Algoritma penentuan bobot dan bias :

1. Inisialisasi semua bobot ($v_{ij}(\text{lama})$) dengan bilangan acak dalam interval $[-0,05,0,05]$
2. Hitung

$$\|v_j\| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2} \quad (2.21)$$

3. Bobot yang dipakai sebagai inisialisasi

$$v_{ij} = \frac{\beta v_{ij}(\text{lama})}{\|v_j\|} \quad (2.22)$$

4. Bias yang dipakai sebagai inisialisasi v_{0j} = bilangan acak antara $-\beta$ dan β .

2.4.2 Jumlah Unit Tersembunyi

Menentukan jumlah lapisan pada *hidden layer* dan menentukan jumlah unit tiap lapisannya adalah sangat sulit. Secara teoritis ditunjukkan bahwa jaringan dengan sebuah *hidden layer* sudah cukup bagi *backpropagation* untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Jumlah unit tersembunyi dapat ditentukan dengan melakukan beberapa percobaan (*trial and error*). Tetapi penambahan jumlah *hidden layer* kadangkala membuat pelatihan menjadi lebih mudah.

2.4.3 Lama Iterasi

Tujuan utama penggunaan *backpropagation* adalah untuk mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Tapi hal tersebut tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat.

Umumnya data dibagi menjadi dua bagian saling asing, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai untuk pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan, kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahannya meningkat, pelatihan tidak ada gunanya untuk diteruskan lagi.

2.5 Borland Delphi

2.5.1 Pengenalan Delphi

Borland Delphi 7 adalah bahasa pemrograman visual yang dirilis oleh Borland Internasional, dan merupakan pengembangan dari bahasa pemrograman Pascal dan Borland Delphi sebelumnya. Delphi 7 mempunyai banyak tambahan fitur baru dibandingkan versi sebelumnya, sehingga Delphi 7 telah menjadi lingkungan pengembangan yang cukup kompleks.

Dalam penciptaan bahasa Borland Delphi 7 ini, Borland Internasional bertujuan menjadikannya sebagai solusi *e-business*, yang salah satu ujung tombaknya adalah *Web-service*, disisi lain untuk aplikasi berdatabase Borland 7 sudah cukup proposional dibandingkan versi sebelumnya.

Borland Delphi adalah program untuk membuat aplikasi berbasis Microsoft Windows secara cepat dan mudah. Borland Delphi menyediakan *tool* untuk membuat aplikasi sederhana sampai aplikasi yang kompleks, baik untuk keperluan pribadi maupun untuk keperluan instansi/perusahaan dengan sistem yang lebih besar. Adapun kemampuan lain Borland Delphi 7 di antaranya :

1. Memiliki sarana pengembangan yang bersifat grafis (Visual).
2. Berorientasi objek (*object oriented*).
3. Dapat bekerja di dalam sistem operasi Windows.
4. Dapat menghasilkan program aplikasi berbasis Windows.
5. Mampu memanfaatkan program aplikasi berbasis Windows, seperti grafis, multimedia, internet, multitasking, dan sebagainya.

Borland Delphi berbasiskan *Object Oriented Programming* (OOP) dan dikembangkan dengan basis Visual yang berarti menggunakan sarana grafis untuk mengembangkannya. Borland Delphi berorientasi pada objek-objek yang dipisah-pisah, sehingga disebut pemrograman berorientasi objek. Borland Delphi juga bersifat modular programming, karena kode-kode program letaknya tersebar di dalam modul-modul (objek-objek) yang terpisah.

2.5.2 Lingkungan Kerja Delphi (IDE)

Lingkungan pengembangan terpadu atau *Intergrated Development Environment* (IDE) dalam program Delphi terbagi menjadi enam bagian utama, yaitu, Main Window, Toolbar, Component Palette, Form Designer, Code Editor, dan Object Inspector. IDE merupakan sebuah lingkungan di mana semua tombol perintah yang diperlukan untuk mendesain aplikasi, menjalankan, dan menguji sebuah aplikasi disajikan dengan baik untuk memudahkan pengembangan program. Sebagai sarana interaktif, Delphi menyediakan bermacam-macam komponen interface aplikasi yang berupa tombol, menu dropdown maupun pop-up, kotak teks, kotak gambar, radio button, checkbox, scrollbar, listbox, combobox, dan panel. Dalam pemrograman Delphi terdapat beberapa elemen pemrograman, antara lain: Identifier, tipe data, dan scope.

2.5.3 Delphi dan Aplikasi *Database*

Delphi menyediakan object yang sangat kuat, canggih, dan lengkap, sehingga memudahkan pemrogram dalam merancang, membuat, dan menyelesaikan aplikasi *database* yang diinginkan. Selain itu juga dapat menangani data dalam berbagai format *database*, misalnya format MS-Access, Sybase, Oracle, FoxPro, Informix, DB2, dan lain sebagainya. Format *database* yang dianggap asli dari Delphi adalah Paradox dan dBase.

2.6 MySQL

MySQL merupakan *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi *General Public Lisence* (GPL). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan sebagai produk yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu *Structure Query Language* (SQL).

SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL adalah salah satu jenis *database* yang sangat terkenal, karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses-nya. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* ataupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan dengan *database server* yang lainnya dalam *query* data.

2.7 PHP

PHP adalah bahasa pemrograman yang memungkinkan para *web developer* untuk membuat aplikasi *web* yang dinamis dan cepat. PHP merupakan singkatan dari “Hypertext Preprocessor”. PHP ditulis dan diperkenalkan pertama kali sekitar tahun 1994 oleh Rasmus Lerdorf melalui situsnya untuk mengetahui siapa saja yang telah mengakses ringkasan *online*-nya.

Untuk dapat berjalan, PHP membutuhkan *web server*, yang bertugas untuk memproses file-file php dan mengirimkan hasil pemrosesan untuk ditampilkan di *browser client*. Oleh karena itu, PHP termasuk *server-side scripting* (*script* yang diproses di sisi *server*). *Web server* sendiri adalah software yang diinstall pada computer local ataupun computer lain yang berada di jaringan intranet/internet yang berfungsi untuk melayani permintaan-permintaan *web* dari *client*.

Karakteristik yang paling unggul dan paling kuat dalam PHP adalah lapisan integrasi *database* (*database integration layer*). *Database* yang didukung PHP antara lain : Oracle, Adabas-D, Sybase, FilePro, mSQL, MySQL, Informix, Solid, dBase, ODBC, Unix dbm, dan PostgreSQL. Contoh script PHP sederhana terdapat pada Listing 1.

Listing 1 : Contoh Script PHP

```
<html>
<head>
<title>coba PHP </title>
</head>
<body>
<?
```

Echo "Hallo dari PHP";

?>

</body>

</html>

2.8 Apache

Server HTTP Apache atau *Server Web/WWW Apache* adalah *web server* yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas *web/www* ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.

BAB III

PERANCANGAN SISTEM

3.1 Deskripsi Sistem

Deskripsi sistem merupakan gambaran tentang sistem yang akan dibuat. Langkah-langkah untuk mendiagnosa penyakit pada tanaman tomat melalui sistem diwujudkan dengan adanya dialog antara *user* (petani dan masyarakat) dengan sistem, sehingga dihasilkan keluaran berupa kesimpulan tentang penyakit yang menyerang tanaman tomat serta solusi untuk mengatasi penyakit pada tanaman tomat tersebut.

3.2 Analisis Sistem

3.2.1 Analisa Kebutuhan Perangkat Lunak

Beberapa spesifikasi kebutuhan yang dibutuhkan dalam rancangan perangkat lunak ini antara lain :

1. Dapat menerima obyek dari pakar berupa nama penyakit, gejala penyakit, serta solusi sebagai masukan untuk pelatihan.
2. Dapat melatih sistem untuk mengenali pola-pola penyakit sesuai dengan aturan Jaringan Saraf Tiruan *Backpropagation*.
3. Dapat menyimpan bobot-bobot hasil pelatihan dalam file pelatihan.
4. Dapat menerima masukan dari *user* (petani dan masyarakat) berupa gejala-gejala penyakit sebagai masukan untuk dapat mendiagnosa penyakit.
5. Dapat membaca pola pelatihan sesuai aturan Jaringan Saraf Tiruan *Backpropagation*.
6. Dapat menghasilkan keluaran jika aturan sudah sesuai dan masukan gejala yang dimasukkan telah dikenali dalam proses pelatihan.
7. Dapat memberikan laporan hasil diagnosa.

3.2.2 Pengolahan Data

Data yang digunakan dalam aplikasi adalah data gejala penyakit tanaman tomat. Data tidak dibatasi, karena tergantung dari jumlah data, dimana data *langsung dimasukkan ke dalam aplikasi tanpa harus mengalami perubahan*. Pengolahan data hanya mengubah data gejala penyakit ke dalam bentuk biner, yaitu bentuk matrik 1x42, untuk menghasilkan suatu pola yang digunakan sebagai masukan data Jaringan Saraf Tiruan. Dengan membedakan apakah gejala bernilai *true* (1) atau *false* (0). Alur pengolahan data yang dilakukan dalam aplikasi antara lain :

- Langkah 1 : Mengambil data gejala yang sudah disimpan dalam berkas.
- Langkah 2 : Setelah gejala diambil, maka maaukan matrik akan mengubah gejala dalam bentuk matrik diikuti munculnya target yang telah ditentukan sebelumnya.
- Langkah 3 : Selesai memasukkan gejala, lakukan dengan masukan untuk pelatihan jaringan saraf.
- Langkah 4 : Masukkan jumlah *hidden layer*, maksimum epoh, target *error*, dan *learning rate*.
- Langkah 5 : Setelah memasukkan data-data yang dibutuhkan dalam jaringan saraf, kemudian tekan tombol proses untuk memproses jaringan.
- Langkah 6 : Kemudian muncul hasil dari segala proses yang telah dilakukan.

3.3 Perancangan Sistem

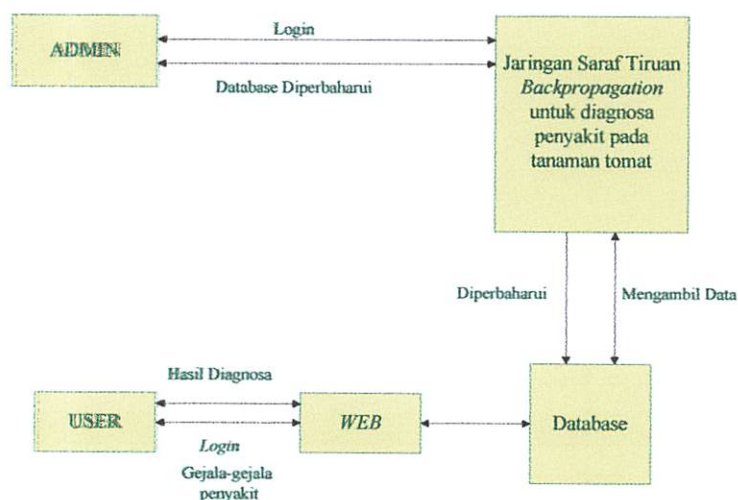
Rancangan dan langkah-langkah yang dilakukan dalam memberikan diagnosis penyakit tanaman tomat menggunakan Jaringan Saraf Tiruan *Backpropagation*.

Penelitian dilakukan dengan langkah-langkah sebagai berikut :

1. Mempelajari metode yang digunakan, yaitu *Backpropagation* dan penyakit pada tanaman tomat.
2. Menganalisa dan merancang sistem dengan menggunakan hasil pembelajaran pada tahap sebelumnya.
3. Membuat sistem berdasarkan analisis dan perancangan yang dilakukan.
4. Uji coba sistem.
5. Evaluasi hasil uji coba.

3.4 Desain Sistem

Sebelum membangun sebuah sistem, perlu merancang desain sistem terlebih dahulu, untuk dapat mempermudah pembuatan sistem, dan memberikan gambaran umum tentang sistem tersebut. Dijelaskan pada gambar 3.1



Gambar 3.1 Desain Sistem Jaringan Saraf Tiruan *Backpropagation* untuk Mendiagnosa Penyakit pada Tanaman Tomat

3.5 Gambaran Umum Perangkat Lunak

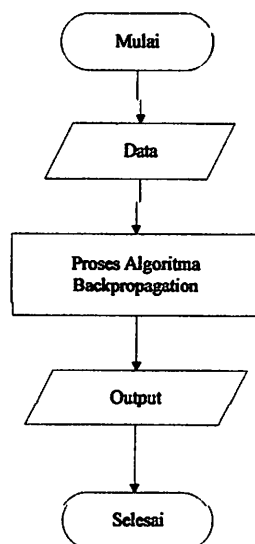
Sistem yang dikembangkan adalah implementasi dari algoritma *Backpropagation* dalam bentuk aplikasi untuk mendiagnosa jenis penyakit pada tanaman tomat. Sistem akan mendiagnosa gejala yang diberikan oleh user dan akan memberikan keluaran berupa hasil diagnosa penyakit terhadap gejala-gejala tersebut.

Dari masukan tersebut akan dikelompokkan ke dalam 12 jenis penyakit, sehingga akan terdapat 12 *neuron* pada lapisan keluaran jaringan saraf. Pada tahap pembelajaran menggunakan metode *feedforward*, lalu akan dihitung besarnya nilai *error*. Jika kesalahan yang ditargetkan belum didapat, maka dilakukan *backpropagation*. Jaringan terdiri dari satu lapisan masukan dengan 42 *neuron*, satu *hidden layer*, dan satu lapisan keluaran dengan 12 *neuron*. Data dilatih hingga mendapatkan *error* yang minimum, bila nilai *error* minimum, maka bobot akan disimpan dalam data penyimpanan. Tempat penyimpanan menggunakan *database MySQL*.

Pada tahap pengujian sistem, dengan menggunakan bobot-bobot akhir yang telah diperoleh selama proses pembelajaran, maka dihasilkan pengelompokan terhadap data pengujian.

Langkah-langkah untuk mendiagnosa penyakit pada tanaman tomat menggunakan Jaringan Saraf Tiruan *Backpropagation* diilustrasikan dalam gambar 3.2, dengan tahapan sebagai berikut :

1. Data masukan (*input*) adalah gejala-gejala klinis penyakit.
2. Proses algoritma *backpropagation* terdiri dari proses pelatihan dan pengujian data.
3. Keluaran (*output*), merupakan hasil diagnosis penyakit pada tanaman tomat.



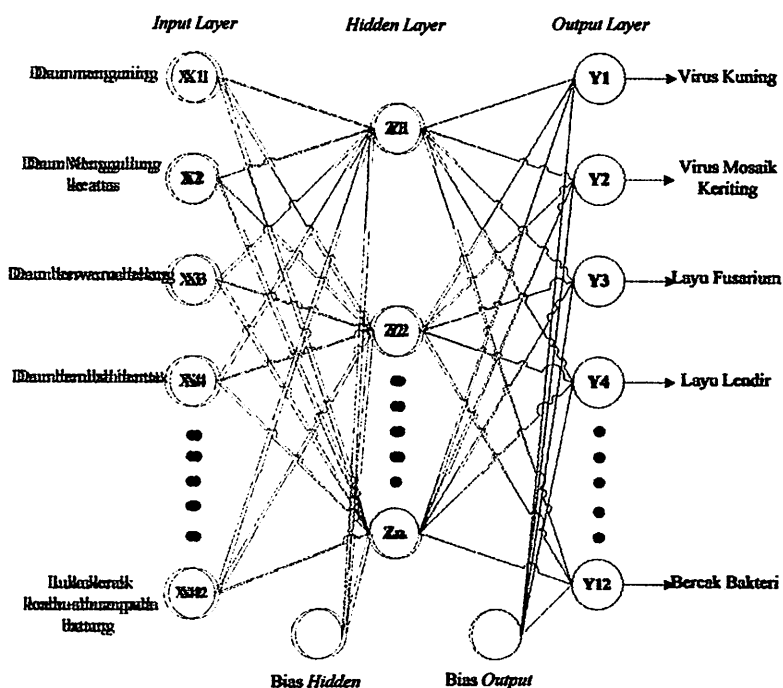
Gambar 3.2 Diagram Alir Sistem Secara Umum

3.6 Proses Jaringan Saraf Tiruan *Backpropagation*

Proses dibagi menjadi tiga, yaitu proses membuat arsitektur jaringan, proses pelatihan, dan proses pengujian.

3.6.1 Arsitektur

Arsitektur dari *backpropagation* terdiri dari lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Arsitektur *jaringan saraf yang digunakan* terdiri dari 42 unit *neuron* pada lapisan masukan dan 12 unit *neuron* pada lapisan keluaran. Jumlah *neuron* pada lapisan tersembunyi (*hidden layer*) adalah n , dimana n akan dicoba dalam beberapa nilai. Hal ini dilakukan untuk mendapatkan arsitektur jaringan yang sesuai dalam melakukan pengenalan. Arsitektur Jaringan Saraf Tiruan *Backpropagation* untuk mendiagnosa penyakit tanaman tomat dijelaskan pada gambar 3.3.



Gambar 3.3 Arsitektur Jaringan Saraf Tiruan *Backpropagation*

3.6.2 Penetapan Masukan (*Input*)

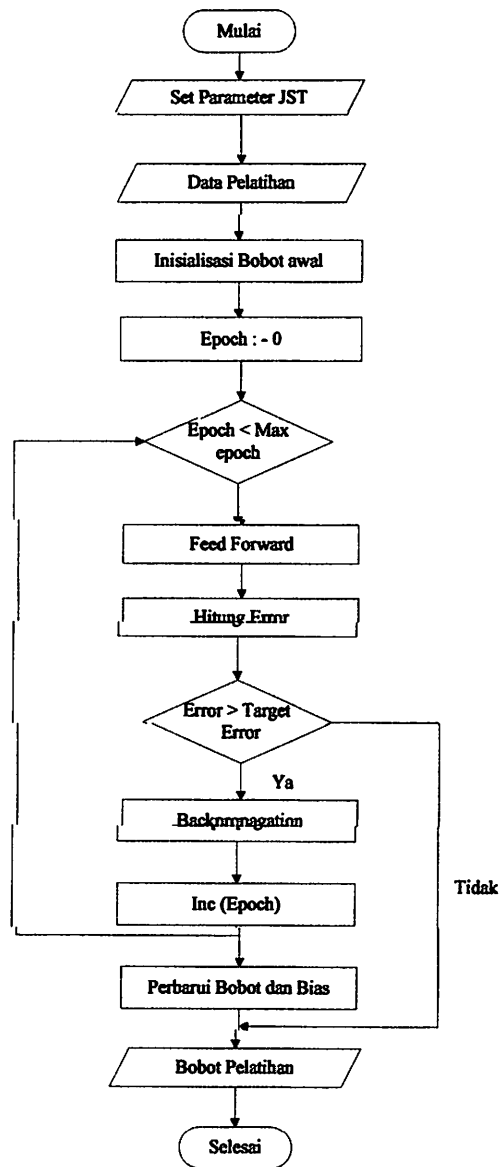
Data masukan yang digunakan dalam aplikasi ada dua jenis. Masukan data pertama adalah data gejala penyakit dan masukan data kedua adalah Jaringan Saraf Tiruan untuk pelatihan jaringan.

Masukan data gejala yang digunakan akan dijadikan data biner, berupa matrik ordo 1x42. Hasil pengolahan data gejala penyakit digunakan sebagai masukan data pelatihan jaringan saraf.

3.6.3 Proses Pelatihan

Algoritma pelatihan backpropagation terdiri dari tiga langkah, yaitu: langkah maju (*feedforward*), propagasi balik (*backpropagation*), dan perubahan bobot. Langkah-langkah terdapat pada gambar 3.4, yaitu :

1. Mulai.
2. Tentukan kesalahan yang ditargetkan, epoch maksimum, dan *neuron hidden*.
3. Masukan data yang akan dilatih.
4. Inisialisasi bobot awal, yang berkisar antara -0,05 sampai 0,05.
5. Tentukan parameter epoch sama dengan 0 (nol).
6. Lakukan *feedforward* setelah jaringan terisi bobot.
7. Hitung kesalahan antara pola keluaran dan pola target.
8. Apakah kesalahan keluaran lebih besar dari kesalahan yang ditargetkan? Jika ya, lakukan langkah 9, jika tidak, maka lakukan langkah 10.
9. Lakukan *backpropagation*.
10. Proses berhenti, bobot dan bias akhir pelatihan disimpan, dan siap untuk digunakan dalam proses pengujian.
11. Selesai.



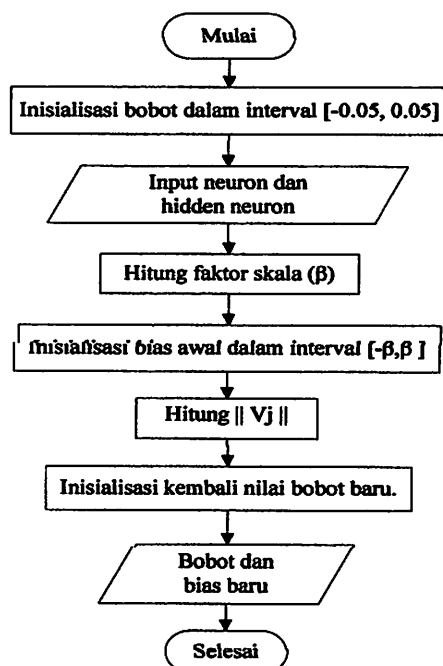
**Gambar 3.4 Diagram Alir Proses Pelatihan Jaringan Saraf Tiruan
*Backpropagation***

3.6.3.1 Inisialisasi Bobot dan Bias Awal Nguyen Widrow

Inisialisasi digunakan untuk menentukan nilai bobot dan bias awal dari unit masukan ke unit tersembunyi. Bertujuan untuk mempercepat iterasi.

Untuk proses inisialisasi terdapat pada gambar 3.5, yaitu :

1. Mulai.
2. Inisialisasi semua bobot dengan bilangan acak antara -0,05 sampai 0,05.
3. Tentukan jumlah unit masukan dan unit tersembunyi.
4. Hitung faktor skala menggunakan rumus $\beta = 0.7^n \sqrt{p}$.
5. Inisialisasi bias awal dari unit masukan ke unit tersembunyi, dengan nilai $[-\beta, \beta]$.
6. Hitung $\|V_j\|$ menggunakan persamaan 2.21.
7. Inisialisasi kembali bobot baru menggunakan persamaan 2.22.
8. Selesai.

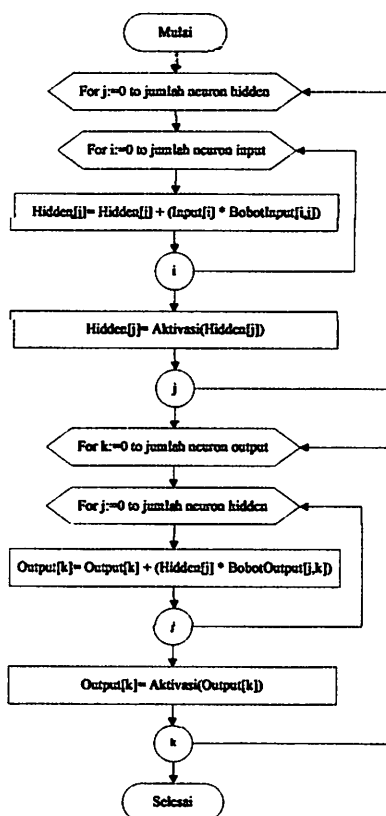


Gambar 3.5 Diagram Alir Proses Inisialisasi Bobot dan Bias Awal Nguyen Widrow

3.6.3.2 Feedforward

Proses yang dilakukan dalam tahap ini adalah menjumlahkan perkalian antara masukan dengan bobot yang ada dan menghitung nilai aktivasinya untuk kemudian hasilnya digunakan untuk masukan lapisan yang ada di atasnya. Langkah-langkahnya terdapat pada gambar 3.6, yaitu :

1. Mulai.
2. Kalikan seluruh data masukan dengan bobot acak pada masing-masing bobot koneksi yang terhubung dengan masukan *neuron*. Lalu jumlahkan seluruh bobot yang menuju unit tersembunyi yang sama.
3. Aktivasikan hasil penjumlahan pada masing-masing *neuron* di lapisan tersembunyi, sehingga keluaran berada antara 0 (nol) dan 1 (satu).
4. Kalikan seluruh data hasil aktivasi masing-masing *neuron* pada lapisan tersembunyi dengan bobot masing-masing bobot koneksinya. Lalu jumlahkan seluruh vektor bobot yang menuju keluaran.
5. Aktivasikan hasil penjumlahan pada tiap *neuron* lapisan keluaran, sehingga keluaran akan bernilai antara 0 (nol) dan 1 (satu).
6. Selesai.



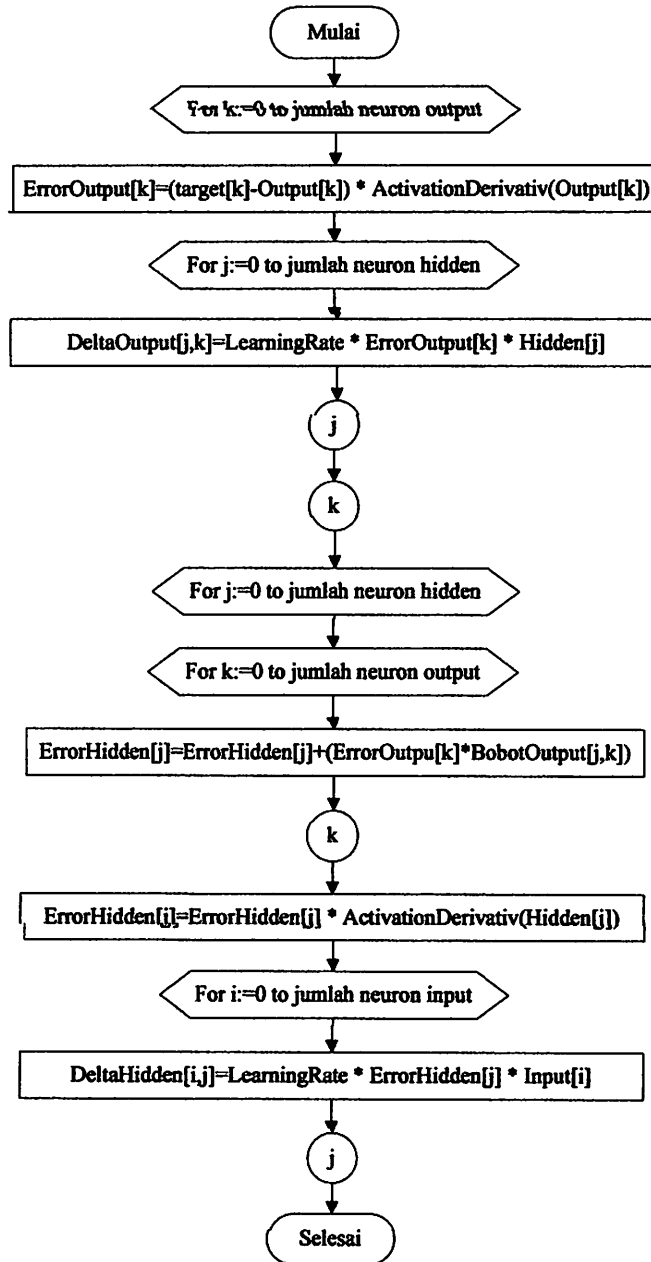
Gambar 3.6 Diagram Alir Proses *Feedforward*

3.6.3.3 Backpropagation

Proses yang dilakukan dalam tahap ini adalah perhitungan informasi kesalahan pada tiap *neuron* masing-masing lapisan, dimulai dari kesalahan pada lapisan keluaran hingga lapisan tersembunyi. Informasi kesalahan berguna untuk menghitung faktor peubah bobot yang digunakan untuk memperbaiki bobot lama.

Langkah-langkah *backpropagation* terdapat pada gambar 3.7, yaitu :

1. Mulai.
2. Hitung selisih antara target pelatihan dengan keluaran. Selisih disebut *error*. Kalikan selisih dengan keluaran yang telah diaktivasi dengan fungsi turunan aktivasi. Hasil perkalian merupakan faktor kesalahan pada lapisan keluaran dan akan digunakan untuk menghitung faktor kesalahan pada lapisan tersembunyi, serta untuk menghitung faktor peubah bobot.
3. Hitung besar faktor peubah bobot terbaru dengan mengalikan *learning rate* dengan lapisan tersembunyi dan *error* pada lapisan keluaran.
4. Masing-masing faktor kesalahan di keluaran dikalikan dengan bobot lama yang terkoneksi dengan *neuron* lapisan keluaran. Lalu hasil perkalian pada semua koneksi pada *neuron* dijumlahkan. Faktor kesalahan ini akan digunakan untuk menghitung peubah bobot dari lapisan masukan ke lapisan tersembunyi.
5. Mengalikan *learning rate* dengan lapisan masukan dan *error* (pada lapisan tersembunyi) untuk mendapatkan nilai faktor peubah bobot baru pada tiap vektor.
6. Selesai.

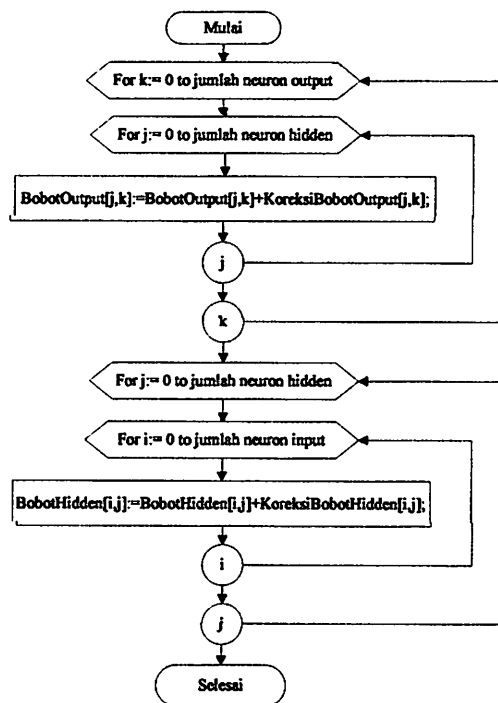


Gambar 3.7 Diagram Alir Proses Backpropagation

3.6.3.4 Update Bobot

Hasil dari proses *backpropagation* adalah nilai faktor peubah bobot yang digunakan untuk melakukan perubahan bobot baru. Langkah-langkah dalam *update bobot* terdapat pada gambar 3.8, yaitu :

1. Mulai.
2. Perbaiki nilai bobot untuk tiap koneksi menuju lapisan keluaran dengan cara menjumlahkan nilai bobot lama dengan peubah bobot yang telah dihitung pada proses *backpropagation*.
3. Perbaiki nilai bobot ke lapisan tersembunyi dengan menjumlahkan nilai bobot lama dengan peubah bobot yang telah dihitung pada proses *backpropagation*.
4. Selesai.

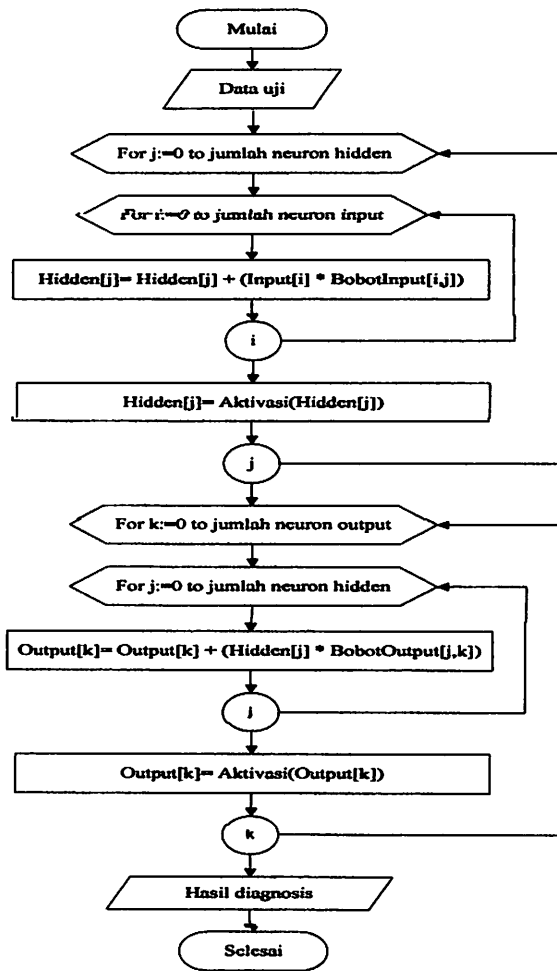


Gambar 3.8 Diagram Alir Proses Perubahan Bobot

3.6.4 Proses Pengujian

Tahap pengujian Jaringan Saraf Tiruan *Backpropagation* dilakukan dengan tahap *feedforward* dari algoritma pelatihan. Dalam proses pengujian, hasil *keluaran jaringan tidak diproses ke backpropagation*. Langkah-langkah pengujian terdapat pada gambar 3.9, yaitu :

1. Mulai.
2. Inisialisasi bobot-bobot yang diperoleh dari proses pelatihan jaringan.
3. Masukkan data pengujian.
4. Kalikan seluruh data masukan pada *neuron* masukan dengan bobot acak pada masing-masing bobot yang terhubung dengan *neuron* masukan. Lalu jumlahkan seluruh vektor bobot menuju *neuron* tersembunyi.
5. Aktivasikan hasil penjumlahan pada masing-masing *neuron* di lapisan tersembunyi, sehingga keluaran akan bernilai antara 0 (nol) dan 1 (satu).
6. Kalikan seluruh data hasil aktivasi masing-masing *neuron* lapisan tersembunyi dengan bobot masing-masing koneksi bobot keluaran. Lalu jumlahkan seluruh vektor bobot yang menuju *neuron* keluaran.
7. Aktivasikan hasil penjumlahan pada masing-masing *neuron* di lapisan keluaran, sehingga keluaran akan bernilai antara 0 (nol) dan 1 (satu).
8. Keluaran berupa hasil diagnosis penyakit.
9. Selesai.



Gambar 3.9 Diagram Alir Proses Pengujian Jaringan Saraf Tiruan

3.7 Perancangan *Form* Utama

Perancangan *form* digunakan untuk menghubungkan perangkat lunak dengan pengguna, agar dapat dioperasikan. Berikut adalah gambar rancangan *menu* untuk *mendiagnosa* penyakit tanaman tomat. Terdapat pada gambar 3.10.

- Tampilan menu utama pada halaman admin

Tampilan ini berada pada halaman admin (pada *software* Delphi 7).

The image shows a screenshot of a software application window titled "Applikasi Deteksi Penyakit Tomat". The window contains a vertical menu on the left side with the following items: "Master", "Master Penyakit", "Master Gejala", "Master Training", "Pengisian Data Training", "Data Training", "Tes Training", "Proses Pelatihan", "Data Pelatihan", "Setting User", and "Master User". To the right of the menu is a large, empty rectangular area, likely intended for a main content or data display.

Gambar 3.10 *Form* Menu untuk Diagnosa Penyakit Tanaman Tomat

- Tampilan menu utama pada halaman pengguna (*user*)
Tampilan ini berada pada halaman *web*. Terdapat pada gambar 3.11.

Beranda	Diagnosa Penyakit Tanaman Tomat	Informasi Tentang Tanaman Tomat
Selamat Datang		Aplikasi Diagnosa Penyakit Tanaman Tomat
Copyright		

Gambar 3.11 *Form Menu Web* untuk Diagnosa Penyakit Tanaman Tomat

a. Rancangan *Form* Pelatihan

Pada rancangan *form* ini terdiri dari tombol Master Penyakit, Master Gejala, Pengisian Data Training, Data Training, Proses Pelatihan, Data Pelatihan, dan Master User. Terdapat pada gambar 3.12.

Aplikasi Deteksi Penyakit Tomat

<p>Master</p> <p>Master Penyakit</p> <p>Master Gejala</p> <p>Master Training</p> <p>Pengisian Data Training</p> <p>Data Training</p> <p>Tes Training</p> <p>Proses Pelatihan</p> <p>Data Pelatihan</p> <p>Setting User</p> <p>Master-User</p>	<table style="width: 100%;"> <tr> <td style="width: 33%;">Hitungan Backpropagation</td> <td style="width: 33%;">Hasil Error</td> <td style="width: 33%;">Ketinggalan</td> </tr> <tr> <td colspan="3" style="text-align: center;">Pelatihan Data Tes Dengan Data Training</td> </tr> <tr> <td colspan="3" style="text-align: center;">Parameter</td> </tr> <tr> <td colspan="3" style="text-align: center;"> <input type="button" value="Proses"/> <input type="button" value="Stop"/> <input type="button" value="Start"/> </td> </tr> </table>	Hitungan Backpropagation	Hasil Error	Ketinggalan	Pelatihan Data Tes Dengan Data Training			Parameter			<input type="button" value="Proses"/> <input type="button" value="Stop"/> <input type="button" value="Start"/>		
Hitungan Backpropagation	Hasil Error	Ketinggalan											
Pelatihan Data Tes Dengan Data Training													
Parameter													
<input type="button" value="Proses"/> <input type="button" value="Stop"/> <input type="button" value="Start"/>													

Gambar 3.12 Rancangan *Form* Pelatihanb. Rancangan *Form* Diagnosa Penyakit (terdapat pada halaman *web*)

Pada rancangan ini berada pada halaman *web*. Terdiri dari 3 menu utama pada *web* (*Beranda*, *Diagnosa Penyakit Tanaman Tomat*, dan *Informasi Tentang Tanaman Tomat*), radio button, dan tombol *Simpan*. Terdapat pada gambar 3.13.

Beranda	Diagnosa Penyakit Tanaman Tomat	Informasi Tentang Tanaman Tomat
Selamat Datang		Aplikasi Diagnosa Penyakit Tanaman Tomat
Kode <input type="text"/>		
Nama <input type="text"/>		
	Gejala	Y N
		<input type="radio"/> <input type="radio"/>
		<input type="radio"/> <input type="radio"/>
		<input type="radio"/> <input type="radio"/>
		<input type="button" value="Simpan"/>
<input type="text" value="Copyright"/>		

Gambar 3.13 Rancangan *Form* Diagnosa Penyakit pada Tampilan *Web*c. Rancangan *Form* Tampilan Hasil Diagnosa Penyakit (terdapat pada halaman *web*)

Pada rancangan ini berada pada halaman *web*. Terdiri dari 3 menu utama pada *web* (*Beranda*, *Diagnosa Penyakit Tanaman Tomat*, dan *Informasi Tentang Tanaman Tomat*), dan tombol *Lihat Hasil Diagnosa*. Terdapat pada gambar 3.14.

Beranda	Diagnosa Penyakit Tanaman Tomat	Informasi Tentang Tanaman Tomat
Selamat Datang		Aplikasi Diagnosa Penyakit Tanaman Tomat
Hasil Diagnosa		
Kode	<input type="text"/>	
Nama	<input type="text"/>	
<input type="button" value="Lihat Diagnosa"/>		
<input type="text" value="Copyright"/>		

Gambar 3.14 Rancangan *Form* Hasil Diagnosa Penyakit pada Tampilan *Web*

3.8 Struktur Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan terdiri dari kumpulan data atau fakta-fakta yang digunakan untuk menyelesaikan permasalahan yang khusus. Basis pengetahuan yang digunakan dalam sistem ini berupa aturan nama penyakit dan gejala penyakit. Dijelaskan pada tabel 3.1 dan tabel 3.2 berikut.

Tabel 3.1 Pembentukan Aturan Penyakit Tanaman Tomat

No	Kode Penyakit	Nama Penyakit	Gejala
1	001	Virus Kuning	1) Daun menguning 2) Daun melengkung ke atas
2	002	Virus Mosaik Keriting	1) Daun berwarna belang (antara hijau tua dan hijau muda) 2) Daun berubah bentuk (keriting) 3) Daun berpola bercak-bercak seperti mosaik
3	003	Layu Fusarium	1) Tanaman Layu 2) Daun berubah warna menjadi kuning 3) Akar mengalami diskolorasi (berwarna

			coklat)
4	004	Layu Lendir	<ol style="list-style-type: none"> 1) Daun layu 2) Bagian bawah daun menguning 3) Daun menggulung ke bawah
5	005	Hawar Daun	<ol style="list-style-type: none"> 1) Bercak berwarna hijau muda sampai kecoklatan pada daun 2) Pada permukaan bawah daun terdapat gejala busuk, yaitu berwarna abu-abu keputihan, sampai menjadi bercak besar berwarna coklat 3) Daun berubah warna menjadi coklat 4) Terlihat bercak gelap seperti berminyak pada buah
6	006	Kapang Daun	<ol style="list-style-type: none"> 1) Terdapat bercak coklat pada permukaan daun 2) Pada bagian bawah daun terdapat spora-spora berwarna kelabu kekuning-kuningan 3) Menyerang pada daun-daun bagian bawah, hingga ke seluruh tanaman
7	007	Busuk Daun	<ol style="list-style-type: none"> 1) Bercak coklat kehitaman pada daun hingga tangkai daun 2) Pada bagian bawah daun terdapat jamur berwarna putih keunguan seperti beludru 3) Bercak berair pada pangkal buah yang berwarna hijau hingga kecoklatan 4) Pada buah terdapat bercak-bercak berbentuk cincin yang berwarna coklat tua 5) Bercak pada buah agak keras dan berkerut
8	008	Bercak Daun Septoria	<ol style="list-style-type: none"> 1) Pada permukaan bawah daun terdapat bercak bulat kecil (bercak berwarna coklat muda) 2) Daun menggulung ke bawah

9	009	Bercak Coklat	<ol style="list-style-type: none"> 1) Daun terdapat bercak berwarna coklat sampai kehitaman 2) Bercak pada daun akan membesar, hingga daun berubah warna menjadi kuning, layu dan mati 3) Menyebabkan keguguran pada bunga 4) Buah tomat yang masih muda atau sudah <i>matang menjadi busuk, berwarna hitam dan cekung</i> 5) Menyerang pangkal buah (ujung tangkai) yang menyebabkan perubahan warna menjadi coklat, dan tertutup spora hitam seperti beludru
10	010	Busuk Buah Rhizoctonia	<ol style="list-style-type: none"> 1) Bercak cekung kecil berwarna coklat pada buah 2) Batang tanaman terpilin 3) Pembusukan dengan warna coklat kemerahan pada pangkal batang
11	011	Bercak Buah Antraknosa	<ol style="list-style-type: none"> 1) Bercak kecil berwarna coklat, bulat, dan berair pada buah 2) Bercak ungu dekat tangkai buah 3) Daun menjadi layu 4) Akar mengalami disolorasi (berwarna coklat)
12	012	Bercak Bakteri	<ol style="list-style-type: none"> 1) Bercak kecil berair pada batang dan daun 2) Daun menggulung (keriting) ke bawah 3) Daun kering 4) Bercak berair, lalu berubah menjadi bercak seperti gabus pada buah 5) Terdapat luka seperti kerak keabu-abuan pada batang

Tabel 3.2 Pengurutan Gejala untuk Penyakit Tanaman Tomat

No	Kode Gejala	Gejala
1	G001	Daun menguning
2	G002	Daun menggulung ke atas
3	G003	Daun berwarna belang (antara hijau tua dan hijau muda)
4	G004	Daun berubah bentuk (keriting)
5	G005	Daun berpola bercak-bercak seperti mosaik
6	G006	Tanaman layu
7	G007	Daun berubah warna menjadi kuning
8	G008	Akar akan mengalami diskolorasi (berwarna coklat)
9	G009	Daun layu
10	G010	Bagian bawah daun menguning
11	G011	Daun menggulung ke bawah
12	G012	Bercak berwarna hijau muda sampai kecoklatan pada daun
13	G013	Pada permukaan bawah daun terdapat gejala busuk yaitu berwarna abu-abu keputihan, sampai menjadi bercak besar berwarna coklat
14	G014	Daun berubah warna menjadi coklat
15	G015	Terlihat bercak gelap seperti berminyak pada buah
16	G016	Terdapat bercak coklat pada permukaan daun

17	G017	Pada bagian bawah daun terdapat spora-spora berwarna kelabu kekuning-kuningan
18	G018	Menyerang pada daun-daun bagian bawah, hingga ke seluruh tanaman
19	G019	Bercak coklat kehitaman pada daun hingga tangkai daun
20	G020	Pada bagian bawah daun terdapat jamur berwarna putih keunguan seperti beludru
21	G021	Bercak berair pada pangkal buah yang berwarna hijau hingga kecoklatan
21	G022	Pada buah terdapat bercak-bercak berbentuk cincin yang berwarna coklat tua
23	G023	Bercak pada buah agak keras dan berkerut
24	G024	Pada permukaan bawah daun terdapat bercak bulat kecil berair (bercak berwarna coklat muda)
25	G025	Daun menggulung ke bawah
26	G026	Daun terdapat bercak berwarna coklat sampai kehitaman
27	G027	Bercak pada daun akan membesar, hingga daun berubah warna menjadi kuning, layu dan mati
28	G028	Menyebabkan keguguran pada bunga
29	G029	Buah tomat yang masih muda atau sudah matang menjadi busuk, berwarna hitam, dan cekung
30	G030	Menyerang pangkal buah (ujung tangkai), yang menyebabkan perubahan warna menjadi coklat, dan tertutup spora hitam seperti beludru

31	G031	Bercak cekung kecil berwarna coklat pada buah
32	G032	Batang tanaman terpilin
33	G033	Pembusukan dengan warna coklat kemerahan pada pangkal batang
34	G034	Bercak kecil berwarna coklat, bulat, dan berair pada buah
35	G035	Bercak ungu dekat tangkai buah
36	G036	Daun menjadi layu
37	G037	Akar akan mengalami diskolorasi (berwarna coklat)
38	G038	Bercak kecil berair pada batang dan daun
39	G039	Daun menggulung (keriting) ke bawah
40	G040	Daun kering
41	G041	Bercak berair, lalu berubah menjadi bercak seperti gabus pada buah
42	G042	Terdapat luka seperti kerak keabu-abuan pada batang

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

Lingkungan implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan untuk pengembangan sistem diagnosis jenis penyakit mata pada manusia menggunakan *JST Backpropagation*.

4.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem diagnosis awal penyakit dalam manusia menggunakan *JST Backpropagation* adalah :

1. Prosesor Intel Atom Inside™ CPU N280 @1,66 GHz (2CPUs)
2. Memori 1016 MB RAM
3. Harddisk kapasitas 160 GB
4. Monitor 10,1 ”
5. Keyboard
6. Mouse

4.1.2 Perangkat Lunak

Perangkat keras yang digunakan dalam pengembangan sistem diagnosis awal penyakit mata pada manusia menggunakan JST Backpropagation ini adalah :

1. Sistem Operasi *Windows XP Home Edition*
2. Borland Delphi 7
3. MySQL
4. Dreamweaver 8

4.2 Implementasi Program

Pada sub bab ini akan dibahas mengenai implementasi dari sistem diagnosa penyakit mata pada manusia menggunakan JST *Backpropagation*.

4.2.1. Struktur Data

Untuk memudahkan dalam implementasi jaringan saraf tiruan, penentuan struktur data harus dilakukan dengan baik. Struktur data yang digunakan dalam proses jaringan saraf tiruan dapat dilihat dalam Gambar 4.1.

```

const
  MaxIterasi = 63000;
  MaxUnitMasukan = 42;
  MaxUnitHidden = 100;
  MaxUnitKeluaran = 12;
  MaxJumlahData = 10000;
var
  X      : array[1..MaxUnitMasukan ] of Extended;
  X_Data : array[1..MaxUnitMasukan,1..MaxJumlahData] of double;

  T      : array[1..MaxUnitKeluaran ] of Extended;
  T_     : array[1..MaxUnitKeluaran ] of Integer;
  T_Data : array[1..MaxUnitKeluaran,1..MaxJumlahData] of double;

  Y_hitung : array[1..MaxUnitKeluaran,1..MaxJumlahData] of double;
  w,delta_w,w_min : array[0..MaxUnitHidden,1..MaxUnitKeluaran] of
double;
  v,delta_v,v_min : array[0..MaxUnitMasukan,1..MaxUnitHidden] of
double;
  std_min : real;
  z,z_in : array[1..MaxUnitHidden] of double;
  y_in,y : array[1..MaxUnitKeluaran] of double;
  delta : array[1..MaxUnitKeluaran] of double;
  delta_in : array[1..MaxUnitHidden] of double;
  deltaHidden : array[1..MaxUnitHidden] of double;

```

```

JumlahData      : integer;
JumlahMasukanUnit : integer;
JumlahHiddenUnit : integer;
JumlahKeluaranUnit : integer;
iterasi         : integer;

MaxMasukan,
MaxKeluaran    : real;

Zj             : array[0..MaxUnitMasukan-1,0..MaxUnitHidden-1] of
Extended;
Vj             : array[0..MaxUnitMasukan] of Real;

Beta,maxError  : Extended;
Alpha          : real;
MaxEpoch     : integer;
Epoch        : Integer;
ErrorVal      : Extended;
ThresHold    : Extended;
str_          : string;
showlogs     : Boolean;

```

Gambar 4.1 *Source code* Struktur Data Jaringan Saraf Tiruan

Berikut adalah penjelasan dari struktur data yang terdapat dalam Gambar 4.1 :

1. Ndata merupakan jumlah *neuron masukan*. Berupa ketetapan yang bernilai 42 dikarenakan jumlah masukan yang akan dimasukkan ke dalam sistem berjumlah 42 neuron.
2. Nhidden adalah jumlah *neuron hidden*. Berupa ketetapan yang bernilai 100 dikarenakan jumlah hidden neuron yang akan dimasukkan maksimal 100 neuron.

3. N_{target} merupakan jumlah *neuron keluaran*. Berupa ketetapan yang bernilai 12 dikarenakan jumlah keluaran yang akan dihasilkan oleh sistem sebanyak 12 neuron.
4. X berisi *inFormasi* mengenai nilai dari unit masukan.
5. $Z, V, \Delta V$
6. V_j, V_{oj}, Z_{in}, Z_j berisi *inFormasi* mengenai nilai bobot, bias, dan nilai keluaran dari lapisan masukan ke lapisan tersembunyi
7. W_{jk} berisi *inFormasi* mengenai nilai bobot dari lapisan tersembunyi ke lapisan keluaran
8. W_{ok}, Y_{in}, Y_k berisi *inFormasi* mengenai bias dan nilai keluaran dari lapisan hidden ke lapisan keluaran
9. $\Delta, \Delta W_{ok}$ berisi *inFormasi* mengenai faktor kesalahan unit keluaran dan nilai perubahan bias dari lapisan tersembunyi ke lapisan keluaran.
10. ΔW_{jk} berisi *inFormasi* mengenai nilai perubahan bobot dari lapisan tersembunyi ke lapisan keluaran.
11. $\Delta I_{in}, \Delta Net, \Delta V_{oj}$ berisi *inFormasi* mengenai faktor kesalahan di unit tersembunyi serta *inFormasi* nilai perubahan bobot dan bias lapisan masukan ke lapisan tersembunyi.
12. T merupakan *inFormasi* mengenai nilai target.

4.2.2 Prosedur Menghitung Faktor Skala (β)

Prosedur untuk menghitung nilai faktor skala (β) dapat dilihat dalam Gambar 4.2.

```

procedure initParam();
begin
  Randomize;
  DecimalSeparator := '.';
  beta      := ( 0.7 *(power(JumlahHiddenUnit,(1/42))));
  epoch     := 0;
  std_min   := 1;
  iterasi   := 0;
  JumlahInputUnit := 42;
  JumlahHiddenUnit := 15;
  JumlahOutputUnit := 12;
  MaxEpoch := MaxIterasi;
end;

```

Gambar 4.2 *Source Code* Hitung Faktor Skala

Prosedur di atas berfungsi untuk menghitung faktor skala yang akan digunakan pada inialisasi bobot dan bias awal Nguyen Widrow.

4.2.3 Prosedur Inialisasi Bobot Awal Nguyen Widrow

Prosedur *InitBobot* dalam Gambar 4.3 merupakan prosedur inialisasi bobot awal Nguyen Widrow yang bertugas untuk membuat inialisasi bobot dan bias awal dari unit-unit yang berada pada lapisan masukan ke unit-unit yang berada pada lapisan tersembunyi sehingga menghasilkan iterasi lebih cepat. Bobot diinisialisasi secara acak dengan rentang nilai antara -0.05 sampai 0.05 dapat dilihat dalam Gambar 4.3.

```

procedure initBobot();
var i,j: integer;
    temp : real;
begin
  //init bobot awal untuk metode nguyen widrow
  for i:= 0 to JumlahMasukanUnit-1 do
  for j:= 0 to JumlahHiddenUnit-1 do
  begin
    Randomize;
    temp := RandomRange(-5,5)/100;
    Zj[i,j] := (temp);
  end;
end;

```



```
end;
end;
```

Gambar 4.3 Prosedur Inisialisasi Bobot Awal Nguyen Widrow

4.2.4 Prosedur Inisialisasi Bobot

Setelah diperoleh nilai bobot dan bias acak awal dari lapisan masukan ke lapisan tersembunyi, maka nilai bobot dan bias tersebut dimodifikasi menggunakan metode Nguyen Widrow. Tahap selanjutnya adalah menginisialisasi nilai awal bobot dan bias dari unit-unit yang berada pada lapisan tersembunyi ke unit-unit yang berada pada layer keluaran. Bobot diinisialisasi secara acak dengan nilai berkisar antara -0,05 sampai 0,05. Prosedurnya ditunjukkan dalam Gambar 4.4.

```
procedure NewBobot();
var i,j: integer,
    temp : real;
    tempDate :real;
begin
//init bobot awal untuk metode nguyen widrow
initBobot();

//hitung VJ
for i:=0 to JumlahMasukanUnit-1 do
begin
temp := 0;
for j:= 0 to JumlahHiddenUnit-1 do
begin
temp := temp + sqr(Zj[i,j]);
end;
Vj[i] := roundVal( sqrt(temp));
end;

//Hitung VJ Baru
```

```

for i:= 0 to JumlahMasukanUnit-1 do
for j:= 0 to JumlahHiddenUnit-1 do
begin
  V[i,j] := (Beta * Zj[i,j])/Vj[i];
end;
//Generate random value Wjk
for i:= 0 to JumlahHiddenUnit-1 do
for j:= 1 to JumlahKeluaranUnit-1 do
begin
  Randomize;
  temp := RandomRange(-5,5)/100;
  W[i,j] := (temp);
end;
end;
end;

```

Gambar 4.4 Prosedur Inisialisasi Bobot dan Bias Awal

4.2.5 Prosedur Feedforward

Prosedur *Feedforward* merupakan prosedur yang bertugas untuk mengaktifkan unit-unit yang berada pada lapisan tersembunyi dan lapisan keluaran dengan menggunakan fungsi aktivasi pada persamaan 2.1. Prosedurnya dapat dilihat dalam Gambar 4.5.

```

procedure feedforward();
var temp: Real;
    temp2 : Extended ;
    temp3 :Extended;
    ij: integer;
begin
  Application.ProcessMessages;
  for j := 0 to NLayer-1 do
  begin
    temp :=0;
    for i := 0 to Ndata-1 do
    begin

```

```

temp:= temp+(X[i]*V[j,i]);
end;
Zin[j] := roundVal( voj[j] + temp );
end;

for j := 0 to NLayer-1 do
begin

temp2 :=1/(1+exp(-Zin[j]));
Zj[j] :=roundVal( temp2 );
end;

for j := 0 to NKeluaran-1 do
begin
temp :=0;
for i := 0 to Nhidden-1 do
begin
temp:= temp+(Zj[i]*Wjk[j,i]);
end;
Yin[j] := roundVal(Wok[j] + temp);
end;

for j := 0 to NKeluaran-1 do
begin
temp := roundVal( 1/(1+exp(-( Yin[j]))) );
Yk[j] := temp ;
end;

```

Gambar 4.5 Prosedur *Feedforward*

4.2.6 Prosedur Hitung *Error*

Nilai keluaran (y) yang diperoleh dari proses feedforward, digunakan untuk menghitung informasi error pada lapisan keluaran. Dalam Gambar 4.6 menunjukkan prosedur untuk menghitung error pada lapisan keluaran. Nilai error

ini digunakan untuk mendapatkan nilai koreksi bobot pada lapisan keluaran yang nantinya digunakan untuk menghitung nilai bobot baru pada lapisan keluaran.

```

procedure hitungError() ;

var
temp : real;
j:integer,
begin
temp :=0;
for j := 0 to NKeluaran-1 do
begin
temp := temp + sqr(T[j] - Yk[j]);
end;
ErrorVal :=(temp/2);
end;

```

Gambar 4.6 Prosedur Hitung *Error*

4.2.7 Prosedur *Backward*

Prosedur untuk melakukan proses *backward* terdiri dari perhitungan faktor kesalahan di unit keluaran, perhitungan nilai perubahan bobot lapisan tersembunyi ke lapisan keluaran, perhitungan perubahan bias lapisan tersembunyi ke lapisan keluaran, perhitungan faktor kesalahan di unit tersembunyi, perhitungan nilai perubahan bobot lapisan masukan ke lapisan tersembunyi, dan perhitungan perubahan bias lapisan masukan ke lapisan tersembunyi. Prosedur *backward* dapat dilihat dalam Gambar 4.7.

```

procedure backward();
var i,j : integer;
temp : Extended;
begin
Application.ProcessMessages;
for j := 0 to NKeluaran-1 do
begin

```

```

Delta[j]:=roundVal((T[j]-Yk[j])*( Yk[j])* (1-Yk[j]));
end;

for j := 0 to NKeluaran-1 do
begin
for i := 0 to NLayer - 1 do
begin
DeltaWjk[j,i]:=roundVal(Delta[j]*Zj[i]*Aplha);
end;
end;

for j := 0 to NKeluaran-1 do
begin
DeltaWok[j]:=roundVal( Aplha * Delta[j]);
end;

for j := 0 to NLayer - 1 do
begin
temp := 0;
for i := 0 to NKeluaran - 1 do
begin
temp:=temp+(Delta[i]*Wjk[i,j]* Aplha);
end;
DeltaIn[j] :=roundVal(temp);
end;

for j := 0 to NLayer-1 do
begin
DeltaNet[j]:=roundVal(DeltaIn[j]*((Zj[j])*(1-Zj[j])));
end;

for i:= 0 to Ndata-1 do
for j:= 0 to NLayer-1 do
begin

```

```

DeltaV[j,i]:=roundVal(Aplha*X[i]*DeltaNet[j]);
end;
end;

```

Gambar 4.7 Prosedur *backward*

4.2.8 Prosedur *Update Bobot*

Prosedur update bobot digunakan untuk menghitung bobot dan bias baru dari unit-unit yang berada pada lapisan masukan ke unit-unit yang berada pada lapisan tersembunyi, dan juga bobot dan bias baru dari unit-unit yang berada pada lapisan tersembunyi ke unit-unit yang berada pada layer keluaran. Prosedur perbarui bobot dan bias dapat dilihat dalam Gambar 4.8

```

procedure PerbaruiBobot ();
var i,j : integer,
    temp : Extended;
begin
  for i:= 0 to NLayer-1 do
    begin
      DeltaVoj[i]:=roundVal(Aplha*DeltaNet[i]);
    end;

    for i:= 0 to Ndata-1 do
      for j:= 0 to NLayer-1 do
        begin
          V[j,i]:= roundVal(V[j,i]+ DeltaV[j,i]);
        end;

        for i:= 0 to NLayer-1 do
          begin
            Voj[i]:=roundVal(Voj[i] + DeltaV[j,i] );
          end;

          for i:= 0 to NKeluaran-1 do

```

```

for j:= 0 to NLayer-1 do

begin
Wjk[i,j]:=roundVal(Wjk[i,j]+DeltaWjk[i,j]);
end;

for j := 0 to NKeluaran-1 do
begin
wok[j] := roundVal(wok[j] + DeltaWok[j]);
end;
end;

```

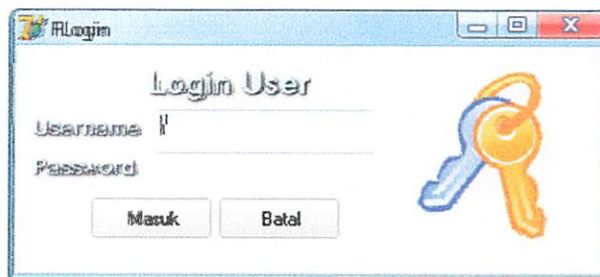
Gambar 4.8 Prosedur perbarui bobot

4.3 Implementasi Program

Berdasarkan perancangan tampilan pada sub bab 3, maka dihasilkan *Form* yang terdiri dari :

1. Tampilan *Form Admin*

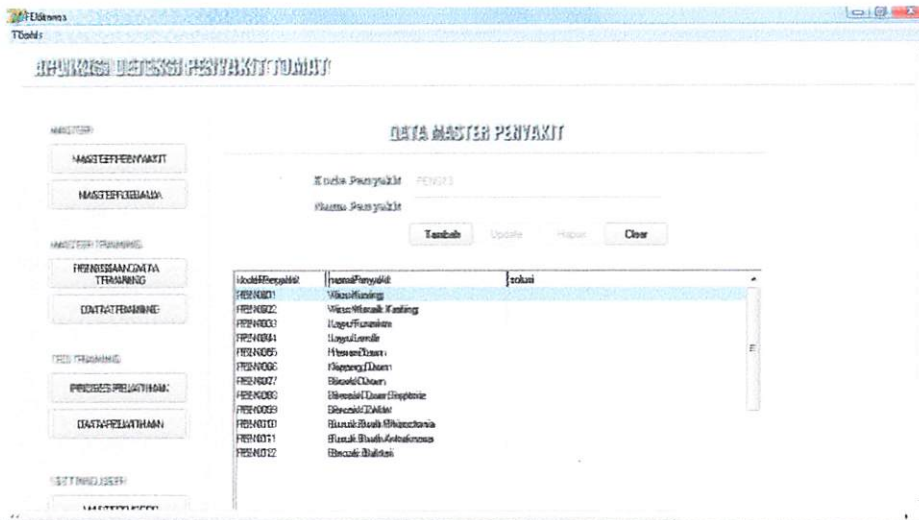
Pada halaman *login*, akan diminta nama *user* dan *password* untuk mengakses aplikasi. *Form User* dapat dilihat Dalam Gambar 4.9



Gambar 4.9 Tampilan *Form Admin*

2. Tampilan *Form Utama*

Menampilkan menu-menu yang dapat dipilih oleh admin untuk dapat menjalankan program. Terdapat pada gambar 4.10.



Gambar 4.10 Tampilan Form Utama

3. Tampilan Form Utama pada Halaman Web

Menampilkan menu-menu yang dapat dipilih oleh pengguna untuk dapat menjalankan program. Terdapat pada gambar 4.11.



Gambar 4.11 Tampilan Form Utama pada Halaman Web

4. Tampilan Form Pelatihan

1. Tampilan pada halaman admin

Langkah-langkah untuk melakukan proses memasukkan dan melatih data (*data training*) adalah dengan cara :

- 1) Klik tombol Pengisian Data Training.

- 2) Pilih jenis penyakit yang akan dilatih, lalu pilih pertanyaan gejala.
 - 3) Setelah memasukkan gejala, klik tombol Simpan.
- Terdapat pada gambar 4.12.

Gambar 4.12 Tampilan Form Pengisian Data Training

2. Tampilan pada halaman pengguna (*web*)

Langkah-langkahnya adalah :

- 1) Pilih menu Diagnosa Penyakit Tanaman Tomat.
- 2) Masukkan nomor kode pengunjung beserta nama.
- 3) Pilih gejala yang sesuai pada *radio button* yang ada. Setelah itu klik tombol Simpan.

Terdapat pada gambar 4.13.

GEJALA	Y	N
00001 Daun menguning	<input type="radio"/>	<input type="radio"/>
00002 Daun menguning ke atas	<input type="radio"/>	<input type="radio"/>
00003 Daun/tanaman belang (antara hijau tua dan hijau muda)	<input type="radio"/>	<input type="radio"/>
00004 Daun/tanaman/bekuk (berling)	<input type="radio"/>	<input type="radio"/>
00005 Daun/tanaman/bekuk/bekuk seperti serok	<input type="radio"/>	<input type="radio"/>

Gambar 4.13 Tampilan Halaman Web untuk Mendiagnosa Penyakit Tomat

3. Tampilan *Form* proses pelatihan pada halaman admin

Langkah-langkahnya adalah :

- 1) Pilih kode tes yang ada pada *Form* (otomatis terhubung pada halaman *web*).
- 2) Tekan tombol Proses, untuk memulai pelatihan, Stop untuk menghentikan proses pelatihan.
- 3) Setelah selesai, maka hasil kesimpulan penyakit, *error*, dan hasil perhitungan akan ditampilkan.

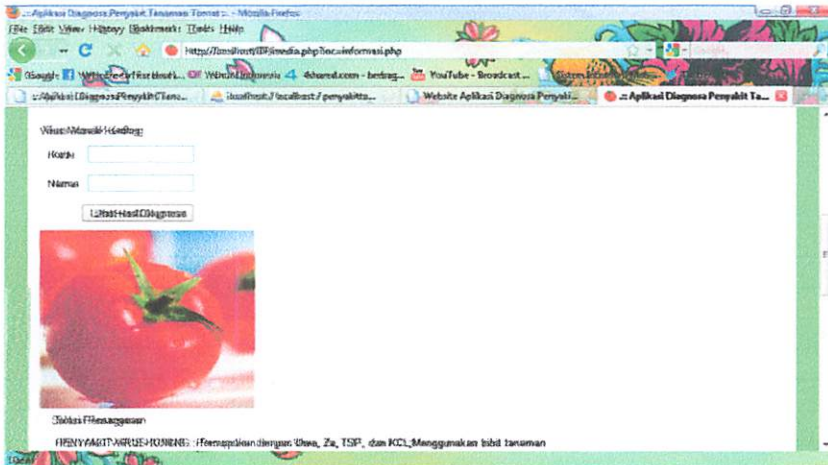
Terdapat pada gambar 4.14.

	TIDAK	TIDAK	TIDAK	0.05	1.7	1.85
GEJ001	Dasar menga 1	1	1	1	1	1
GEJ002	Dasar menga 1	1	1	1	1	1
GEJ003	Dasar bewaa 0	0	0	0	0	0
GEJ004	Dasar bewaa 0	0	0	0	0	0
GEJ005	Dasar bewaa 0	0	0	0	0	0
GEJ006	Tanam lay 0	1	1	1	1	1
GEJ007	Dasar bewaa 0	0	1	1	1	1
GEJ008	Akar akan es 0	1	1	1	1	1
GEJ009	Dasar layu 0	0	0	0	0	0
GEJ010	Bagian bewaa 0	0	0	0	0	0
GEJ011	Dasar menga 0	0	1	1	1	1
GEJ012	Dasar menga 1	1	1	1	1	1

Gambar 4.14 Tampilan *Form* Proses Pelatihan pada Halaman Admin

4. Tampilan *Form* Hasil Diagnosa Penyakit pada halaman *web*

Setelah memasukkan gejala-gejala penyakit, lalu menekan tombol Simpan, maka akan ditampilkan hasil diagnosa penyakit beserta solusi untuk menanganinya. Terdapat pada gambar 4.15.



Gambar 4.15 Tampilan Hasil Diagnosa pada Halaman *Web*

4.4 Penetapan Keluaran (*Output*)

Jika hasil *output* berada pada kisaran $0 \leq output \leq 0,5$ maka hasil *output* tersebut dapat dikategorikan pada nilai 0 atau *false*, sementara jika nilai *output* berada pada kisaran $0,5 > output \leq 1$ maka hasil *output* tersebut dikategorikan pada nilai 1 atau *true*. Penentuan nilai inilah yang nantinya akan menentukan pola gejala penyakit pada masing-masing penyakit mata apakah bernilai *false* atau *true*.

4.5 Hasil dan Pembahasan

Untuk memperoleh struktur Jaringan Saraf Tiruan yang terbaik yang digunakan untuk diagnosis penyakit, maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan melatih jaringan Saraf tiruan dengan parameter-parameter yang berbeda, yang nantinya akan diambil satu struktur jaringan yang terbaik yang digunakan untuk melakukan prediksi jenis penyakit pada tanaman tomat.

4.6 Proses Pelatihan

Untuk proses pelatihan, langkah yang harus dilakukan adalah memasukkan data-data penyakit, beserta gejala-gejalanya. Pertama masukkan terlebih dahulu nama-nama penyakit (12 penyakit), setelah itu memasukkan gejala-gejalanya satu persatu. Terdapat pada gambar 4.16.



Gambar 4.16 Proses Memasukkan Data Penyakit dan Gejala

Setelah memasukkan seluruh data penyakit beserta gejalanya, maka data tersebut akan dilatih. Proses pelatihan tersebut dilakukan dengan cara memilih gejala yang sesuai dengan jenis penyakit yang akan dilatih satu persatu. Pola yang dimasukkan adalah : Ya (bernilai 1), dan Tidak (bernilai 0). Terdapat pada gambar 4.17.

Gambar 4.17 Proses Melatih Data Penyakit dan Gejalanya

Setelah memasukkan pola, maka tekan tombol Simpan. Hasil Pelatihan sesuai pola dapat dilihat pada menu Data Pelatihan. Terdapat pada gambar 4.18.

		TINGKAT1	TINGKAT2	TINGKAT3	TINGKAT4	TINGKAT5	TINGKAT6	TPE
		Virus/Kuning	Virus/Hitam	Lupa	Fusaria	Lupa	Lendir	Hama
		Daun	Daun	Kapur	Daun	Daun	Kapur	Daun Bot
GEA001	Daun mengkilap 1	0	0	0	1	1	0	
GEA002	Daun mengkilap 0	0	0	0	1	1	0	
GEA003	Daun berwarna 0	1	0	0	0	0	0	
GEA004	Daun berubah 0	1	0	0	0	0	0	
GEA005	Daun lepuh 0	1	0	0	1	1	0	
GEA006	Tanaman layu 0	0	1	0	1	1	0	
GEA007	Daun berubah 0	0	1	0	0	1	0	
GEA008	Akris dikun 0	0	0	0	1	1	0	
GEA009	Daun layu 0	0	0	1	1	1	0	
GEA010	Bagian bawah 0	0	0	1	1	0	0	
GEA011	Daun mengkilap 0	0	0	1	1	0	0	

Gambar 4.18 Tampilan Hasil Data Training

Tabel 4.1 Hasil Pelatihan Data Penyakit dan Gejala

Nama Penyakit	Pola Training
Virus Kuning	1100000000000000000000000000000000000000
Virus Mosaik Keriting	0011100000000000000000000000000000000000
Layu Fusarium	0000011000000000000000000000000000000000
Layu Lendir	0000000011100000000000001000000000000000
Hawar Daun	0000000000111100000000000000000000000000
Kapang Daun	0000000000000011100000000000000000000000
Busuk Daun	0000000000000000111110000000000000000000
Bercak Daun Septoria	0000000000100000000000110000000000000000
Bercak Coklat	0000000000000000000000001111100000000000
Busuk Buah Rhizoctonia	0000000000000000000000000000111000000000
Busuk Buah Antraknosa	000000100000000000000000000000111100000
Bercak Bakteri	00000000000000000000000000000000000011111

4.6 Analisa Hasil

Setelah mendapatkan arsitektur jaringan yang optimal, kemudian dilanjutkan dengan pengujian terhadap data yang sudah pernah dilakukan pembelajaran. Terdapat 12 data pengujian yang sudah dilakukan pembelajaran sebelumnya. Bobot yang dipakai untuk pengujian adalah bobot dari hasil pembelajaran dengan jumlah unit pada *hidden layer* sebesar 100 unit, nilai *learning rate* sebesar 0.8, nilai *error* sebesar 0.01, dan *max epoch* sebesar 1000 (yang sudah ditentukan sebelumnya). Hasil pengujian terhadap data yang sudah pernah dilakukan pembelajaran dapat dilihat dalam tabel 4.2.

Tabel 4.2 Hasil Pengujian pada Data yang telah dilakukan Pembelajaran

No	Id_User	Hasil Diagnosa Sistem	Keterangan
1	1	Virus Kuning	2 gejala terpenuhi; tidak ada gejala tambahan
2	2	Virus Mosaik Keriting	3 gejala terpenuhi; tidak ada gejala tambahan
3	3	Layu Fusarium	2 gejala terpenuhi dengan 1 gejala tidak terpenuhi
4	4	Hawar Daun	4 gejala terpenuhi; dengan 7 gejala tambahan
5	5	Busuk Daun	5 gejala terpenuhi; tidak ada gejala tambahan
6	6	Busuk Buah Antraknosa	4 gejala terpenuhi; tidak ada gejala tambahan
7	7	Bercak Bakteri	5 gejala terpenuhi; tidak ada gejala tambahan
8	8	Busuk Buah Rhizoctonia	3 gejala terpenuhi; tidak ada gejala tambahan

Hasil diagnosa didapatkan dari kesesuaian antara data pelatihan (yang sudah dilatih terlebih dahulu) dengan masukan gejala penyakit.

Berdasarkan hasil pengujian data yang sudah dilakukan pelatihan sebelumnya, sistem mampu mengenali data yang sudah diujikan (tingkat keakuratan sebesar 99,9985 %).

Berikut ini adalah hasil perhitungan rata-rata tingkat keakuratan berdasarkan hasil data yang telah diuji.

- Virus Kuning : 100%
- Virus Mosaik Keriting : 100%
- Layu Fusarium : 99,990%
- Hawar Daun : 100%
- Busuk Daun : 100%
- Busuk Buah Antraknosa : 99,998%
- Bercak Bakteri : 100%
- Busuk Buah Rhizoctonia : 100%

Untuk menghitung nilai keakuratan secara keseluruhan, maka dihitung nilai rata-ratanya.

$$\text{Rata-rata} = \frac{100\% + 100\% + 99,990\% + 100\% + 100\% + 99,998\% + 100\% + 100\%}{8} = 99,9985\%$$

Berikut adalah penyakit yang belum diuji datanya, namun telah dilakukan pembelajaran. Terdapat 4 penyakit, dapat dilihat pada tabel 4.3.

Tabel 4.3 Data Penyakit yang Belum Diuji Datanya

Nama Penyakit	Syarat
Layu Lendir	3 gejala terpenuhi
Kapang Daun	3 gejala terpenuhi
Bercak Coklat	5 gejala terpenuhi
Busuk Daun	5 gejala terpenuhi

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil percobaan dan pembahasan dapat disimpulkan beberapa hal sebagai berikut:

1. Dihasilkan perangkat lunak untuk mendiagnosa jenis penyakit pada tanaman tomat yang dibangun menggunakan algoritma jaringan saraf tiruan *backpropagation* yang sudah dilatih mampu memberikan diagnosa penyakit dengan cukup baik. Pernyataan tersebut dapat dibuktikan pada hasil pengujian untuk data yang pernah dilakukan pembelajaran (Tabel 4.1).
2. Adanya solusi dari seluruh penyakit pada tanaman tomat, yang terdapat pada halaman *web*, yang dapat memberikan informasi tambahan bagi para pengguna aplikasi.
3. Dari hasil penelitian, arsitektur Jaringan Saraf Tiruan terbaik yang berhasil disusun adalah jaringan saraf *backpropagation* yang terdiri atas lapisan *input* dengan 42 *neuron*, satu lapisan tersembunyi dengan 100 *neuron* dan lapisan *output* dengan 12 *neuron*, dan nilai *error* sebesar 0.01. Dimana nilai *learning rate* adalah 0,8 dan maksimal epoch sebesar 1000 (yang sudah ditentukan sebelumnya). Dapat dibuktikan pada hasil pengujian, yang menghasilkan tingkat keakuratan sistem dalam memberikan diagnosa penyakit sebesar 99,9999 % untuk data yang pernah dilakukan pembelajaran (Tabel 4.2).
4. Dalam Jaringan Saraf Tiruan, diperlukan adanya pemilihan konfigurasi jaringan (jumlah *neuron hidden* dan nilai *learning rate*) yang tepat untuk mempercepat proses pelatihan.

5.2 Saran

Adapun saran penulis untuk pengembangan penelitian di bidang yang sama adalah:

1. Penggunaan data pelatihan yang semakin banyak akan memberikan keakuratan jaringan terhadap data uji. Pada penelitian ini masih membutuhkan data training dari masing-masing penyakit yang lebih diperbanyak lagi sehingga dapat mewakili karakteristik data yang akan diujikan.
2. Uji coba aplikasi dibandingkan dengan aplikasi sejenis dengan algoritma atau teknik-teknik klasifikasi data yang lain (misalnya pohon keputusan, *bayesian belief network* dan lain lain) dengan data yang sama sehingga dapat diketahui perbandingan kinerja satu sama lain.
3. Aplikasi dapat dikembangkan lebih lanjut. Dimana jumlah masukan yang berupa variabel pemeriksaan dan keluaran penyakit lebih diperbanyak. Dikarenakan penelitian di bidang kesehatan khususnya penyakit mata pada manusia akan terus bertambah seiring dengan perkembangan zaman dan teknologi.

DAFTAR PUSTAKA

- [1] Kusumadewi, *Artificial Intelligence*, Graha Ilmu, Yogyakarta, 2003
- [2] Puspitaningrum, *Pengantar Jaringan Saraf Tiruan*, Penerbit Andi, Yogyakarta, 2006.
- [3] Siang, 2005, *Jaringan Saraf Tiruan dengan Pemrogramannya Menggunakan Matlab*, Andi, Yogyakarta.
- [4] A.Ciancio, K.G. Mukerji, 2008, *Integrated Management of Diseases Caused by Fungi, Phytoplasma and Bacteria*, University of Delhi, India.
- [5] Henryk Czosnek, 2007, *Tomato Yellow Leaf Curl Virus Disease*, The Hebrew University of Jerusalem, Israel.
- [6] J. Benton Jones, Jr, 2008, *Tomato Plant Culture*, CRC Press, United States of America.
- [7] Divisi Penelitian dan Pengembangan, 2003, *Programan Borland Delphi 7*, Andi, Yogyakarta.
- [8] S, Husni, 2003, *Pemrograman Database dengan Delphi*, Graha Ilmu, Yogyakarta.
- [9] Kadir, Abdul., 2001, "*Pemrograman Database Menggunakan Delphi Jilid 1*", Salemba Infotek, Jakarta.
- [10] Wikipedia, MySQL, <http://id.wikipedia.org/wiki/MySQL>. Tanggal akses : 20 Juli 2011.
- [11] Penyakit Tumbuhan Akibat Virus, <http://www.docstoc.com/docs/33799231/Penyakit-tumbuhan-akibat-virus>. Tanggal akses :: 23 Mei 2011.
- [12] Tomato Troubles, http://gardener.wikiia.com/wiki/Tomato_troubles, tanggal akses :: 23 Mei 2011.
- [13] Penyakit Tomat, <http://www.scribd.com/doc/33648869/1126>, tanggal akses :: 20 Juli 2011.
- [14] Purnomo, H. dan Agus, K, 2006. *Supervised Neural Networks dan Aplikasinya*. Graha Ilmu : Yogyakarta.

LAMPIRAN



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

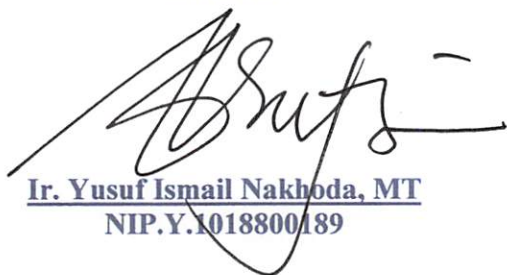
NAMA : ROSEVINE HUTABARAT
NIM : 07.12.569
JURUSAN : Teknik Komputer dan Informatika S-1
JUDUL : **APLIKASI JARINGAN SARAF TIRUAN BERBASIS WEB
UNTUK DIAGNOSA PENYAKIT PADA TANAMAN TOMAT**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

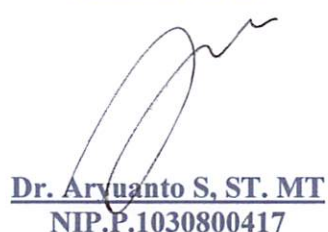
Pada Hari : Sabtu
Tanggal : 13 Agustus 2011
Dengan Nilai : 89,95 (A) *or*

PANITIA UJIAN SKRIPSI

KETUA


Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

SEKRETARIS

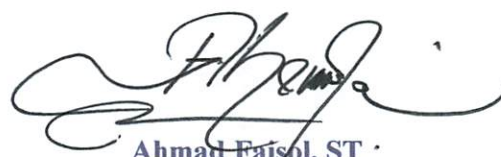

Dr. Aryanto S, ST, MT
NIP.P.1030800417

ANGGOTA PENGUJI

PENGUJI I


M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

PENGUJI II


Ahmad Faisol, ST
NIP.P. 1031000431



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : ROSEVINE HUTABARAT
NIM : 07.12.569
JURUSAN : Teknik Komputer dan Informatika S-1
JUDUL : APLIKASI JARINGAN SARAF TIRUAN BERBASIS WEB
UNTUK DIAGNOSA PENYAKIT PADA TANAMAN TOMAT

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	13 Agustus 2011	1. Tambahkan Keterangan Gambar dan Tabel.	
2.	Penguji II	13 Agustus 2011	1. Hilangkan kolom solusi pada aplikasi desktop. 2. Cek jawaban pertanyaan sudah dijawab lengkap/belum, kalau belum tidak bisa menyimpan. 3. Perbaiki tampilan hasil diagnosa, jangan terlalu sederhana. 4. Tambahkan presentasi kebenaran.	

Disetujui :

Dosen Penguji I

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

Dosen Penguji II

Ahmad Faisal, ST
NIP.P. 1031000431

Mengetahui :

Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Dosen Pembimbing II

Dr. Eng. Aryunto Soetedjo, ST. MT
NIP.P.1030800417



FORMULIR BIMBINGAN SKRIPSI

Nama : ROSEVINE HUTABARAT
Nim : 07.12.569
Masa Bimbingan : 2 April 2011 s/d 2 September 2011
Judul Skripsi : APLIKASI JARINGAN SARAF TIRUAN BERBASIS *WEB* UNTUK
DIAGNOSA PENYAKIT PADA TANAMAN TOMAT

NO.	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	20-7-11	<ul style="list-style-type: none">Demo Program	
2.	24-7-11	<ul style="list-style-type: none">Revisi program "desain web kurang menarik, perhitungan error"	
3.	28-7-11	<ul style="list-style-type: none">Revisi program "desain web kurang lengkap, header, menu bar dibuat lebih menarik"	
4.	31-8-11	<ul style="list-style-type: none">Revisi program "tambahkan identitas pada web"	
5.	2-8-11	<ul style="list-style-type: none">ACC program	
6.	4-8-11	<ul style="list-style-type: none">Revisi makalah seminar hasil "jarak gambar dan keterangan terlalu dekat, gambar kurang jelas"	
7.	5-8-11	<ul style="list-style-type: none">ACC Makalah Seminar Hasil	
8.	29-9-11	<ul style="list-style-type: none">ACC Laporan	

Malang,

Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189



FORMULIR BIMBINGAN SKRIPSI

Nama : Rosevine Hutabarat
Nim : 07.12.569
Masa Bimbingan : 2 April 2011 s/d 2 September 2011 *BY*
Judul Skripsi : Aplikasi Jaringan Saraf Tiruan Berbasis *Web* untuk Diagnosa Penyakit pada Tanaman Tomat

No	Tanggal	Uraian	Paraf Pembimbing
1	29/4/11	Bab I (Pendahuluan)	
2	9/7 '11	Bab II (Tinjauan Pustaka)	
3	15/7 '11	Demo Program (Revisi hitungan Backpropagation)	
4	17/7 '11	Revisi Program: "proses pelatihan & error"	
5	19/7 '11	Revisi Program: "proses pelatihan & error"	
6	23/7 '11	Bab III (Perancangan Sistem)	
7	28/7 '11	Bab IV & Bab V (Implementasi & Kesimpulan)	
8	29/9 '11	ACC Laporan	
9			
10			

Malang,
Dosen Pembimbing II

Dr. Eng. Aryuanto Soetedio, ST, MT
NIP. P. 1030800417

- *Listing program untuk form menu utama*

```
object FUTama: TFUtama
    Left = 148
    Top = 72
    Width = 1036
    Height = 588
    Caption = 'FUtama'
    Color = clBtnFace
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -11
    Font.Name = 'MS Sans Serif'
    Font.Style = []
    Menu = MainMenuI
    OldCreateOrder = False
    Position = poDesktopCenter
    WindowState = wsMaximized
    OnClose = FormClose
    OnShow = FormShow
    PixelsPerInch = 96
    TextHeight = 13
object sPanel1: TsPanel
    Left = 16
    Top = 8
    Width = 1329
    Height = 41
    TabOrder = 0
    SkinData.SkinSection = 'PANEL'
object sLabelFX5: TsLabelFX
    Left = 15
    Top = 2
    Width = 337
```

```
Height = 37

Caption = 'APLIKASI DETEKSI PENYAKIT TOMAT '

ParentFont = False

WordWrap = True

Font.Charset = ANSI_CHARSET

Font.Color = clSilver

Font.Height = -24

Font.Name = 'Impact'

Font.Style = []

end

end

object sPanel2: TsPanel

Left = 16

Top = 56

Width = 1329

Height = 649

TabOrder = 1

SkinData.SkinSection = 'PANEL'

object sPanel3: TsPanel

Left = 16

Top = 16

Width = 185

Height = 617

BevelOuter = bvNone

TabOrder = 0

SkinData.SkinSection = 'PANEL'

object sLabelFX1: TsLabelFX

Left = 16

Top = 16

Width = 53

Height = 21

Caption = 'MASTER'

ParentFont = False
```

```
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clSilver
    Font.Height = -11
    Font.Name = 'MS Sans Serif'
    Font.Style = []
end
object sLabelFX2: TsLabelFX
    Left = 16
    Top = 144
    Width = 108
    Height = 21
    Caption = 'MASTER TRAINING'
    ParentFont = False
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clSilver
    Font.Height = -11
    Font.Name = 'MS Sans Serif'
    Font.Style = []
end
object sLabelFX3: TsLabelFX
    Left = 16
    Top = 272
    Width = 84
    Height = 21
    Caption = 'TES TRAINING'
    ParentFont = False
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clSilver
    Font.Height = -11
    Font.Name = 'MS Sans Serif'
    Font.Style = []
end
object sLabelFX4: TsLabelFX
```

```
Left = 16
Top = 408
Width = 88
Height = 21
Caption = 'SETTING USER'
ParentFont = False
Font.Charset = DEFAULT_CHARSET
Font.Color = clSilver
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
end
object sBitBtn1: TsBitBtn
Left = 16
Top = 40
Width = 153
Height = 33
Caption = 'MASTER PENYAKIT'
TabOrder = 0
OnClick = sBitBtn1Click
SkinData.SkinSection = 'BUTTON'
end
object sBitBtn2: TsBitBtn
Left = 16
Top = 80
Width = 153
Height = 33
Caption = 'MASTER GEJALA'
TabOrder = 1
OnClick = sBitBtn2Click
SkinData.SkinSection = 'BUTTON'
end
object sBitBtn4: TsBitBtn
```

```
Left = 16
Top = 168
Width = 153
Height = 33
Caption = 'PENGISIAN DATA TRAINING'
TabOrder = 2
OnClick = sBitBtn4Click
SkinData.SkinSection = 'BUTTON'
end
object sBitBtn5: TsBitBtn
Left = 16
Top = 208
Width = 153
Height = 33
Caption = 'DATA TRAINING'
TabOrder = 3
OnClick = sBitBtn5Click
SkinData.SkinSection = 'BUTTON'
end
object sBitBtn6: TsBitBtn
Left = 16
Top = 296
Width = 153
Height = 33
Caption = 'PROSES PELATIHAN'
TabOrder = 4
OnClick = sBitBtn6Click
SkinData.SkinSection = 'BUTTON'
end
object sBitBtn3: TsBitBtn
Left = 16
Top = 432
Width = 153
```

```
    Height = 33
    Caption = 'MASTER USER'
    TabOrder = 5
    OnClick = sBitBtn3Click
    SkinData.SkinSection = 'BUTTON'
end
object sBitBtn7: TsBitBtn
    Left = 16
    Top = 336
    Width = 153
    Height = 33
    Caption = 'DATA PELATIHAN'
    TabOrder = 6
    OnClick = sBitBtn7Click
    SkinData.SkinSection = 'BUTTON'
end
end
object sPanel4: TsPanel
    Left = 208
    Top = 16
    Width = 1105
    Height = 617
    TabOrder = 1
    SkinData.SkinSection = 'PANEL'
end
end
object MainMenu1: TMainMenu
    Left = 264
    Top = 80
    object Tools: TMenuItem
        Caption = 'Tools'
    object Help: TMenuItem
        Caption = 'Help'
```

```

end

object Exit1: TMenuItem
    Caption = 'Exit'
    OnClick = Exit1Click
end

end

end

object sSkinProvider1: TsSkinProvider
    AddedTitle.Font.Charset = DEFAULT_CHARSET
    AddedTitle.Font.Color = clNone
    AddedTitle.Font.Height = -11
    AddedTitle.Font.Name = 'MS Sans Serif'
    AddedTitle.Font.Style = []
    SkinData.SkinSection = 'FORM'
    TitleButtons = <>
    Left = 368
    Top = 192
end

object q1: TADOQuery
    Connection = dm.con
    Parameters = <>
    Left = 280
    Top = 20
end

end
end

```

- *Listing program untuk form data pelatihan*

```

object FrDataPelatihan: TFrDataPelatihan
    Left = 0
    Top = 0
    Width = 320
    Height = 240
    TabOrder = 0

```



```
object sPanel1: TsPanel

    Left = 0

    Top = 0

    Width = 320

    Height = 240

    Align = alClient

    TabOrder = 0

    SkinData.SkinSection = 'PANEL'

    object DBGrid1: TDBGrid

        Left = 8

        Top = 24

        Width = 1321

        Height = 665

        DataSource = DataSource1

        Options = [dgTitles, dgColumnResize, dgColLines, dgTabs,
dgRowSelect, dgConfirmDelete, dgCancelOnExit]

        TabOrder = 0

        TitleFont.Charset = DEFAULT_CHARSET

        TitleFont.Color = clWindowText

        TitleFont.Height = -11

        TitleFont.Name = 'MS Sans Serif'

        TitleFont.Style = []

        Columns = <

            item

                Expanded = False

                FieldName = 'kodeTes'

                Title.Alignment = taCenter

                Title.Caption = 'KODE TES'

                Visible = True

            end

            item

                Expanded = False

                FieldName = 'nama'
```

```
Title.Alignment = taCenter
Title.Caption = 'NAMA'
Visible = True
end
item
Expanded = False
FieldName = 'hasilPenyakit'
Title.Alignment = taCenter
Title.Caption = 'HASIL PENYAKIT'
Visible = True
end>
end
end
object qDataPelatihan: TADOQuery
Active = True
Connection = dm.con
CursorType = ctStatic
Parameters = <>
SQL.Strings = (
    'select * from tbTes')
Left = 152
Top = 8
end
object DataSource1: TDataSource
DataSet = qDataPelatihan
Left = 112
Top = 16
end
end
```

- *Listing program untuk form perhitungan backpropagation*

```
procedure TFormUjian.SpeedButton1Click(Sender: TObject);
var i:integer;
begin
    cari.Left :=2;
    con.close;
    con.sql.clear;
    con.sql.add('select * from pasien ');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
        begin
            for i := 1 to con.RecordCount do
                begin
                    with data do
                        begin
                            Cells[0,i] := con.fieldByName('kode_pasien').AsString;
                            Cells[1,i] := con.fieldByName('nama').AsString;
                            Cells[2,i] :=
con.fieldByName('alamat').AsString;
                            Cells[3,i] := con.fieldByName('jk').AsString;
                            RowCount :=i+1;
                            con.Next;
                        end;
                    end;
                SpeedButton1.Hide;
                jadiCari.Hide;
            end;
        end;
    procedure TFormUjian.batalCariClick(Sender: TObject);
    begin
```

```

cari.Left := -1200;

SpeedButton1.Show;

pasien.Clear;

end;

procedure TFormUjian.SpeedButton2Click(Sender: TObject);

begin
    if FormMain.fpasien =nil then
        FormMain.fpasien := TFormPasien.Create(Application)
    else
        FormMain.fpasien.Show ;
end;

procedure TFormUjian.FormCreate(Sender: TObject);

var i:integer;

begin
    with data do
        begin
            ColCount := 4;

            Cells[0,0] := 'No.';

            Cells[1,0] := 'Nama';

            Cells[2,0] := 'Alamat';

            Cells[3,0] := 'J.K';

            ColWidths[1] := 100;

            ColWidths[2] := 200;

        end;

        con.close;

        con.sql.clear;

        con.sql.add('select * from penyakit order by kode_penyakit asc ');

        con.Prepared;

        con.open;

```

```

TPenyakit := TStringList.Create;

if con.recordcount >0 then
begin
    for i := 1 to con.RecordCount do
        begin
            Penyakit[i] := con.fieldByName('nama').AsString;
            TPenyakit.Add(' '+con.fieldByName('nama').AsString);
            con.Next;
        end
    end;

con.sql.clear;

con.sql.add('SELECT * FROM `gejala` order by kode_gejala asc ');
con.Prepared;
con.open;

if con.recordcount >0 then
begin
    for i := 1 to con.RecordCount do
        begin
            Gejala.Items.Add(' '+con.fieldByName('nama').AsString);
            con.Next;
        end
    end;
end;

end;

procedure TFormUjian.dataSelectCell(Sender: TObject; ACol, ARow:
Integer;
    var CanSelect: Boolean);
begin
    pasien.Text := data.Cells[0,arow];
    jadiCari.Show;
end;

```

```

procedure TFormUjian.jadiCariClick(Sender: TObject);
begin
    cari.Left := -1200;
    SpeedButton1.Show;
end;

procedure TFormUjian.GejalaClickCheck(Sender: TObject);
var i,jml : integer;
begin
    jml := 0;
    for i:=0 to Gejala.Items.Count -1 do
    begin
        if Gejala.Checked [i]=true then
            inc(jml);
        end;
    proses.Visible := (jml>0) and (pasien.Text<>'');
end;

procedure TFormUjian.pasienChange(Sender: TObject);
begin
    GejalaClickCheck(sender);
end;

procedure TFormUjian.getBobot();
const
    brl
    ='.....';
    .....';
var i,ii,jj : integer; tmp : string;
begin

```

```

con.close;

con.sql.clear;

con.sql.add('select * from `v` ');

con.Prepared;

con.open;

if con.recordcount >0 then

begin

    for i := 1 to con.RecordCount do

        begin

            tmp := trim( con.fieldByName('kode').AsString);

            ii := StrToInt(LeftStr(tmp,pos( '-',tmp)-1));

            jj := StrToInt(copy(tmp,pos( '-',tmp)+1,2));

            v_min[jj,ii] := con.fieldByName('nilai').AsFloat;

            con.Next;

        end

    end;

con.sql.clear;

con.sql.add('select * from `w` ');

con.Prepared;

con.open;

if con.recordcount >0 then

begin

    for i := 1 to con.RecordCount do

        begin

            tmp := trim( con.fieldByName('kode').AsString);

            ii := StrToInt(LeftStr(tmp,pos( '-',tmp)-1));

            jj := StrToInt(copy(tmp,pos( '-',tmp)+1,2));

            w_min[jj,ii] := con.fieldByName('nilai').AsFloat;

            con.Next;

        end

    end;

end;

```

```

con.sql.clear;

con.sql.add('select * from `parameter` ');

con.Prepared;

con.open;

if con.recordcount >0 then

begin

    for i := 1 to con.RecordCount do

        begin

            tmp := trim( con.fieldByName('kode').AsString);

            if (tmp='hiden') then

                JumlahHiddenUnit:= con.fieldByName('nilai').AsInteger;

            if (tmp='maxoutput') then

                MaxOutput:= con.fieldByName('nilai').AsInteger;

            if (tmp='maxinput') then

                MaxInput:= con.fieldByName('nilai').AsInteger;

            con.Next;

        end;

    end;

end;

procedure TFormUjian.prosesClick(Sender: TObject);

const

    brl
    ='.....
    .....
    .....';

label outs;

var i,j,k : integer;

temp,temp2 : Extended;

str_,str2 :string;

jk :string;

h,m,s,d,b,ts,ms : word;

starts,ends : Tdatetime;

```



```

t1,t2,t3 : integer;

begin

  starts := Now;

  initParam;

  getBobot;

  logs.Lines.Add(brl);

  logs.Lines.Add('                                DATA PENGUJIAN' );

  logs.Lines.Add(brl);

  logs.Lines.Add('DATA GEJALA PASIEN :' );

  logs.Lines.Add(brl);

  for i:= 0 to Gejala.Items.Count -1 do

  begin

    if Gejala.Checked[i] then

    begin

      X[i+1] := 1;

      logs.Lines.Add(' -'+Gejala.Items.ValueFromIndex[I]);

    else X[i+1] := 0;

  end;

  for j:=1 to JumlahHiddenUnit do begin

    z_in[j] :=v_min[0,j];

    for i:= 1 to JumlahInputUnit do begin

      z_in[j]:=z_in[j]+x[i]*v_min[i,j];

    end;

    z[j] := fx(z_in[j]);

  end;

  for k:=1 to JumlahOutputUnit do begin

    y_in[k] :=w_min[0,k];

    for j:=1 to JumlahHiddenUnit do begin

      y_in[k]:=y_in[k]+z[j]*w_min[j,k];

    end;

  end;

```

```

end;

str2 := '';

for i:=1 to JumlahOutputUnit do
begin
    y_in[i] := y_in[i]*MaxOutput;
    str(y_in[i]:8:0, str_);

    RemoveSpace(str_);

    str2 := str2 + ' '+str_ ;

end;

{ logs.Lines.Add(brl);

logs.Lines.Add('DATA TARGET HASIL' );

logs.Lines.Add(brl);

logs.Lines.Add(' T: ' +str2);}

str2 := '';

for i := 1 to JumlahOutputUnit do
begin
    str(y_in[i]:8:0, str_);

    if trim(str_)='1' then
        begin
            kode.Caption := IntToStr(i) ;

            con.Close ;

            con.SQL.Clear;

            con.sql.add('select * from penyakit where `kode_penyakit`='+
IntToStr(i) +' order by kode_penyakit asc ');

            con.Prepared;

            con.open;

            if con.recordcount >0 then
                begin
                    str2 :=str2+#13#10+'Penyakit :
'+con.fieldByName('nama').AsString+#13#10+'Solusi:
'+#13#10+con.fieldByName('solusi').AsString;

                end;

```

```

        end;

    end;

    ends := Now-starts;

    DecodeDateTime(ends,ts,b,d,h,m,s,ms);

    logs.Lines.Add(brl);

    logs.Lines.Add('Start Time : ' + TimeToStr(starts) );

    logs.Lines.Add('Duration : ' +
IntToStr(h)+' ':'+IntToStr(m)+' ':'+IntToStr(s)+' ':'+IntToStr(ms) );

    simpan.Visible:= true;

    if Trim(str2)='' then
    begin
        str2:= 'Penyakit Tidak Dikenali';

        simpan.Visible := false;

    else str2:=str2;

    logs.Lines.Add(format('Hasil Pengujian : %s ',[str2]) );

    logs.Lines.Add(brl);
end;

procedure TFormUjian.FormClose(Sender: TObject; var Action:
TCloseAction);

begin
    FormMain.fuji :=nil;

    FormMain.fuji.Free;

    Action := caFree;
end;

procedure TFormUjian.simpanClick(Sender: TObject);

var i:integer;

begin
    for i:= 0 to Gejala.Items.Count -1 do
        begin

```

```

if Gejala.Checked[i] then

begin

    Gejala_[i+1] := 1;

    else Gejala_[i+1] := 0;

end;

if (MessageDlg('Yakin Untuk Simpan
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then

begin

    con.SQL.Clear;

    con.SQL.Add('INSERT INTO diagnosa');

    con.SQL.Add('(kode_pasien, kode_penyakit , G1, G2, G3, G4, G5,
G6, G7, G8, G9, G10, G11, G12,');

    con.SQL.Add('G13, G14, G15, G16, G17, G18, G19, G20, G21, G22,
G23, G24, G25,');

    con.SQL.Add('G26, G27, G28, G29, G30, G31, G32, G33, G34, G35,
G36, G37, G38,');

    con.SQL.Add('G39, G40, G41, G42, G43, G44, G45, G46, G47, G48,
G49, G50, G51,');

    con.SQL.Add('G52, G53, G54)');

    con.SQL.Add('VALUES ("'+pasien.Text +'",' + kode.caption+',
'+floatToStr(Gejala_[1])+', '+floatToStr(Gejala_[2])+',
'+floatToStr(Gejala_[3])

    +', '+floatToStr(Gejala_[4])+', '+floatToStr(Gejala_[5])

    +', '+floatToStr(Gejala_[6])+', '+floatToStr(Gejala_[7])+',
'+floatToStr(Gejala_[8])+', '+floatToStr(Gejala_[9])+',
'+floatToStr(Gejala_[10])

    +', '+floatToStr(Gejala_[11])+', '+floatToStr(Gejala_[12])+',
'+floatToStr(Gejala_[13])+', '+floatToStr(Gejala_[14])+',
'+floatToStr(Gejala_[15])

    +', '+floatToStr(Gejala_[16])+', '+floatToStr(Gejala_[17])+',
'+floatToStr(Gejala_[18])+', '+floatToStr(Gejala_[19])+',
'+floatToStr(Gejala_[20])

    +', '+floatToStr(Gejala_[21])+', '+floatToStr(Gejala_[22])+',
'+floatToStr(Gejala_[23])+', '+floatToStr(Gejala_[24])+',
'+floatToStr(Gejala_[25])

    +', '+floatToStr(Gejala_[26])+',');

    con.SQL.Add(''+floatToStr(Gejala_[27])+',
'+floatToStr(Gejala_[28])+', '+floatToStr(Gejala_[29])+',

```

```

'+floatToStr(Gejala_[30])
    +', '+floatToStr(Gejala_[31])
    +', '+floatToStr(Gejala_[32])+', '+floatToStr(Gejala_[33])+',
'+floatToStr(Gejala_[34])+', '+floatToStr(Gejala_[35])+',
'+floatToStr(Gejala_[36])
    +', '+floatToStr(Gejala_[37])
    +', '+floatToStr(Gejala_[38])+', '+floatToStr(Gejala_[39])+',
'+floatToStr(Gejala_[40])+', '+floatToStr(Gejala_[41])+',
'+floatToStr(Gejala_[42])
    +', '+floatToStr(Gejala_[43])+', '+floatToStr(Gejala_[44])+',
'+floatToStr(Gejala_[45])+', '+floatToStr(Gejala_[45])+',
'+floatToStr(Gejala_[47])
    +', '+floatToStr(Gejala_[48])+', '+floatToStr(Gejala_[49])+',
'+floatToStr(Gejala_[50])+', '+floatToStr(Gejala_[51])+',
'+floatToStr(Gejala_[52])
    +', '+floatToStr(Gejala_[53])+',
'+floatToStr(Gejala_[54])+')');

    con.ExecSQL;

    MessageDlg('Data Berhasil Disimpan.',mtInformation,[mbok],0);

    batalClick(self);

end;

end;

procedure TFormUjian.batalClick(Sender: TObject);

var i:integer;

begin
    proses.Hide;
    simpan.Hide;
    logs.Clear;
    for i:= 0 to Gejala.Items.Count-1 do
        Gejala.Checked[i] := false;
end;

end.

```

• Listing program untuk form web

```

<html>
<head>
<script src="css/clock.js" type="text/javascript"></script>
<script src="jquery-1.6.4.js" type="text/javascript"></script>

<link rel="stylesheet" href="css/style.css" type="text/css" />
<title>... Aplikasi Diagnosa Penyakit Tanaman Tomat ... </title>
<link href="/BP/images/tomato.png" rel='SHORTCUT ICON' />

</head>

<body onLoad="startclock()">
  <div id="container">
    <div id="wrapper">
      <div id="top">
        <!-- motto -->
          <ul class="motto">
            <li class="left">&nbsp;</li>
            <li>Aplikasi Diagnosa Penyakit
Tanaman Tomat</li>
          </ul>
          <!-- / motto -->
        </div> <!-- / top -->

        <!-- MENU -->
        <ul class="nav1">
          <li class="left"></li><li class="right">&nbsp;</li>
          <span class="preload1"></span>
          <span class="preload2"></span>

          <ul id="nav">
            <li class="top"><a href="media.php?inc=utama.php"
class="top_link"><span>Beranda </span></a></li>
            <li class="top"><a href="media.php?inc=inputan.php"
class="top_link"><span>Diagnosa</span></a></li>
            <li class="top"><a
href="media.php?inc=informasi.php"
class="top_link"><span>Informasi</span></a></li>
          </ul>
        </ul>
        <!-- / End Menu -->

        <!-- HEADER -->
        <div id="header">
          <div class="intro">
            <p><span id="date"></span>, <span
id="clock"></span></p> <!-- jam -->
          </div>
        </div> <!-- /header -->

        <div id="content">

          <script
type='text/javascript'>
            function autentikasi(){

```

```
        var
nilai_kode=document.forms["myform"]["kode"].value;
        var
nilai_nama=document.forms["myform"]["nama"].value;
        var
nilai_radio_yesGEJ001=document.getElementById("yes_radioGEJ001").
checked;
        var
nilai_radio_noGEJ001=document.getElementById("no_radioGEJ001").ch
ecked;
        var
nilai_radio_yesGEJ002=document.getElementById("yes_radioGEJ002").
checked;
        var
nilai_radio_noGEJ002=document.getElementById("no_radioGEJ002").ch
ecked;
        var
nilai_radio_yesGEJ003=document.getElementById("yes_radioGEJ003").
checked;
        var
nilai_radio_noGEJ003=document.getElementById("no_radioGEJ003").ch
ecked;
        var
nilai_radio_yesGEJ004=document.getElementById("yes_radioGEJ004").
checked;
        var
nilai_radio_noGEJ004=document.getElementById("no_radioGEJ004").ch
ecked;
        var
nilai_radio_yesGEJ005=document.getElementById("yes_radioGEJ005").
checked;
        var
nilai_radio_noGEJ005=document.getElementById("no_radioGEJ005").ch
ecked;
        var
nilai_radio_yesGEJ006=document.getElementById("yes_radioGEJ006").
checked;
        var
nilai_radio_noGEJ006=document.getElementById("no_radioGEJ006").ch
ecked;
        var
nilai_radio_yesGEJ007=document.getElementById("yes_radioGEJ007").
checked;
        var
nilai_radio_noGEJ007=document.getElementById("no_radioGEJ007").ch
ecked;
        var
nilai_radio_yesGEJ008=document.getElementById("yes_radioGEJ008").
checked;
        var
nilai_radio_noGEJ008=document.getElementById("no_radioGEJ008").ch
ecked;
        var
nilai_radio_yesGEJ009=document.getElementById("yes_radioGEJ009").
checked;
        var
nilai_radio_noGEJ009=document.getElementById("no_radioGEJ009").ch
ecked;
        var
```

```
nilai_radio_yesGEJ010=document.getElementById("yes_radioGEJ010").
checked;
                                var
nilai_radio_noGEJ010=document.getElementById("no_radioGEJ010").ch
echecked;
                                var
nilai_radio_yesGEJ011=document.getElementById("yes_radioGEJ011").
checked;
                                var
nilai_radio_noGEJ011=document.getElementById("no_radioGEJ011").ch
echecked;
                                var
nilai_radio_yesGEJ012=document.getElementById("yes_radioGEJ012").
checked;
                                var
nilai_radio_noGEJ012=document.getElementById("no_radioGEJ012").ch
echecked;
                                var
nilai_radio_yesGEJ013=document.getElementById("yes_radioGEJ013").
checked;
                                var
nilai_radio_noGEJ013=document.getElementById("no_radioGEJ013").ch
echecked;
                                var
nilai_radio_yesGEJ014=document.getElementById("yes_radioGEJ014").
checked;
                                var
nilai_radio_noGEJ014=document.getElementById("no_radioGEJ014").ch
echecked;
                                var
nilai_radio_yesGEJ015=document.getElementById("yes_radioGEJ015").
checked;
                                var
nilai_radio_noGEJ015=document.getElementById("no_radioGEJ015").ch
echecked;
                                var
nilai_radio_yesGEJ016=document.getElementById("yes_radioGEJ016").
checked;
                                var
nilai_radio_noGEJ016=document.getElementById("no_radioGEJ016").ch
echecked;
                                var
nilai_radio_yesGEJ017=document.getElementById("yes_radioGEJ017").
checked;
                                var
nilai_radio_noGEJ017=document.getElementById("no_radioGEJ017").ch
echecked;
                                var
nilai_radio_yesGEJ018=document.getElementById("yes_radioGEJ018").
checked;
                                var
nilai_radio_noGEJ018=document.getElementById("no_radioGEJ018").ch
echecked;
                                var
nilai_radio_yesGEJ019=document.getElementById("yes_radioGEJ019").
checked;
                                var
nilai_radio_noGEJ019=document.getElementById("no_radioGEJ019").ch
echecked;
```



```
var
nilai_radio_yesGEJ020=document.getElementById("yes_radioGEJ020").
checked;
var
nilai_radio_noGEJ020=document.getElementById("no_radioGEJ020").ch
ecked;
var
nilai_radio_yesGEJ021=document.getElementById("yes_radioGEJ021").
checked;
var
nilai_radio_noGEJ021=document.getElementById("no_radioGEJ021").ch
ecked;
var
nilai_radio_yesGEJ022=document.getElementById("yes_radioGEJ022").
checked;
var
nilai_radio_noGEJ022=document.getElementById("no_radioGEJ022").ch
ecked;
var
nilai_radio_yesGEJ023=document.getElementById("yes_radioGEJ023").
checked;
var
nilai_radio_noGEJ023=document.getElementById("no_radioGEJ023").ch
ecked;
var
nilai_radio_yesGEJ024=document.getElementById("yes_radioGEJ024").
checked;
var
nilai_radio_noGEJ024=document.getElementById("no_radioGEJ024").ch
ecked;
var
nilai_radio_yesGEJ025=document.getElementById("yes_radioGEJ025").
checked;
var
nilai_radio_noGEJ025=document.getElementById("no_radioGEJ025").ch
ecked;
var
nilai_radio_yesGEJ026=document.getElementById("yes_radioGEJ026").
checked;
var
nilai_radio_noGEJ026=document.getElementById("no_radioGEJ026").ch
ecked;
var
nilai_radio_yesGEJ027=document.getElementById("yes_radioGEJ027").
checked;
var
nilai_radio_noGEJ027=document.getElementById("no_radioGEJ027").ch
ecked;
var
nilai_radio_yesGEJ028=document.getElementById("yes_radioGEJ028").
checked;
var
nilai_radio_noGEJ028=document.getElementById("no_radioGEJ028").ch
ecked;
var
nilai_radio_yesGEJ029=document.getElementById("yes_radioGEJ029").
checked;
var
nilai_radio_noGEJ029=document.getElementById("no_radioGEJ029").ch
```

```
ecked;
                                var
nilai_radio_yesGEJ030=document.getElementById("yes_radioGEJ030").
checked;
                                var
nilai_radio_noGEJ030=document.getElementById("no_radioGEJ030").ch
ecked;
                                var
nilai_radio_yesGEJ031=document.getElementById("yes_radioGEJ031").
checked;
                                var
nilai_radio_noGEJ031=document.getElementById("no_radioGEJ031").ch
ecked;
                                var
nilai_radio_yesGEJ032=document.getElementById("yes_radioGEJ032").
checked;
                                var
nilai_radio_noGEJ032=document.getElementById("no_radioGEJ032").ch
ecked;
                                var
nilai_radio_yesGEJ033=document.getElementById("yes_radioGEJ033").
checked;
                                var
nilai_radio_noGEJ033=document.getElementById("no_radioGEJ033").ch
ecked;
                                var
nilai_radio_yesGEJ034=document.getElementById("yes_radioGEJ034").
checked;
                                var
nilai_radio_noGEJ034=document.getElementById("no_radioGEJ034").ch
ecked;
                                var
nilai_radio_yesGEJ035=document.getElementById("yes_radioGEJ035").
checked;
                                var
nilai_radio_noGEJ035=document.getElementById("no_radioGEJ035").ch
ecked;
                                var
nilai_radio_yesGEJ036=document.getElementById("yes_radioGEJ036").
checked;
                                var
nilai_radio_noGEJ036=document.getElementById("no_radioGEJ036").ch
ecked;
                                var
nilai_radio_yesGEJ037=document.getElementById("yes_radioGEJ037").
checked;
                                var
nilai_radio_noGEJ037=document.getElementById("no_radioGEJ037").ch
ecked;
                                var
nilai_radio_yesGEJ038=document.getElementById("yes_radioGEJ038").
checked;
                                var
nilai_radio_noGEJ038=document.getElementById("no_radioGEJ038").ch
ecked;
                                var
nilai_radio_yesGEJ039=document.getElementById("yes_radioGEJ039").
checked;
                                var
```

```

nilai_radio_noGEJ039=document.getElementById("no_radioGEJ039").checked;
                                var
nilai_radio_yesGEJ040=document.getElementById("yes_radioGEJ040").checked;
                                var
nilai_radio_noGEJ040=document.getElementById("no_radioGEJ040").checked;
                                var
nilai_radio_yesGEJ041=document.getElementById("yes_radioGEJ041").checked;
                                var
nilai_radio_noGEJ041=document.getElementById("no_radioGEJ041").checked;
                                var
nilai_radio_yesGEJ042=document.getElementById("yes_radioGEJ042").checked;
                                var
nilai_radio_noGEJ042=document.getElementById("no_radioGEJ042").checked;

                                if (nilai_kode==null || nilai_kode==""){
                                    alert("Masukkan Kode");
                                    return false;
                                    //location.href =
'media.php?inc=inputan.php';
                                }else if(nilai_nama==null ||
nilai_nama==""){
                                    alert("Masukkan Nama");
                                    return false;
                                }if(nilai_radio_yesGEJ001==" " ||
nilai_radio_noGEJ001==""){
                                    alert("Nilai Kode Gejala GEJ001
Belum Diisi");
                                    return false;
                                }if(nilai_radio_yesGEJ002==" " ||
nilai_radio_noGEJ002==""){
                                    alert("Nilai Kode Gejala GEJ002
Belum Diisi");
                                    return false;
                                }if(nilai_radio_yesGEJ003==" " ||
nilai_radio_noGEJ003==""){
                                    alert("Nilai Kode Gejala GEJ003
Belum Diisi");
                                    return false;
                                }if(nilai_radio_yesGEJ004==" " ||
nilai_radio_noGEJ004==""){
                                    alert("Nilai Kode Gejala GEJ004
Belum Diisi");
                                    return false;
                                }if(nilai_radio_yesGEJ005==" " ||
nilai_radio_noGEJ005==""){
                                    alert("Nilai Kode Gejala GEJ005
Belum Diisi");
                                    return false;
                                }if(nilai_radio_yesGEJ006==" " ||
nilai_radio_noGEJ006==""){
                                    alert("Nilai Kode Gejala GEJ006
Belum Diisi");

```

```
return false;
}if(nilai_radio_yesGEJ007=="") ||
nilai_radio_noGEJ007==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ008=="") ||
nilai_radio_noGEJ008==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ009=="") ||
nilai_radio_noGEJ009==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ010=="") ||
nilai_radio_noGEJ010==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ011=="") ||
nilai_radio_noGEJ011==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ012=="") ||
nilai_radio_noGEJ012==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ013=="") ||
nilai_radio_noGEJ013==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ014=="") ||
nilai_radio_noGEJ014==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ015=="") ||
nilai_radio_noGEJ015==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ016=="") ||
nilai_radio_noGEJ016==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ017=="") ||
nilai_radio_noGEJ017==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ018=="") ||
nilai_radio_noGEJ018==""){
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ018=="") ||
alert("Nilai Kode Gejala GEJ018
```

```

Belum Diisi");
return false;
}if(nilai_radio_yesGEJ019=="") ||
nilai_radio_noGEJ019==""){
alert("Nilai Kode Gejala GEJ019
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ020=="") ||
nilai_radio_noGEJ020==""){
alert("Nilai Kode Gejala GEJ020
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ021=="") ||
nilai_radio_noGEJ021==""){
alert("Nilai Kode Gejala GEJ021
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ022=="") ||
nilai_radio_noGEJ022==""){
alert("Nilai Kode Gejala GEJ022
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ023=="") ||
nilai_radio_noGEJ023==""){
alert("Nilai Kode Gejala GEJ023
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ024=="") ||
nilai_radio_noGEJ024==""){
alert("Nilai Kode Gejala GEJ024
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ025=="") ||
nilai_radio_noGEJ025==""){
alert("Nilai Kode Gejala GEJ025
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ026=="") ||
nilai_radio_noGEJ026==""){
alert("Nilai Kode Gejala GEJ026
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ027=="") ||
nilai_radio_noGEJ027==""){
alert("Nilai Kode Gejala GEJ027
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ028=="") ||
nilai_radio_noGEJ028==""){
alert("Nilai Kode Gejala GEJ028
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ029=="") ||
nilai_radio_noGEJ029==""){
alert("Nilai Kode Gejala GEJ029
Belum Diisi");
return false;
}if(nilai_radio_yesGEJ030=="") ||
nilai_radio_noGEJ030==""){

```



```

nilai_radio_noGEJ042==""){
    alert("Nilai Kode Gejala GEJ042
Belum Diisi");
    return false;
}else{
    alert("Data Berhasil Disimpan");
}
}
</script>
<form name="myform" method="post"
action="media.php?inc=simpan.php" onsubmit='return
autentikasi()''>
    <table cellpadding="5" cellspacing="5">
        <tr>
            <td>Kode</td>
            <td><input name="kode" id="kode"
value="10"/></td>
        </tr>
        <tr>
            <td>Nama</td>
            <td><input name="nama" id="nama" value="" /></td>
        </tr>
    </table>

    <table border="0" cellpadding="10" cellspacing="10">
        <tr>
            <td align="center" colspan="2" width="100">GEJALA
<font style="font-family:Arial, Helvetica, sans-serif"
</font>>"</td>
            <td align="center" width="50px" >Y</td>
            <td align="center" width="50px" >N</td>
        </tr>
        <tr class='radiobtn'><td width='100'>GEJ001</td><td
width='400'>Daun menguning</td><td width='50px'
align='center'><input type='radio' name='radioGEJ001'
value='GEJ001' id='yes_radioGEJ001' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ001' value=''
id='no_radioGEJ001' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ002</td><td width='400'>Daun menggulung ke
atas</td><td width='50px' align='center'><input type='radio'
name='radioGEJ002' value='GEJ002' id='yes_radioGEJ002'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ002' value='' id='no_radioGEJ002' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ003</td><td width='400'>Daun
berwarna belang (antara hijau tua dan hijau muda)</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ003' value='GEJ003' id='yes_radioGEJ003'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ003' value='' id='no_radioGEJ003' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ004</td><td width='400'>Daun
berubah bentuk (keriting)</td><td width='50px'
align='center'><input type='radio' name='radioGEJ004'
value='GEJ004' id='yes_radioGEJ004' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ004' value=''
id='no_radioGEJ004' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ005</td><td width='400'>Daun berpola bercak-bercak
seperti mosaik</td><td width='50px' align='center'><input
type='radio' name='radioGEJ005' value='GEJ005'

```

```

id='yes_radioGEJ005' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ005' value=''
id='no_radioGEJ005' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ006</td><td width='400'>Tanaman layu</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ006' value='GEJ006' id='yes_radioGEJ006'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ006' value='' id='no_radioGEJ006' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ007</td><td width='400'>Daun
berubah warna menjadi kuning</td><td width='50px'
align='center'><input type='radio' name='radioGEJ007'
value='GEJ007' id='yes_radioGEJ007' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ007' value=''
id='no_radioGEJ007' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ008</td><td width='400'>Akar akan mengalami
diskolorasi (berwarna coklat)</td><td width='50px'
align='center'><input type='radio' name='radioGEJ008'
value='GEJ008' id='yes_radioGEJ008' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ008' value=''
id='no_radioGEJ008' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ009</td><td width='400'>Daun layu</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ009' value='GEJ009' id='yes_radioGEJ009'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ009' value='' id='no_radioGEJ009' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ010</td><td
width='400'>Bagian bawah daun menguning</td><td width='50px'
align='center'><input type='radio' name='radioGEJ010'
value='GEJ010' id='yes_radioGEJ010' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ010' value=''
id='no_radioGEJ010' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ011</td><td width='400'>Daun menggulung ke
bawah</td><td width='50px' align='center'><input type='radio'
name='radioGEJ011' value='GEJ011' id='yes_radioGEJ011'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ011' value='' id='no_radioGEJ011' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ012</td><td
width='400'>Bercak berwarna hijau muda sampai kecoklatan pada
daun</td><td width='50px' align='center'><input type='radio'
name='radioGEJ012' value='GEJ012' id='yes_radioGEJ012'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ012' value='' id='no_radioGEJ012' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ013</td><td width='400'>Pada
permukaan bawah daun terdapat gejala busuk, yaitu berwarna abu-
abu keputihan, sampai menjadi be</td><td width='50px'
align='center'><input type='radio' name='radioGEJ013'
value='GEJ013' id='yes_radioGEJ013' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ013' value=''
id='no_radioGEJ013' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ014</td><td width='400'>Daun berubah warna menjadi
coklat</td><td width='50px' align='center'><input type='radio'
name='radioGEJ014' value='GEJ014' id='yes_radioGEJ014'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ014' value='' id='no_radioGEJ014' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ015</td><td
width='400'>Terlihat bercak gelap berminyak pada buah</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ015' value='GEJ015' id='yes_radioGEJ015'
/></td><td width='50px' align='center'><input type='radio'

```



```

name='radioGEJ015' value='' id='no_radioGEJ015' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ016</td><td
width='400'>Terdapat bercak coklat pada permukaan daun</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ016' value='GEJ016' id='yes_radioGEJ016'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ016' value='' id='no_radioGEJ016' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ017</td><td width='400'>Pada
bagian bawah daun terdapat spora-spora berwarna kelabu kekuning-
kuningan</td><td width='50px' align='center'><input type='radio'
name='radioGEJ017' value='GEJ017' id='yes_radioGEJ017'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ017' value='' id='no_radioGEJ017' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ018</td><td
width='400'>Menyerang pada daun-daun bagian bawah, hingga ke
seluruh tanaman</td><td width='50px' align='center'><input
type='radio' name='radioGEJ018' value='GEJ018'
id='yes_radioGEJ018' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ018' value=''
id='no_radioGEJ018' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ019</td><td width='400'>Bercak coklat kehitaman
pada daun hingga tangkai daun</td><td width='50px'
align='center'><input type='radio' name='radioGEJ019'
value='GEJ019' id='yes_radioGEJ019' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ019' value=''
id='no_radioGEJ019' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ020</td><td width='400'>Pada bagian bawah daun
terdapat jamur berwarna putih keunguan seperti beludru</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ020' value='GEJ020' id='yes_radioGEJ020'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ020' value='' id='no_radioGEJ020' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ021</td><td
width='400'>Bercak berair pada pangkal buah yang berwarna hijau
hingga kecoklatan</td><td width='50px' align='center'><input
type='radio' name='radioGEJ021' value='GEJ021'
id='yes_radioGEJ021' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ021' value=''
id='no_radioGEJ021' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ022</td><td width='400'>Pada buah terdapat bercak-
bercak berbentuk cincin yang berwarna coklat tua</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ022' value='GEJ022' id='yes_radioGEJ022'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ022' value='' id='no_radioGEJ022' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ023</td><td
width='400'>Bercak pada buah agak keras dan berkerut</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ023' value='GEJ023' id='yes_radioGEJ023'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ023' value='' id='no_radioGEJ023' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ024</td><td width='400'>Pada
permukaan bawah daun terdapat bercak bulat kecil berair (bercak
berwarna coklat muda)</td><td width='50px' align='center'><input
type='radio' name='radioGEJ024' value='GEJ024'
id='yes_radioGEJ024' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ024' value=''
id='no_radioGEJ024' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ025</td><td width='400'>Daun menggulung ke

```

```

bawah</td><td width='50px' align='center'><input type='radio'
name='radioGEJ025' value='GEJ025' id='yes_radioGEJ025'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ025' value='' id='no_radioGEJ025' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ026</td><td width='400'>Daun
terdapat bercak berwarna coklat sampai kehitaman</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ026' value='GEJ026' id='yes_radioGEJ026'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ026' value='' id='no_radioGEJ026' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ027</td><td
width='400'>Bercak pada daun akan membesar, hingga daun berubah
warna menjadi kuning, layu dan mati</td><td width='50px'
align='center'><input type='radio' name='radioGEJ027'
value='GEJ027' id='yes_radioGEJ027' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ027' value=''
id='no_radioGEJ027' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ028</td><td width='400'>Menyebabkan keguguran pada
bunga</td><td width='50px' align='center'><input type='radio'
name='radioGEJ028' value='GEJ028' id='yes_radioGEJ028'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ028' value='' id='no_radioGEJ028' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ029</td><td width='400'>Buah
tamat yang masih muda atau sudah matang menjadi busuk, berwarna
hitam, dan cekung</td><td width='50px' align='center'><input
type='radio' name='radioGEJ029' value='GEJ029'
id='yes_radioGEJ029' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ029' value=''
id='no_radioGEJ029' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ030</td><td width='400'>Menyerang pangkal buah
(ujung tangkai), yang menyebabkan perubahan warna menjadi coklat,
dan tertutu</td><td width='50px' align='center'><input
type='radio' name='radioGEJ030' value='GEJ030'
id='yes_radioGEJ030' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ030' value=''
id='no_radioGEJ030' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ031</td><td width='400'>Bercak cekung kecil
berwarna coklat pada buah</td><td width='50px'
align='center'><input type='radio' name='radioGEJ031'
value='GEJ031' id='yes_radioGEJ031' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ031' value=''
id='no_radioGEJ031' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ032</td><td width='400'>Batang tanaman
terpilin</td><td width='50px' align='center'><input type='radio'
name='radioGEJ032' value='GEJ032' id='yes_radioGEJ032'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ032' value='' id='no_radioGEJ032' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ033</td><td
width='400'>Pembusukan dengan warna coklat kemerahan pada pangkal
batang</td><td width='50px' align='center'><input type='radio'
name='radioGEJ033' value='GEJ033' id='yes_radioGEJ033'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ033' value='' id='no_radioGEJ033' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ034</td><td
width='400'>Bercak kecil berwarna coklat, bulat, dan berair pada
buah</td><td width='50px' align='center'><input type='radio'
name='radioGEJ034' value='GEJ034' id='yes_radioGEJ034'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ034' value='' id='no_radioGEJ034' /></td></tr><tr

```

```

class='radiobtn'><td width='100'>GEJ035</td><td
width='400'>Bercak ungu dekat tangkai buah</td><td width='50px'
align='center'><input type='radio' name='radioGEJ035'
value='GEJ035' id='yes_radioGEJ035' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ035' value=''
id='no_radioGEJ035' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ036</td><td width='400'>Daun menjadi layu</td><td
width='50px' align='center'><input type='radio'
name='radioGEJ036' value='GEJ036' id='yes_radioGEJ036'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ036' value='' id='no_radioGEJ036' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ037</td><td width='400'>Akar
akan mengalami diskolorasi (berwarna coklat)</td><td width='50px'
align='center'><input type='radio' name='radioGEJ037'
value='GEJ037' id='yes_radioGEJ037' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ037' value=''
id='no_radioGEJ037' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ038</td><td width='400'>Bercak kecil berair pada
batang dan daun</td><td width='50px' align='center'><input
type='radio' name='radioGEJ038' value='GEJ038'
id='yes_radioGEJ038' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ038' value=''
id='no_radioGEJ038' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ039</td><td width='400'>Daun menggulung (keriting)
ke bawah</td><td width='50px' align='center'><input type='radio'
name='radioGEJ039' value='GEJ039' id='yes_radioGEJ039'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ039' value='' id='no_radioGEJ039' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ040</td><td width='400'>Daun
kering</td><td width='50px' align='center'><input type='radio'
name='radioGEJ040' value='GEJ040' id='yes_radioGEJ040'
/></td><td width='50px' align='center'><input type='radio'
name='radioGEJ040' value='' id='no_radioGEJ040' /></td></tr><tr
class='radiobtn'><td width='100'>GEJ041</td><td
width='400'>Bercak berair, lalu berubah menjadi bercak seperti
gabus pada buah</td><td width='50px' align='center'><input
type='radio' name='radioGEJ041' value='GEJ041'
id='yes_radioGEJ041' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ041' value=''
id='no_radioGEJ041' /></td></tr><tr class='radiobtn'><td
width='100'>GEJ042</td><td width='400'>Terdapat luka seperti
kerak keabu-abuan pada batang</td><td width='50px'
align='center'><input type='radio' name='radioGEJ042'
value='GEJ042' id='yes_radioGEJ042' /></td><td width='50px'
align='center'><input type='radio' name='radioGEJ042' value=''
id='no_radioGEJ042' /></td></tr>
<tr>
<td colspan="4" align="right"><input
name="submit" type="submit" value="Simpan"/></td>
</tr>
</table>
</form>

</center>
</div>
<!-- FOOTER -->
<div id="footer">
<div class="foot_1"></div>
<div class="foot_content">
<p>&nbsp;</p>
<p>Copyright &copy; 2011

```

```
Rosevine Hutabarat. All rights reserved.</p>
    </div>
    <div class="foot_r"></div>
  </div><!-- / end footer -->
</div><!-- / end wrapper -->
</div><!-- / end container -->
</body>
</html>
```