BAB III

METODE PENELITIAN

Metode penelitian pada bab III ini akan dibagi menjadi beberapa subbab, yang masing-masing akan membahas berbagai sudut pandang mengenai sistim yang akan dibuat. Sudut pandang tersebut adalah pemakaian metode backpropagation tanpa activex, keadaan sistim yang akan dibuat atau diinginkan, langkah-langkah dalam pembuatan, dan distribusi. Langkah-langkah pembuatan sendiri dibagi menjadi dua, pertama, pembuatan property dan yang kedua pembuatan flowchart

3.1 Pemakaian Metode Backprogation Tanpa Activex

Aplikasi yang dibuat oleh pengembang perangkat lunak tidak luput dari kesalahan. Banyak faktor yang menyebabkan sebuah aplikasi mempunyai kesalahan pemrograman, atau sering disebut dengan *bug*.

Pada tingkat pengembang perangkat lunak kegagalan itu disebabkan oleh berbagai macam faktor: tenggat waktu penyelesaian, kemahiran pemrogram, pengalaman pemrogram, pengenalan akan masalah yang dihadapi dan lain sebagainya. Memang kesalahan dalam pembuatan aplikasi tidak dapat dihindari, namun dapat diminimalkan. Salah satu cara dengan membuat modul-modul aplikasi. Pembuatan ini dilakukan dengan membagi-bagi permasalahan menjadi bagian-bagian terkecil. Dengan adanya pembuatan modul tersebut, maka pencarian bug sebuah aplikasi dapat diawasi dengan lebih akurat karena daerah

STIKOM SURABAYA

pengawasan juga lebih kecil. Jika bagian yang diperiksa tersebut telah mencapai standar tertentu yang dianggap benar maka pemrogram boleh yakin bagian tersebut juga menghasilkan kerja yang benar apabila digabung dengan berbagai modul yang lebih besar.

Activex adalah sebuah modul program yang dibuat oleh microsoft dengan maksud nantinya akan ditempelkan pada sebuah aplikasi. Dengan adanya Activex ini maka pembuatan aplikasi dapat dipercepat. Pemrogram sendiri tidak perlu lagi membuat bagian program yang sudah diambil oleh Activex. Pemrogram hanya perlu berkonsentrasi memikirkan bagian yang lain seperti antar muka atau database saja.

3.2 Keadaan yang Diinginkan

Pada pembuatan Activex jaringan syaraf tiruan ini, diharapkan akan dicapai beberapa hal berikut ini:

1. Kecepatan dalam membuat aplikasi jaringan syaraf tiruan dengan metode backpropagation of error dibandingkan jika pemrogram harus mengetikkan program jaringan syaraf tiruan setiap kali harus membuat aplikasi.
2. Membantu memberitahukan kesalahan dalam memakai algoritma backpropagation of error
3. Mengurangi testing program untuk jaringan syaraf tiruan. Pemrogram hanya melakukan testing diluar algoritma jaringan syaraf tiruan.
4. Memperkecil kode program aplikasi jaringan syaraf tiruan.
5. Dapat melakukan penyimpanan setiap langkah perhitungan backpropagation of error, sehingga dapat melakukan evaluasi terhadap pemakaian nilai learning

STIKOM SURABAYA

rate dan momentum yang tepat sehingga dapat menghasilkan nilai output yang akurat.

Activex akan mulai bekerja pada saat property yang berhubungan dengan keaktifan Activex di manipulasi. Hal pertama yang dilakukan adalah memeriksa keabsahan masukan dari pemakai Activex. Bagian ini sangat penting karena masukan yang salah akan membuat proses pelatihan akan melakukan kesalahan juga.

Setelah masukan dinyatakan valid. Activex akan memanggil prosedur penerapan backpropagation. Proses berjalan pada setiap putaran atau *looping* pada setiap pengaktifan Activex, dengan demikian Activex mempunyai kesempatan merekam jalannya perhitungan setiap waktu.

Pada saat Activex selesai menghitung fungsi aktifasi node output, sistem akan berusaha mendeteksi terjadinya error pada sistem, kemudian akan menyimpan nilai error sistem, error node input ke hidden, error hidden ke output, error bias ke hidden dan error bias ke output. Setelah diketahui error masing-masing layer, dilakukan perhitungan perubahan bobot dan perubahan tersebut akan ditambahkan pada bobot lama pada masing-masing layer.

3.3 Langkah-Langkah Pembuatan

Langkah-langkah pembuatan Activex dibagi menjadi dua bagian besar, yaitu:

1. Pembuatan property

   Pembuatan property dilakukan dengan memperhatikan permasalahan jaringan syaraf metode backpropagation yang ada. Permasalahan tersebut adalah sebagai berikut:

STIKOM SURABAYA

- Kesulitan mencegah semua kesalahan yang dapat terjadi pada perhitungan jaringan syaraf tiruan
- Kesulitan implementasi Activex pada sebagaian besar konfigurasi komputer untuk dengan jumlah data besar dan nilai ketepatan bilangan desimal dengan digit di belakang koma yang besar
- Kesulitan dalam merekam setiap langkah perulangan
- Kesulitan menentukan nilai output yang tepat untuk berhenti pada suatu saat

2. Pembuatan Flowchart

   Pembuatan flowchart untuk landasan pembuatan kode program. Dengan adanya flowchart, dapat diketahui kemungkinan menyingkat pembuatan prosedur jika ditemukan adanya implementasi program yang sama dan memeriksa kebenaran penulisan kode program.

3.3.1 Pembuatan property

Dengan memperhatikan keadaan yang diinginkan, maka Activex yang akan dibuat akan mempunyai property-property sebagai berikut.

1. ErrorHidden

   Jumlah perhitungan error node hidden

2. ErrorOutput

   Jumlah perhitungan error node output

3. ErrorSistem

   Jumlah perhitungan error system

STIKOM SURABAYA

4. Hitung

   Memulai perhitungan backpropagation of error

5. InputAI

   Nilai tiap node pada node input

6. LearningRate

   Nilai learning rate

7. Momentum

   Nilai momentum

8. OutputAI

   Nilai target dari tiap node output

9. ResetAI

   Menghapus setiap nilai hasil p[illegible]n

10. StateLoop

    Jumlah loop pada saat [illegible]u

11. SetBobot

    Membuat bobot oto[illegible]pada awal perhitungan

12. BobotBi[illegible]

    Nilai [illegible]obot anta[illegible]a bias dan hidden baik karena hasil otomatis dari Activex atau nilai manual yang dimasukkan pemakai Activex

13. B[illegible]otBiasOutput

    Nilai bobot antara bias dan output baik karena hasil otomatis dari Activex atau nilai manual yang dimasukkan pemakai Activex

14. BobotHiddenOutput

STIKOMP SURABAYA

Nilai bobot antara Hidden dan Output baik karena hasil otomatis dari Activex atau nilai manual yang dimasukkan pemakai Activex

15. TambahBobotInputHidden

    Perubahan nilai bobot antara input dan hidden yang nantinya ditambahkan pada bobot lama karena adanya error pada perhitungan.

16. TambahBobotHiddenOutput

    Perubahan nilai bobot antara hidden dan output yang nantinya ditambahkan pada bobot lama karena adanya error pada perhitungan.

17. TambahBobotBiasHidden

    Perubahan nilai bobot antara bias dan hidden yang nantinya ditambahkan pada bobot lama karena adanya error pada perhitungan.

18. TambahBobotBiasOutput

    Perubahan nilai bobot antara bias dan output yang nantinya ditambahkan pada bobot lama karena adanya error pada perhitungan.

19. TotalHidden

    Jumlah seluruh node hidden pada layer hidden. Nilai ini akan diisi otomatis jika dikehendaki.

Activex yang akan dibuat terdiri dari beberapa property-property. Daftar property yang dipakai dapat dilihat pada tabel 3.1.

STIKOM SURABAYA

Tabel 3.1 Daftar Property Pada Activex yang Dibuat

Nama	Tipe	Default
AboutBox	Form	
Name	String	LindaAI
ErrorHidden	Double	Kosong
ErrorOutput	Double	Kosong
ErrorSistem	Double	0
Hitung	Pilihan: htMulai, htHenti	htH
InputAI	String	Koso
LearningRate	Double	
Momentum	Double	0
OutputAI	Double	Koson
ResetAI	Pilihan:rsFalse, rsTrue	
StateLoop	Integer	
SetBobot	Pilihan:rsSet, rsManua	rsManual
BobotBiasHidden	String	Kosong
BobotBiasOutput	String	Kosong
BobotHiddenOutput	String	Kosong
BobotInputHidden	String	Kosong

Property AboutBox digunakan untuk menampilkan kotak dialog hak cipta dari Activex. Gambar dari AboutBox dapat dilihat pada gambar berikut ini:

Property name adalah nama activex dalam pemrograman. Secara default akan bernama LindaAIx dengan x adalah nomor urutan pemunculannya pada form. Name tidak boleh kosong, jika kosong akan menampilkan pesan kesalahan seperti pada gambar 3.1.

Gambar 3.1 Kotak AboutBox

STIKOM SURABAYA

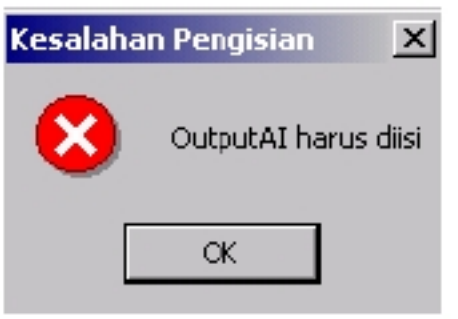
Property ErrorHidden dan ErrorOutput akan berisi daftar Error pada node hidden dan output. Contoh tampilan isi dari kedua property ini adalah sebagai berikut : -0.055, -0.057, -0.004. Pada saat Activex pertama kali dijalankan, ErrorHidden dan ErrorOutput akan di set ke kosong. Sementara Property ErrorSistem bertipe bilangan desimal dengan tipe double. Property ErrorSistem akan diisi dari Error sistem yang telah dihitung.

Property Hitung adalah property yang digunakan untuk memberitahukan kepada Activex kapan dimulainya perhitungan. Propety ini dijalankan jika property yang berhubungan dengan proses perhitungan telah terisi. Property yang harus terisi terlebih dahulu sebelum property hitung dijalankan adalah sebagai berikut: Momentum, LearningRate, InputAI dan OutputAI

Nilai dari InputAI dan OutputAI pada saat Activex pertama kali diletakkan pada form adalah kosong. Property ini adalah sebagai interface antara programmer dan Activex untuk memasukkan sinyal input dan output. Contoh isian yang valid untuk InputAI dan OutputAI adalah: 1,-1,-1. InputAI dan OutputAI akan diperiksa isinya pada saat property Hitung diubah nilainya ke rsHitung. Jika tidak diisi maka akan tampil pesan kesalahan seperti pada gambar 3.2 dan gambar 3.3.

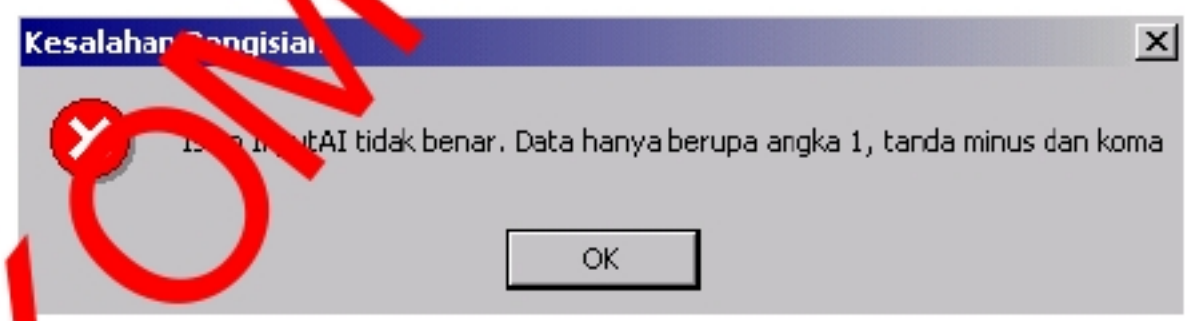
Gambar 3.2 Pesan Keslahan Property InputAI

STIKOMP SURABAYA



Gambar 3.3 Pesan Kesalahan Property Output

Jika ditemukan kesalahan-kesalahan lainnya pada property InputAI dan OutputAI, Activex akan membatalkan proses sampai isian property InputAI dan OutputAI benar. Karakter InputAI dan OutputAI yang dibenarkan adalah angka satu, koma, dan tanda minus. Dialog box yang akan muncul jika InputAI diisi dengan karakter yang tidak benar dapat dilihat pada gambar 3.4, sedangkan pada OutputAI pada gambar 3.5

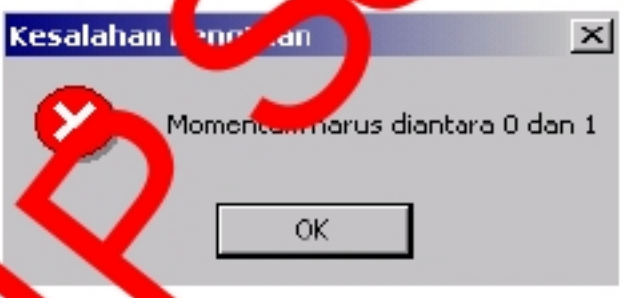


Gambar 3.4 Pesan Kesalahan Isian InputAI tidak benar

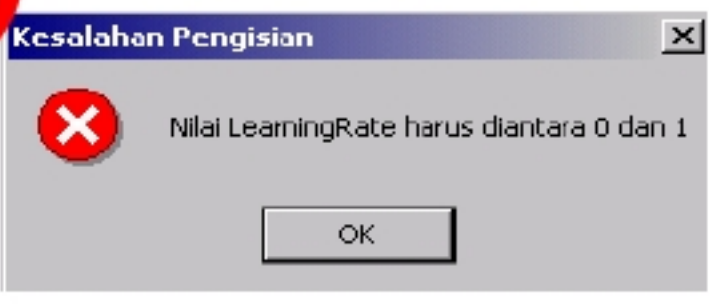
SURABAYA
STIKOMP

Gambar 3.5 Pesan Kesalahan Isian OutputAI tidak benar

LearningRate dan Momentum bertipe Double. Nilai dari kedua property ini harus bernilai antara 0 dan 1. Diluar jangkauan nilai ini Activex akan menampilkan pesan kesalahan seperti pada gambar 3.6 untuk momentum, dan gambar 3.7 untuk kesalahan LearningRate.



Gambar 3.6 Kesalahan Nilai Momentum



Gambar 3.7 Kesalahan Nilai LearningRate

STIKOM SURABAYA

Property ResetAI dipergunakan untuk membuat semua nilai bobot menjadi nol. Dengan adanya property ini programmer tidak perlu membuat nilai bobot menjadi secara manual. Setelah property ini dijalankan maka jika property hitung dijalankan lagi, maka perhitungan backpropagation dimulai dari awal atau dengan kata lain property StateLoop dimulai dari nol kembali. Property StateLoop sendiri adalah property yang menyatakan jumlah looping yang dilakukan untuk mendapatkan hasil saat ini.

Property SetBobot dipergunakan untuk mengeset nilai random untuk bobot. Jika propety ini diakses, bobot-bobot akan diberi nilai random antara 0 dan 1 secara otomatis. Jika tidak, maka pemakai Activex sendiri yang harus mengisi sendiri.

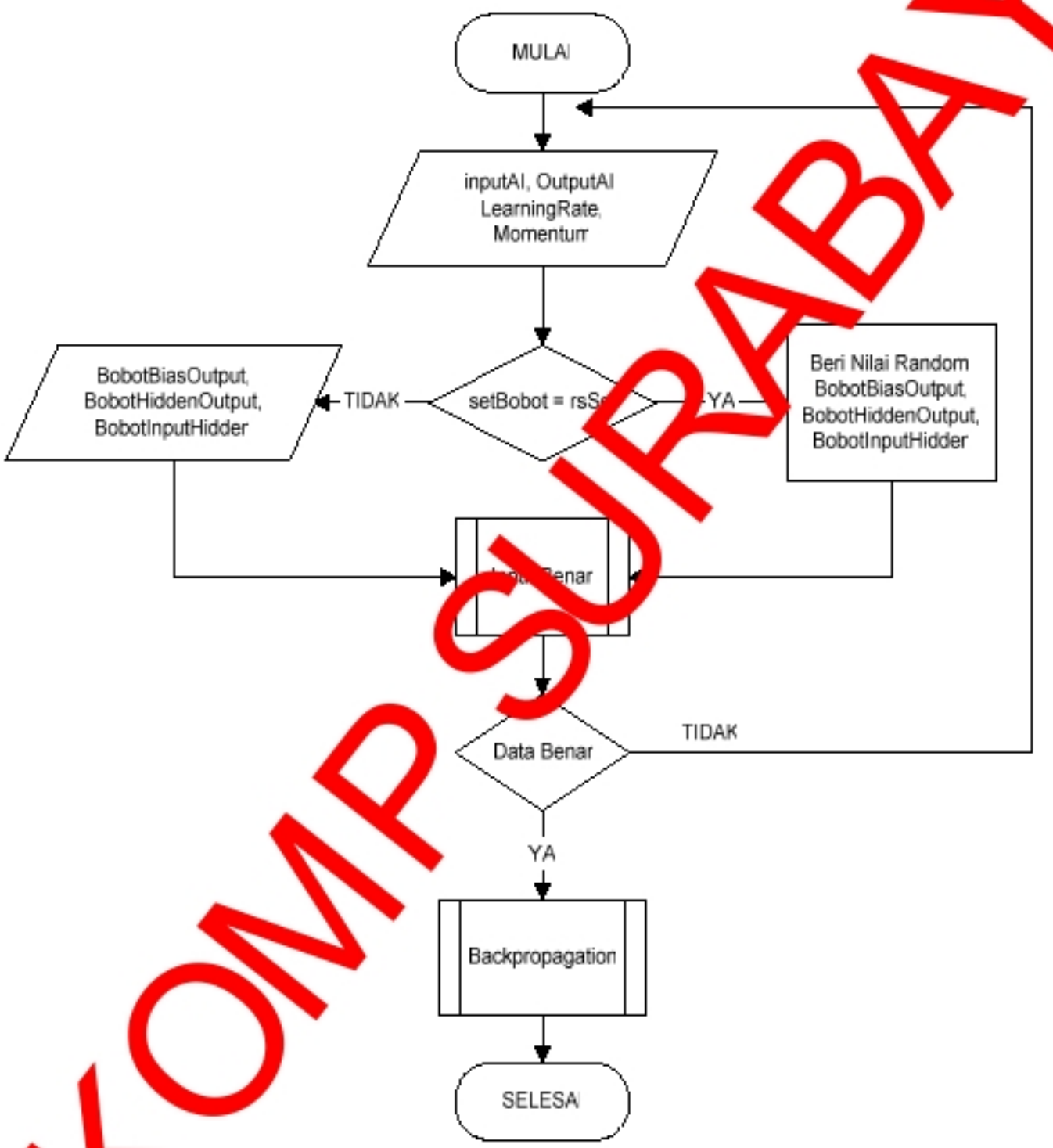
Property BobotBiasHidden, BobotBiasOutput, BobotHiddenOutput, dan BobotInputHidden adalah property yang dapat diatur oleh Activex atau oleh programmer. Jika programmer menginginkan pengaturan secara otomatis, ia dapat mengakses property SetBobot. Pengaturan dilakukan dengan mengubah nilai property ke rsSet.

3.3.2 Pembuatan flowchart

Cara pemakaian Activex mengikuti alur dari gambar 3.8. Mula-mula, pemakai Activex harus mengisi InputAI, OutputAI, LearningRate, Momentum. Sesudah itu Activex akan memeriksa, apakah Bobot awal looping akan diberi oleh Activex sendiri atau diberikan oleh pemakai. Jika tugas ini diserahkan kepada Activex, maka Activex akan memanggil prosedur Inisialisasi, jika tidak, maka

STIKOMP SURABAYA

Activex menunggu masukan dari pemakai. Sesudah itu, masukan dari pemakai akan diperiksa kebenarannya



Gambar 3.8 Flowchart Utama Kerja Activex

STIKOMPS SURABAYA

diperkenankan terisi nilai diluar karakter yang valid. Karakter yang valid untuk kedua masukan ini adalah tanda minus, angka satu dan koma.

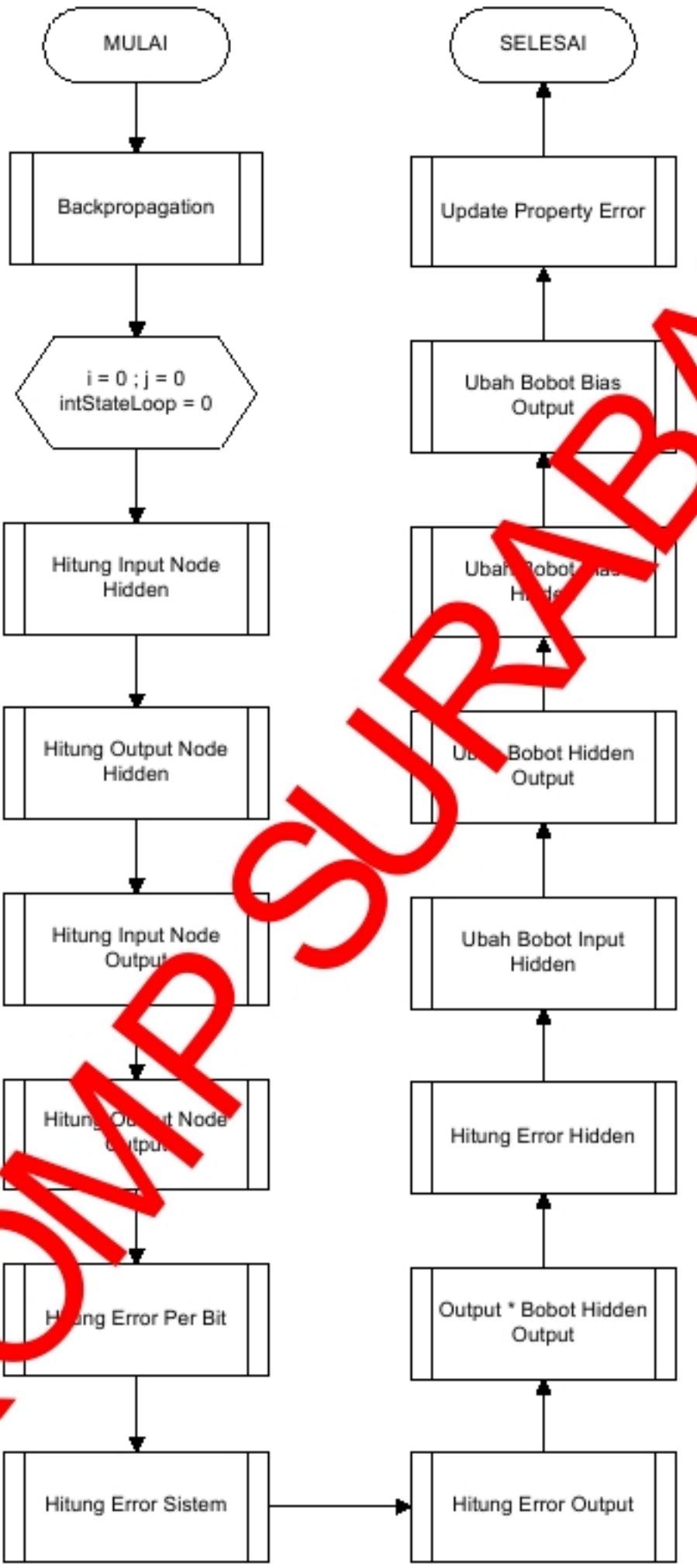
Setelah input dinyatakan benar, maka sistem akan mulai melakukan pembelajaran dengan backpropagation. Flow chart dari backpropagation dapat dilihat pada gambar 3.10. Pertama-tama variabel i dan j diberi nilai 0. Kedua variabel ini adalah variabel lokal yang ada pada prosedur backpropagation.

Input pada node hidden dari input dihitung dengan cara demikian. Dengan j adalah variabel penghitung perulangan untuk sejumlah node hidden dan i adalah variabel penghitung perulangan untuk sejumlah input, diadakan perhitungan menghitung input dikali dengan bobot antara input ke hidden. aryInput sendiri adalah sinyal yang masuk ke perhitungan, sementara wih(i,j) adalah array yang menyimpan bobot antara input ke hidden. Jumlah dari semua bobot disimpan dalam variabel array sum_hid(j).

```
For j = 1 To JumlahHidden
  For i = 1 To JumlahInput
    sum_hid(j) = sum_hid(j) + aryInput(i) * wih(i, j)
  Next i
  sum_hid(j) = sum_hid(j) + wbh(j)
Next j
```

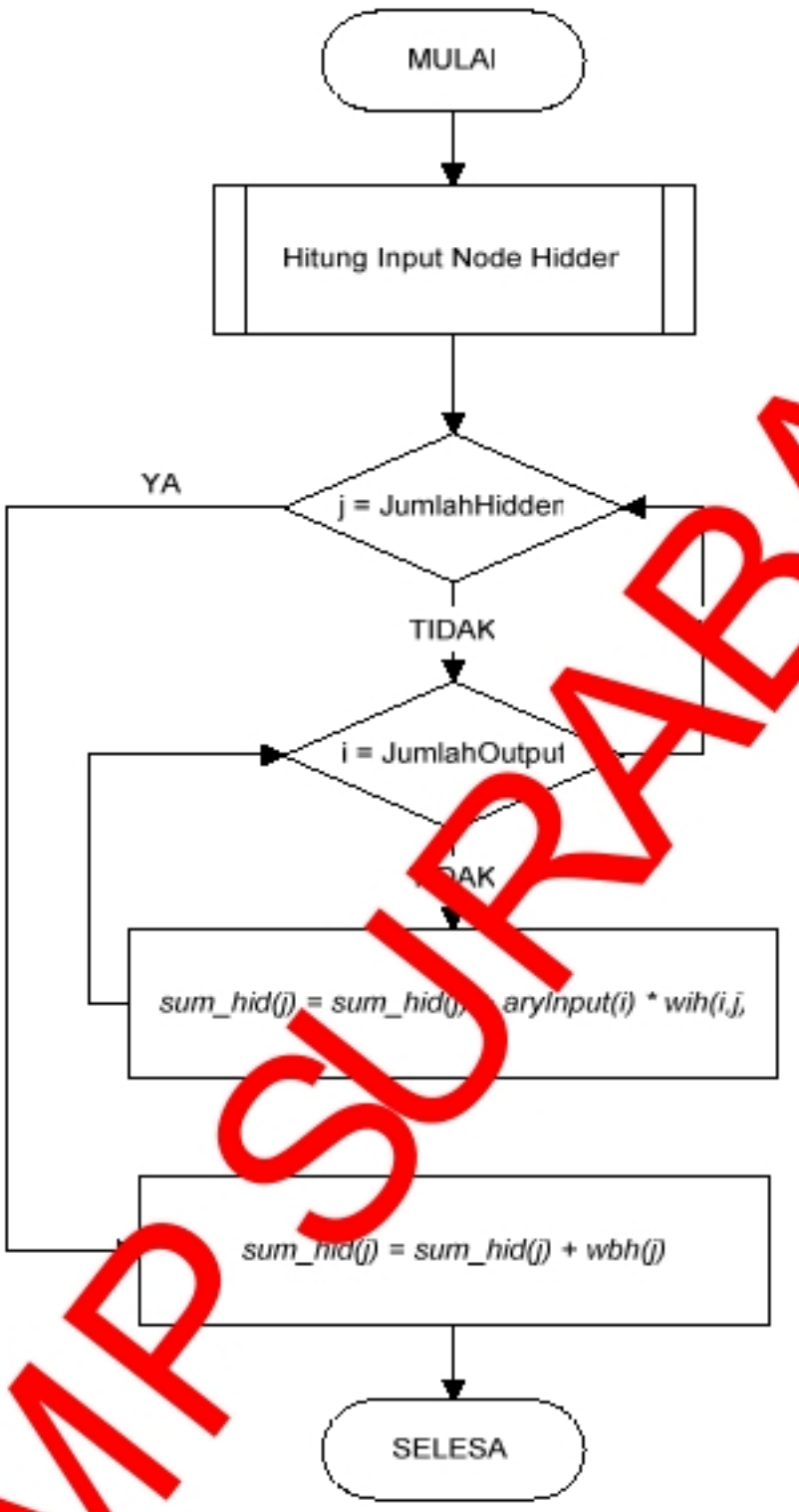
Setelah jumlah bobot ditambahkan, maka jumlah bobot tadi ditambah dengan jumlah bobot bias dari input ke hidden. Flowchart dari potongan kode program diatas dapat dilihat pada gambar 3.11.

STIKOM SURABAYA



Gambar 3.10 Flowchart Prosedur Backpropagation

STIKOMPS SURABAYA



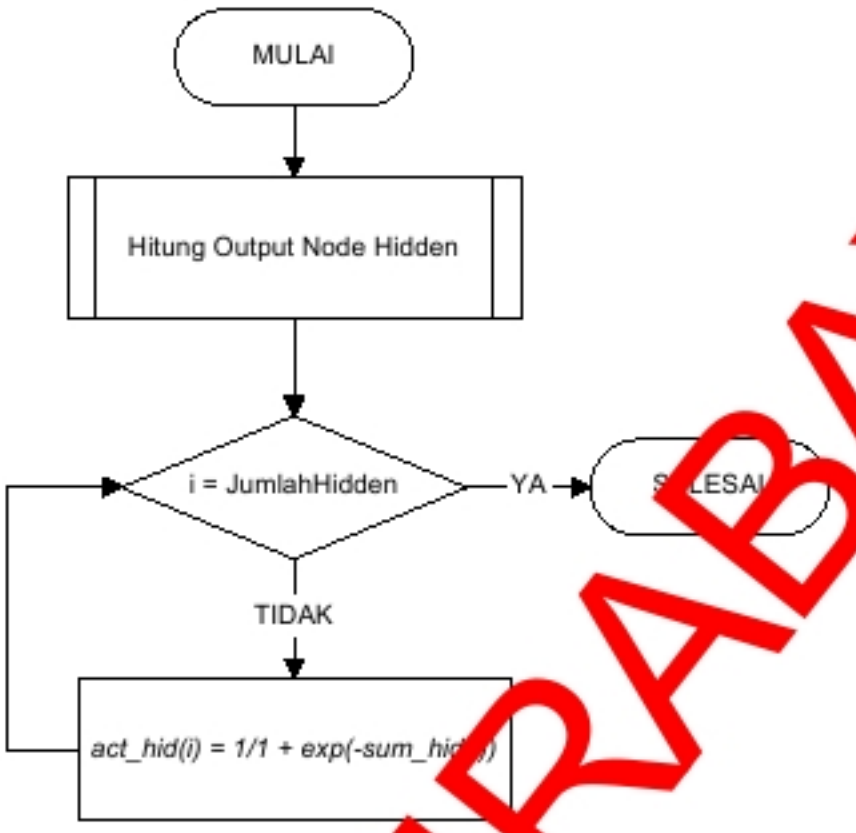
Gambar 3.11 Flowchart Input Node Hidden

Langkah selanjutnya adalah menghitung fungsi aktifasi atau output dari Hidden. Berikut adalah kode program untuk menghitungnya:

```
For i = 1 To JumlahHidden
    act_hid(i) = 1 / (1 + Exp(-sum_hid(i)))
Next i
```

Flowchart dari kode program diatas dapat dilihat pada gambar 3.12.

STIKOMPS SURABAYA

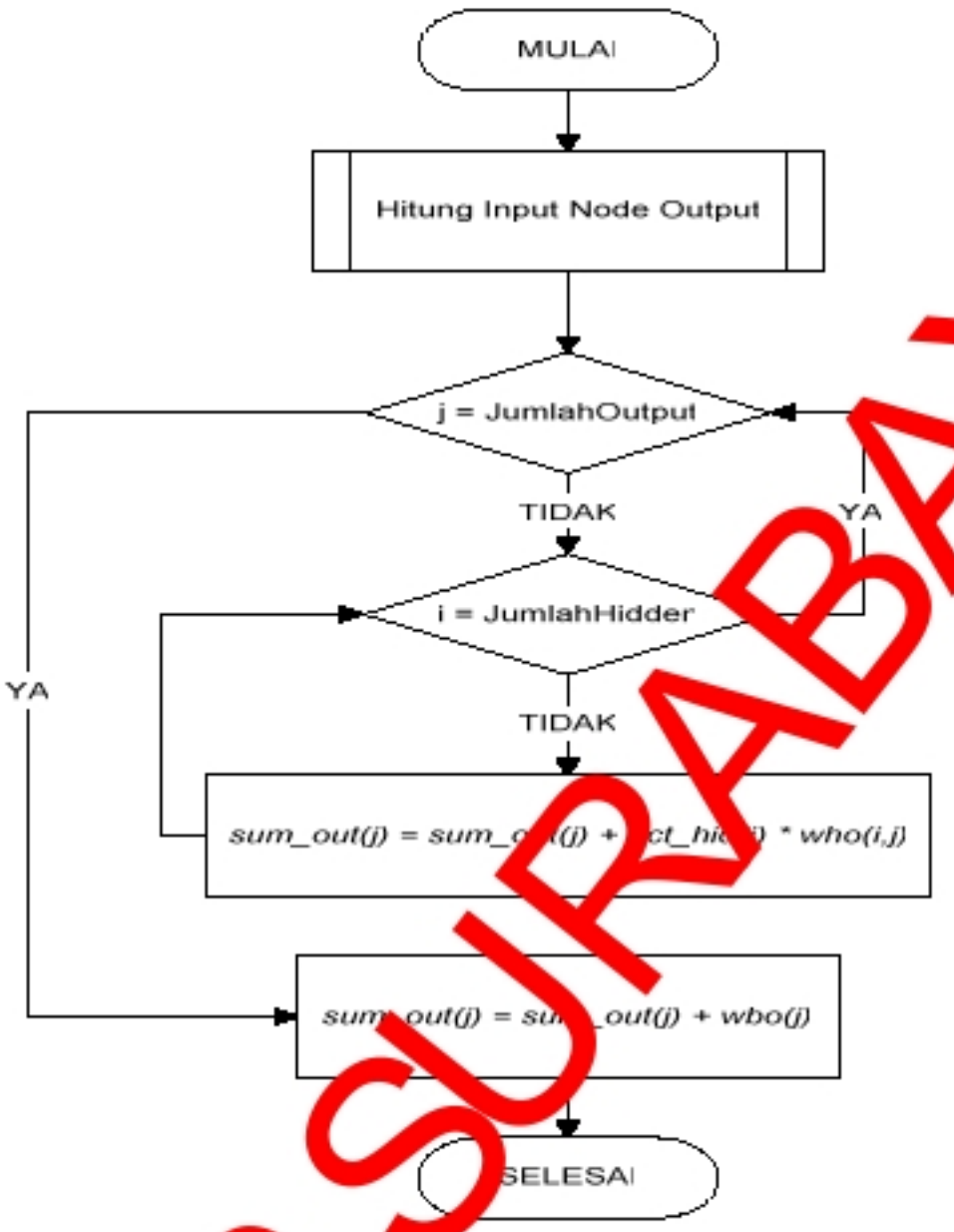


Gambar 3.12 Flowchart Output Node Hidden

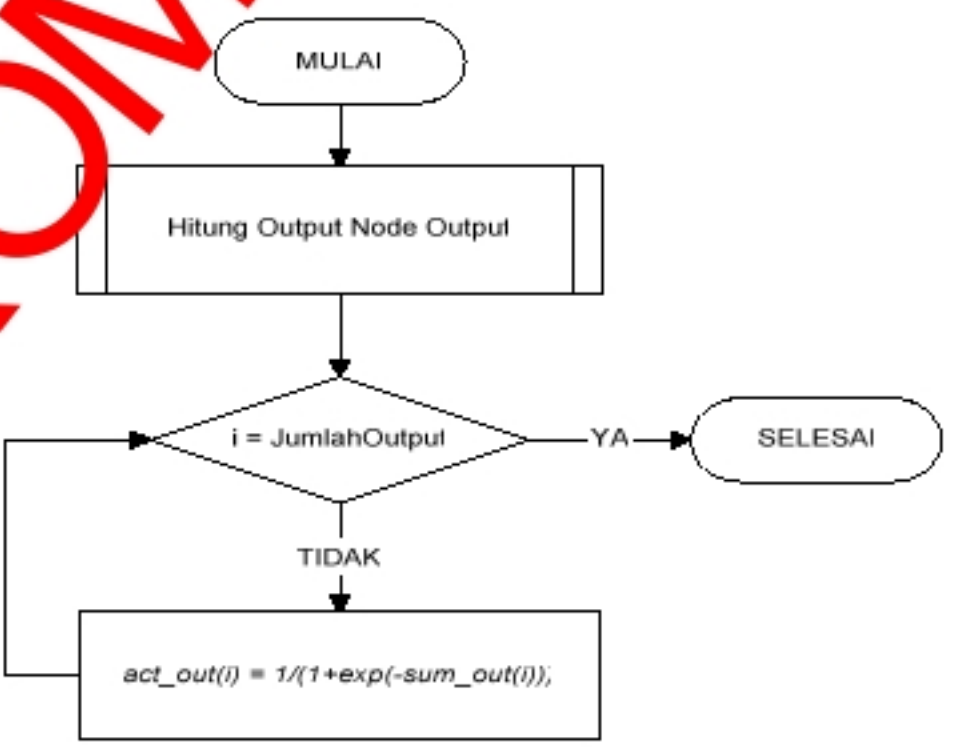
Setelah menghitung output dari hidden, sistem akan menghitung input yang masuk ke node output. Potongan kode program berikut menunjukkan penerapan flowchart pada gambar 3.13 yang menggambarkan alur kerja menghitung input ke node output. Sum_out adalah variabel array yang menyimpan hasil penambahan nilai antara perkalian nilai fungsi aktifasi node sebelumnya dan bobot antara hidden dan output, yaitu who. Hasil penjumlahan tadi ditambah dengan bobot bias antara hidden dan output wbo.

Langkah selanjutnya adalah menghitung aktifasi atau output dari node output. Gambar 3.14 menunjukkan alur kerja dari perhitungan ini.

STIKOMPP SURABAYA



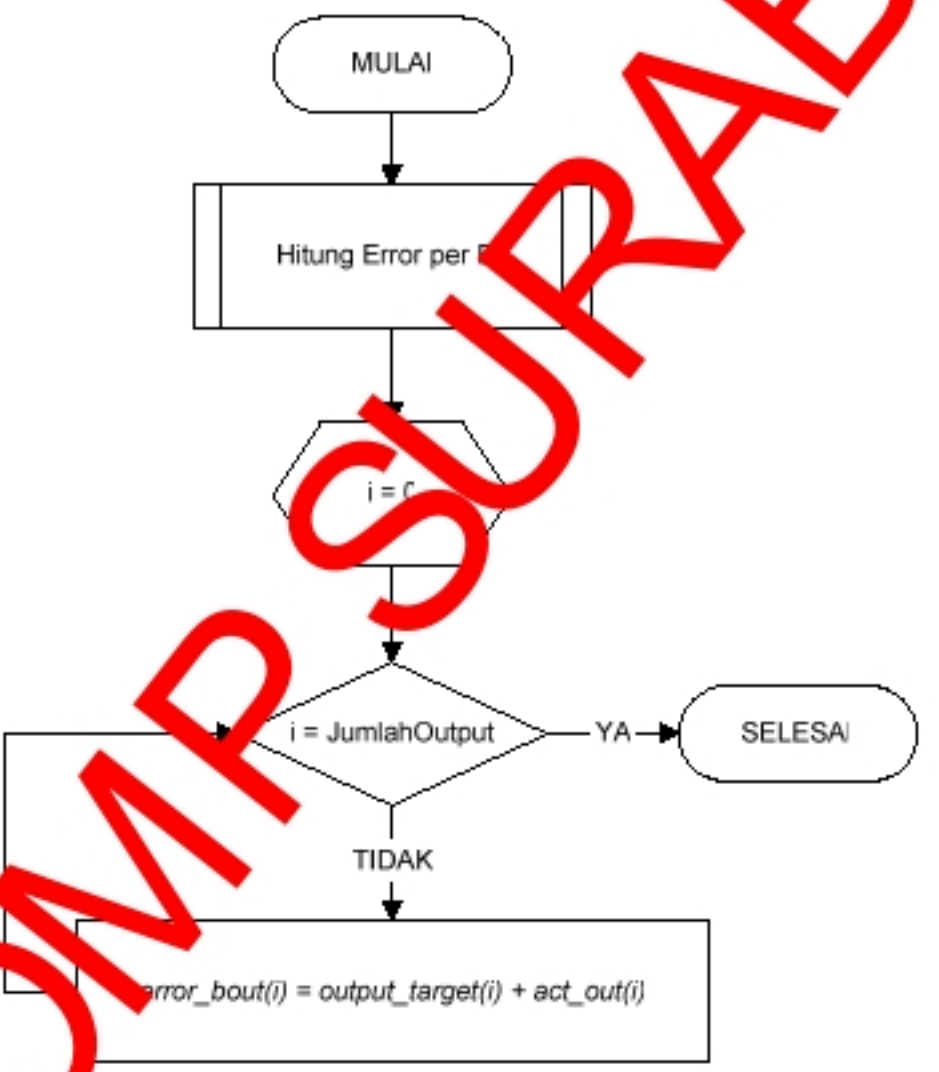
Gambar 3.1 Flowchart Input Node Output



Gambar 3.14 Flowchart Output Node Output

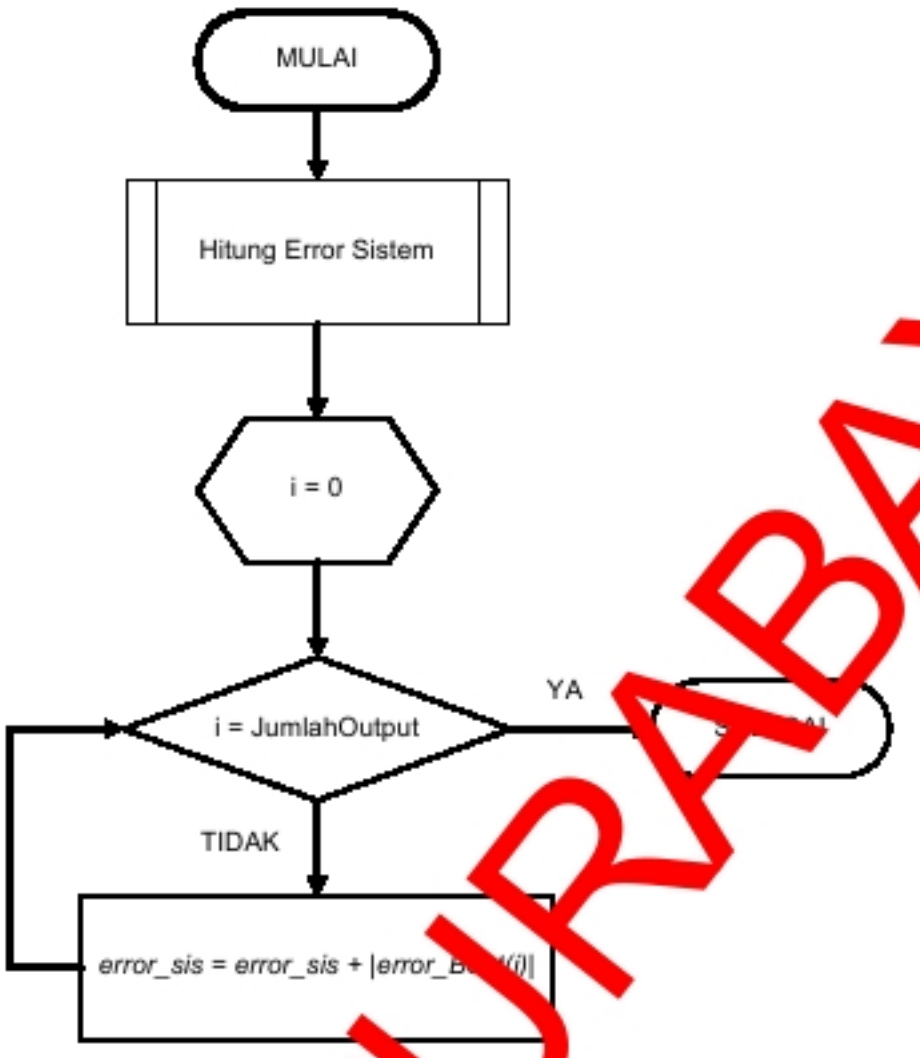
STIKOMPS SURABAYA

Perubahan harus dilakukan terhadap bobot awal setelah ditemukan error sistem pada perhitungan, flow chart error sistem dapat dilihat pada gambar 3.16. Namun sebelum menghitung error sistem, Activex akan menghitung nilai error per bit. Nilai ini nantinya akan dipakai oleh perhitungan error sistem, flow chart dari perhitungan error per bit dapat dilihat pada gambar 3.15.



Gambar 3.15 Flowchart Error Per Bit

SURABAYA
STIKOMP



Gambar 3.6 Flowchart ErrorSistem

```
For i = 1 To JumlahOutput
  error_bout(i) = aryOutput(i) - act_out(i)
Next i
```

aryOuput adalah array untuk menyimpan data output target yang sudah ditentukan diawal perhitungan. Hasil dari perhitungan disimpan dalam variabel error_bout. Isi dari error_bout ini akan dipakai pada perhitungan error sistem, seperti ditunjukkan dibawah ini:

```
For i = 1 To JumlahOutput
  error_sis = error_sis + Abs(error_Bout(i))
Next i
```

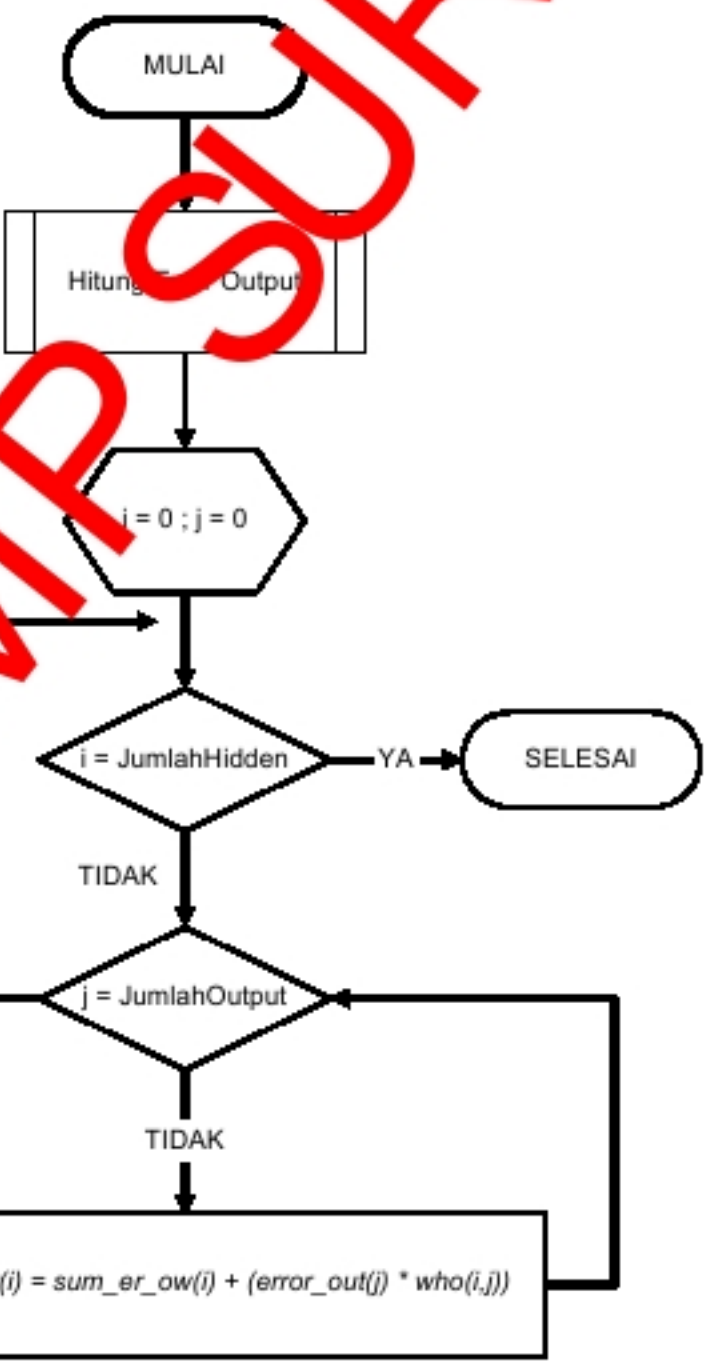
SURABAYA
STIKOMP S

error_bout dianggap selalu positif, sehingga jika hasil di error_bout adalah -1 maka tetap dianggap sebagai 1.

Error dari output dapat dihitung setelah error per bit diketemukan (error_bout).

Error output ditentukan dengan algoritma sebagai berikut (flowchart dari algoritma ini dapat dilihat pada gambar 3.17).

```
For i = 1 To JumlahHidden
  For j = 1 To JumlahOutput
    sum_er_ow(i) = sum_er_ow(i) + (error_ou(i) * wh(i, j))
  Next j
Next i
```



Gambar 3.17 Flowchart Hitung Error Output

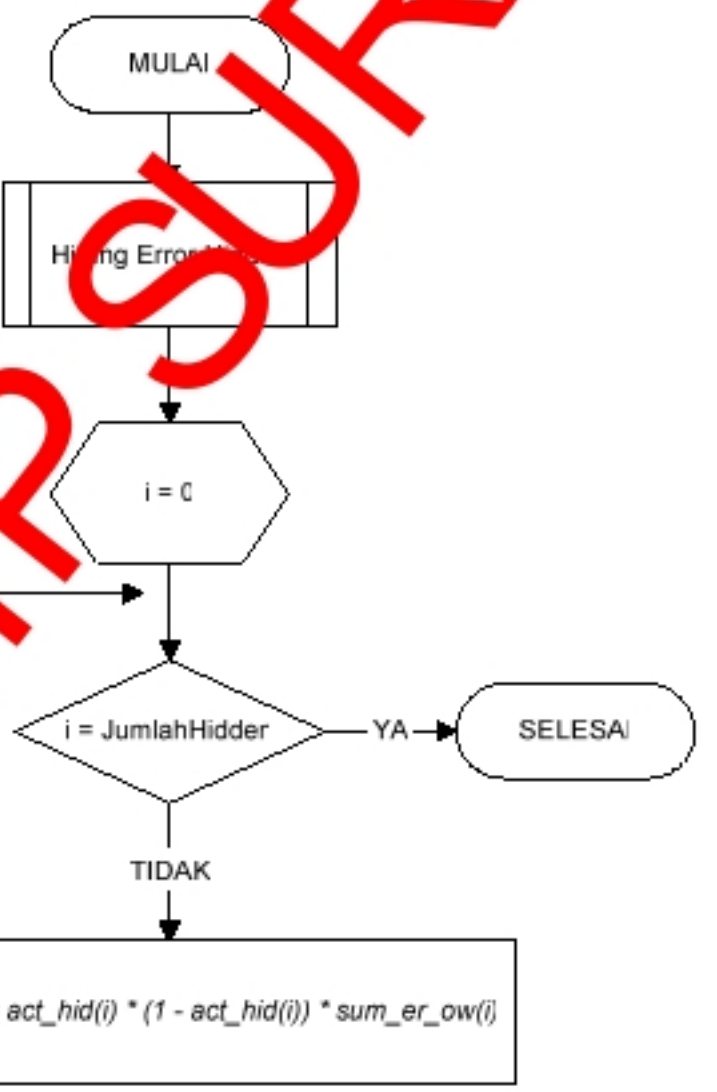
SURABAYA
STIKOMPS

Sementara eror_out pada algoritma diatas dihitung dengan cara sebagai berikut:

```
For i=1 To JumlahOutput
    error_out(i) = error_Bout(i)* act_out(i) *
    (1 - act_out(i))
Next i
```

Pada node hidden, perhitungan error dapat dilakukan dengan algoritma sebagai berikut (Flowchart pada gambar 3.18).

```
For i = 1 To JumlahHidden
 error_hid(i) = act_hid(i) * (1 - act_hid(i)) * sum_er_ow(i)
Next i
```

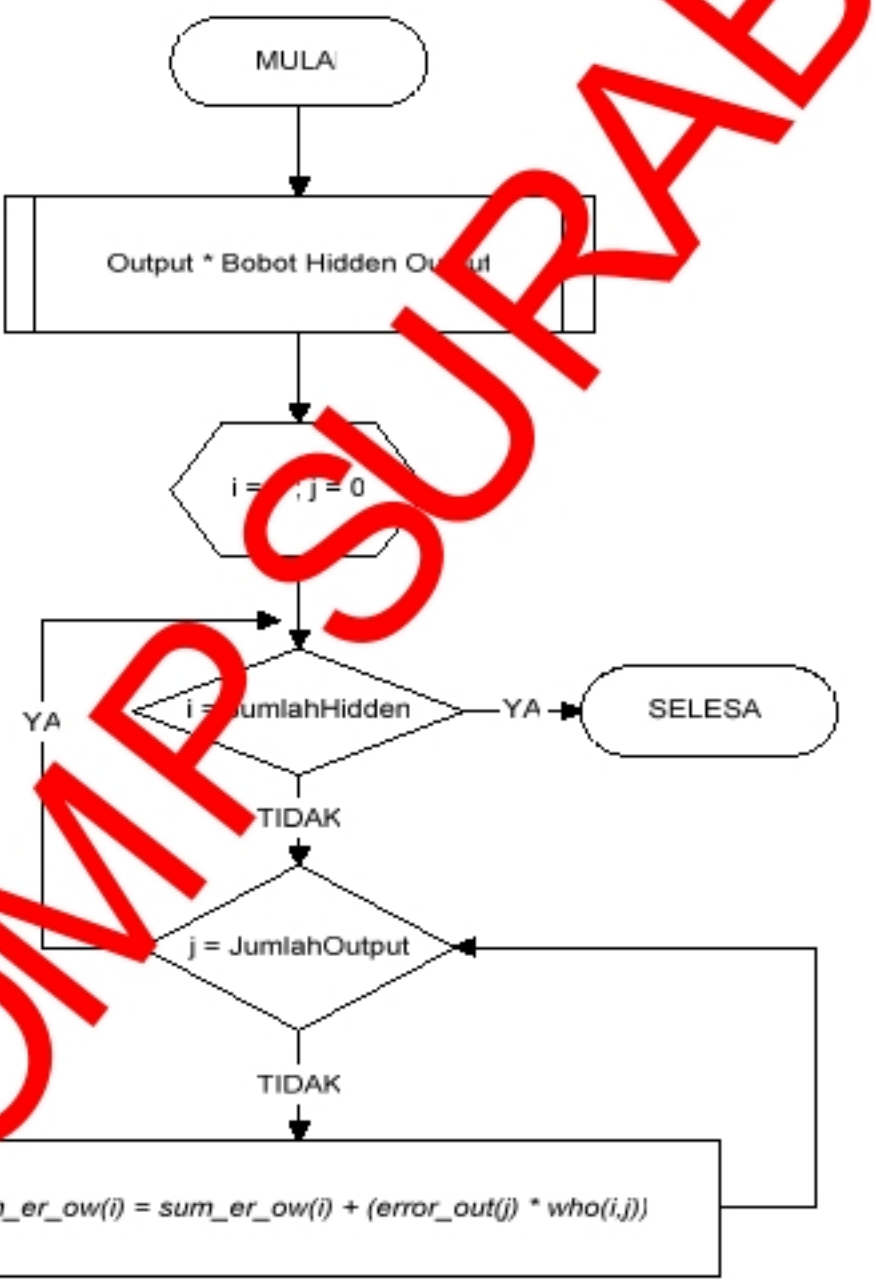


Gambar 3.18 Flowchart Menghitung Error Hidden

SURABAYA
STIKOMP

Sementara sum_er_ow didapat lewat algoritma berikut ini: (Flowchart dapat dilihat pada Gambar 3.19).

```
For i = 1 To JumlahHidden
  For j = 1 To JumlahOutput
    sum_er_ow(i) = sum_er_ow(i) + (error_out(j) * who(i, j))
  Next j
Next I
```



Gambar 3.19 Flowchart Output * Bobot Hidden Ouput

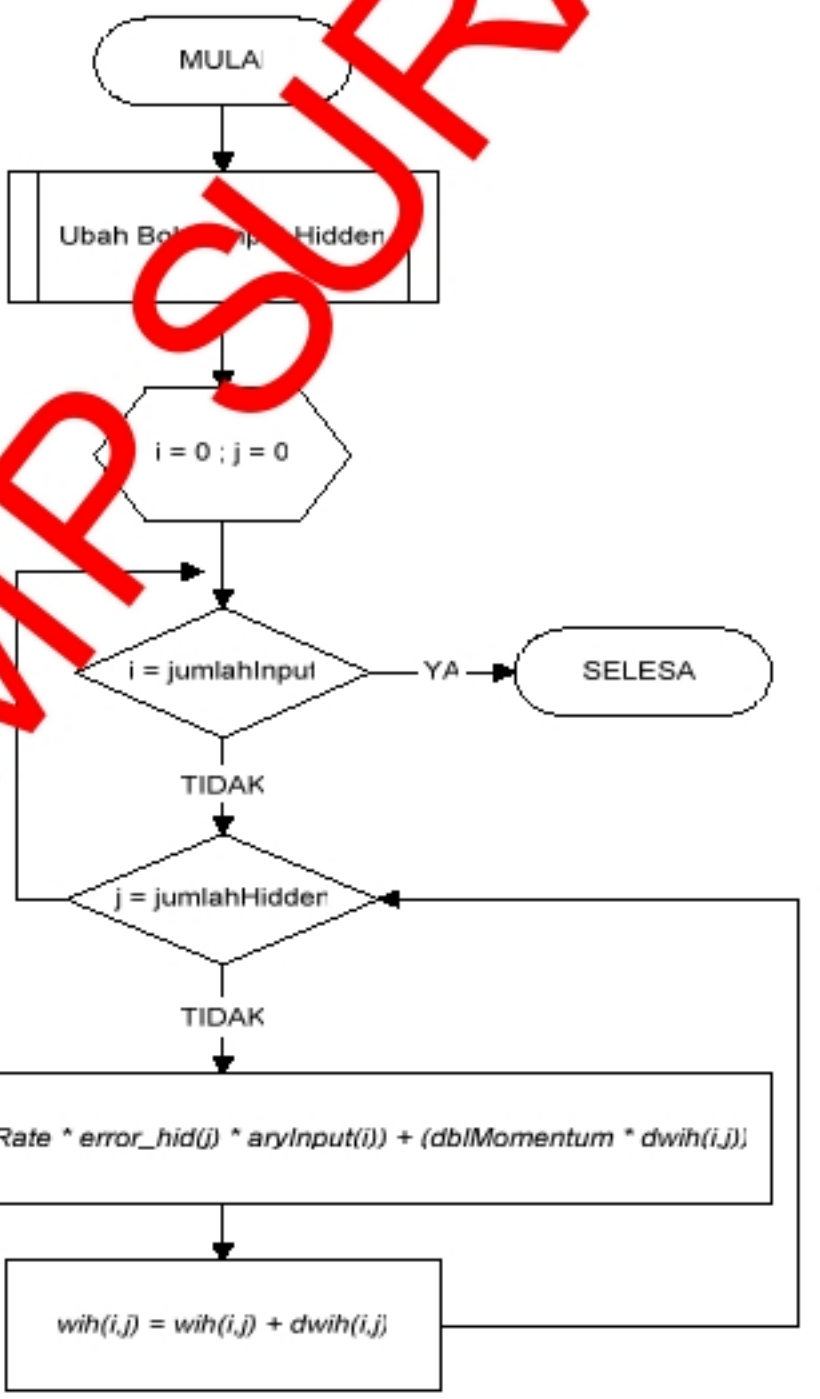
Error yang didapat pada algoritma yang sudah dijelaskan diatas telah diketahui semua. Sebagai tugas terakhir Activex adalah memperbaiki bobot yang ada.

STIKOMPS SURABAYA

Perubahan bobot yang diperbaiki adalah bobot antara input dan hidden, hidden dan output, bias dan hidden serta bias dan output.

Perubahan bobot antara input dan hidden dilakukan dengan algoritma sebagai berikut (flowchart dapat dilihat pada gambar 3.20).

```
For i=1 To JumlahInput
 For j=1 To JumlahHidden
   dwih(i,j) = (dblLearningRate * error_hid(j) * aryInput(i))+
   (dblMomentum * dwih(i,j))
   wih(i,j) = wih(i, j) + dwih(i,j)
 Next j
Next I
```



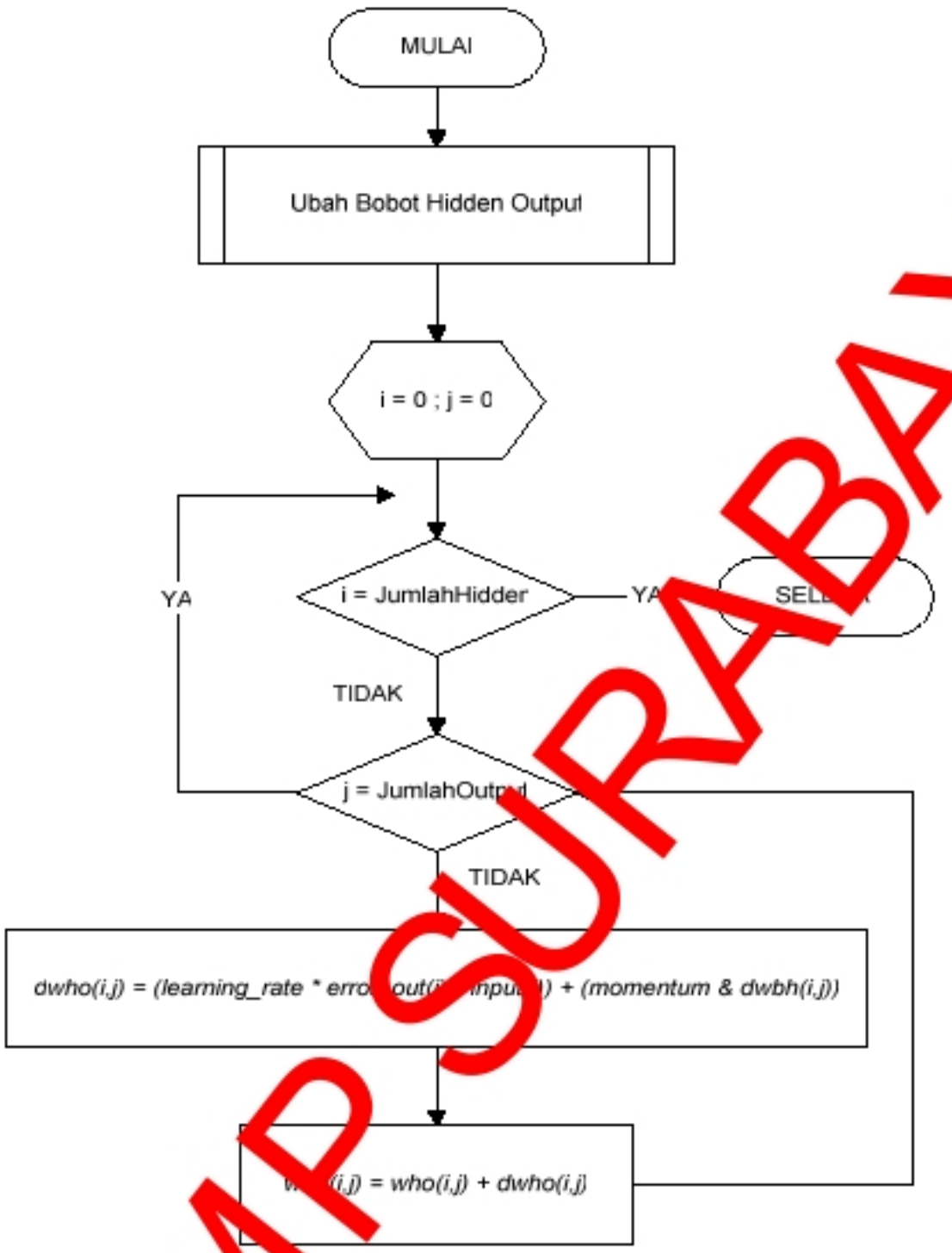
Gambar 3.20 Flowchart Ubah Bobot Hidden

STIKOMP SURABAYA

dblLearningRate adalah variabel untuk menyimpan nilai learning rate yang dimasukkan pada property LearningRate sedangkan dblMomentum adalah variabel untuk menyimpan nilai momentum yang dimasukkan oleh pengguna Activex pada property momentum. Baik dblLearningRate dan dblMomentum dibuat agar dapat menerima nilai bilangan desimal dalam jumlah besar. Variabel dwih adalah menyimpan perubahan bobot pada bobot antara input dan hidden. Isi dari dwih ditambah dengan bobot input hidden adalah bobot baru yang dipersiapkan untuk looping backpropagation selanjutnya.

Perubahan bobot antara hidden dan output mempunyai algoritma yang sama dengan perubahan bobot antara input dan hidden dengan penggantian nilai-nilai yang bersesuaian. Misalnya jika pada perubahan bobot input hidden memakai bobot yang sudah diisi di wi[illegible]aka pada perubahan bobot antara hidden output akan digantikan dengan who seperti pada gambar 3.21 berikut :

SURABAYA
STIKOMPS

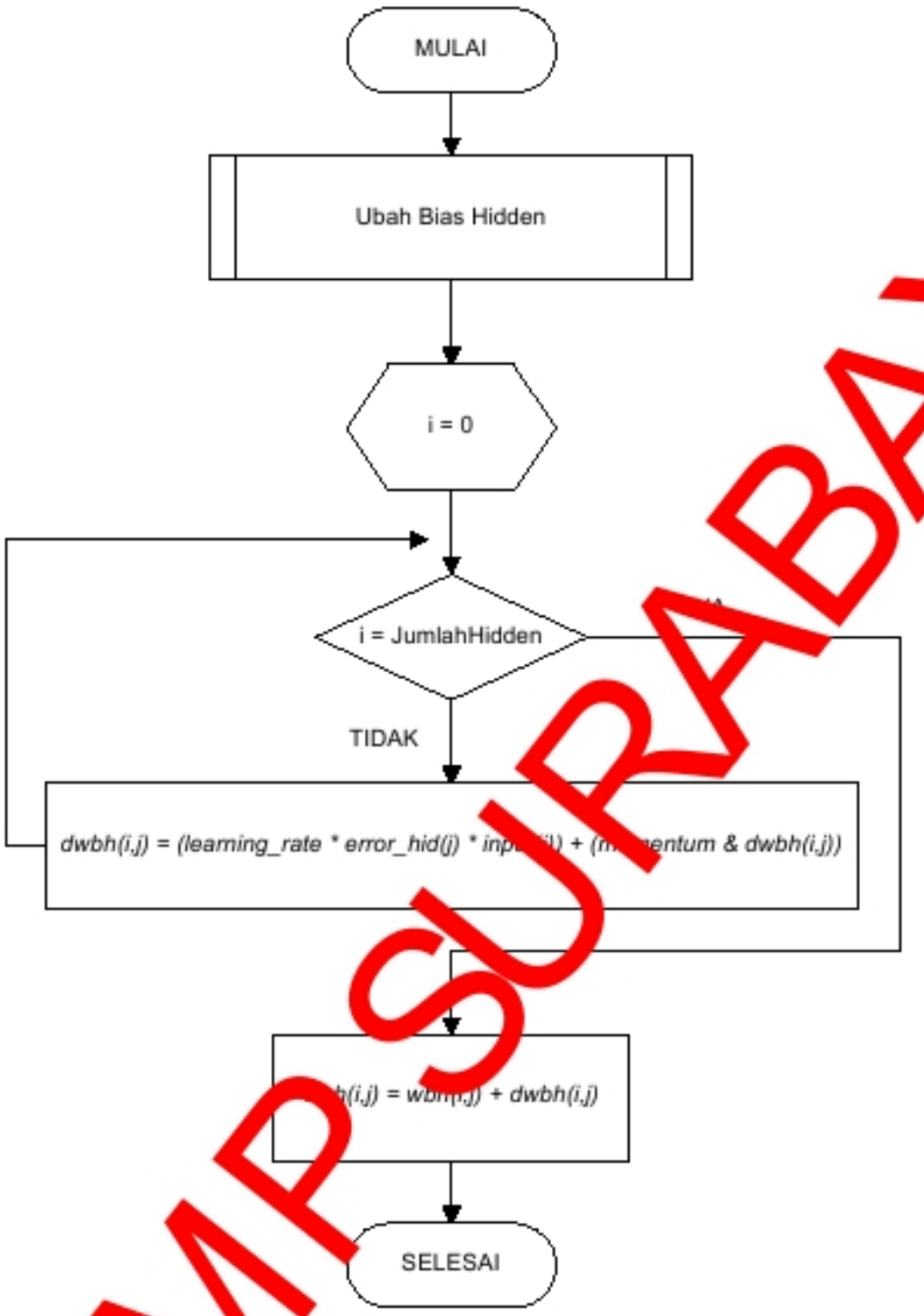


Gambar 3.21 Flowchart Ubah Bobot Hidden Output

Perubahan bobot bias hanya ada dua tipe, yaitu bias antara bias hidden dan bias output. Algoritma ubah bobot bias hidden berikut menunjukkan perhitungan tersebut sedangkan flowchart dari algoritma ini dapat dilihat pada gambar 3.22.

```
For i = 1 To JumlahHidden
 dwbh(i) = (dblLearningRate * error_hid(i)) + (dblMomentum *
bh(i))
Next i
wbh(i) = wbh(i) + dwbh(i)
```

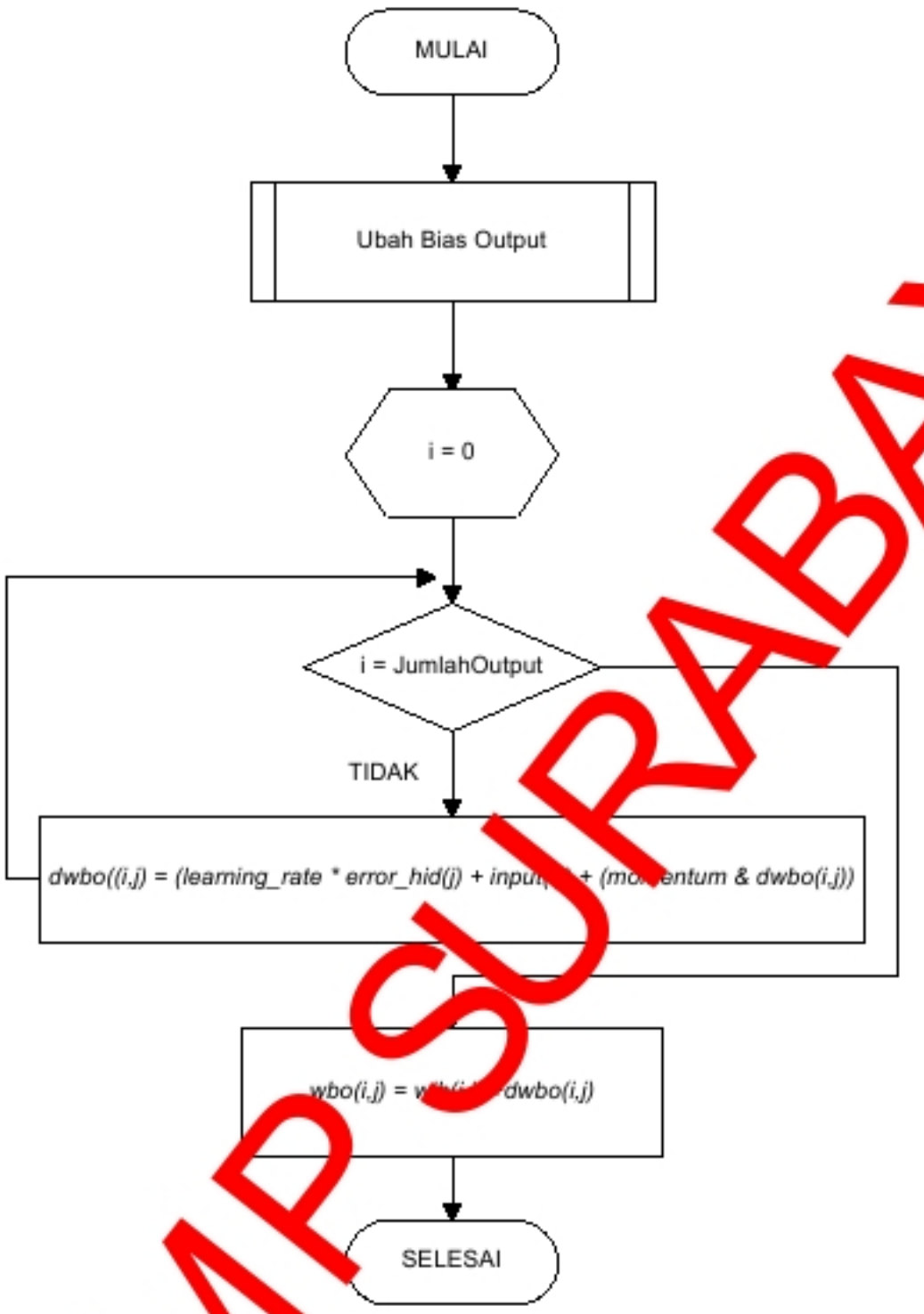
STIKOM SURABAYA



Gambar 3.22 Flowchart Ubah Bias Hidden

Perubahan bobot bias output mempunya algoritma yang sama dengan algoritma bias hidden, namun nilai yang bersesuaian diganti dengan nilai bias output. Flowchart dari algoritma ubah bias output dapat dilihat pada gambar 3.23 berikut ini:

STIKOMPS SURABAYA



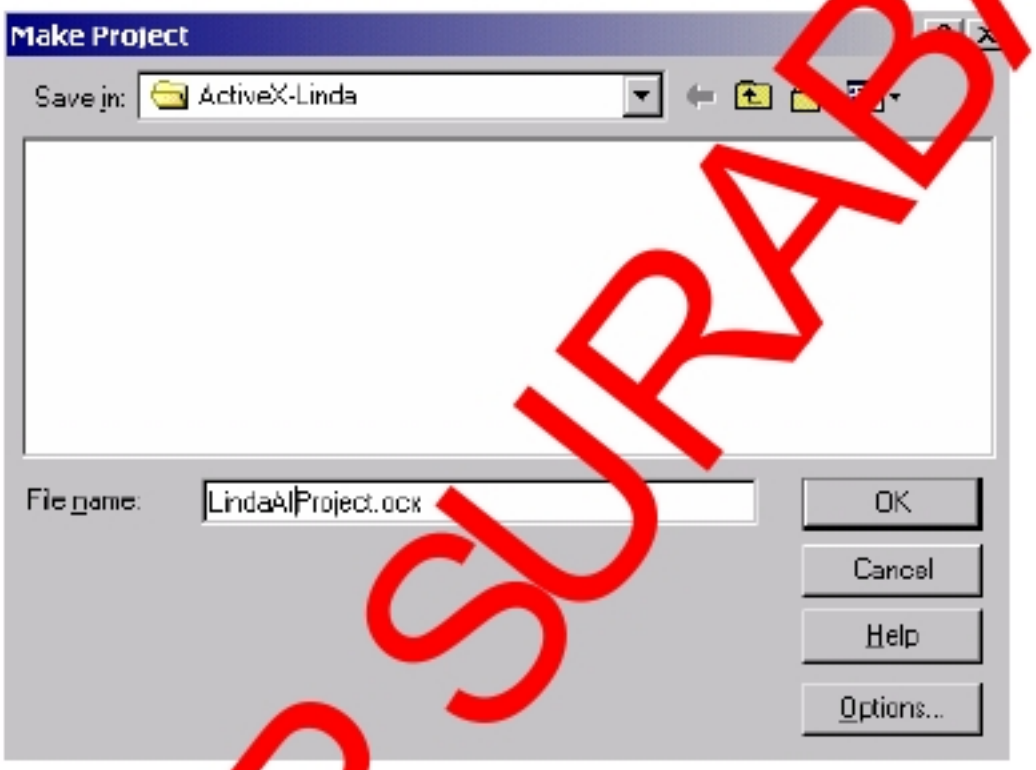
Gambar 3.23 Flowchart Ubah Bias Output

3.4 Distribusi Activex

Pada Visual Basic, setelah selesai membuat AcitveX, maka akan terdapat beberapa file. File-file tersebut ialah: *.ctl dan *.ctx. Jika kode program turut disertakan pada pengguna control, sertakan kedua tipe file diatas, namun jika pengguna tidak diperkenankan untuk melihat, maka control tersebut harus dicompile terlebih dahulu, seperti gambar 3.24.

STIKOM SURABAYA

Untuk mengkompile activex menjadi file dengan tipe ocx, dapat dilakukan dengan cara-cara berikut. Pada project explorer window pilih project Activex lebih dahulu. Kemudian pilih menu File, kemudian cari menu Make NamaFile. Pada tulisan ini nama file adalah LindaAIProject.ocx.



Gambar 3.4 Kotak Pilihan Nama Activex yang Dikompile

Setelah mengisi nama Activex yang diinginkan, tekan tombol OK. Sekarang Activex telah dikompile menjadi file dengan ekstensi ocx. Setelah dikompile, Activex harus didaftarkan pada register windows. Biasanya Activex ditakkan pada folder C:\Windows\System atau di C:\Program Files\Common Files\Microsoft Shared\. Misalkan kita mempunyai file Activex linda.ocx berada di C:\Windows\System, maka cara mendaftarkannya adalah sebagai berikut. Click tombol Start pada windows. Pilih RUN kemudian ketik:

```
Regsvr32.exe C:\Windows\System\linda.ocx
```

STIKOM SURABAYA

Kemudian klik tombol OK. Untuk membuang daftar Activex pada Windows, ketik perintah berikut:

```
Regsvr32.exe  /U C:\Windows\System\linda.ocx
```

STIKOMP SURABAYA