**APLIKASI JARINGAN SARAF TIRUAN METODE PERCEPTRON PADA PENGENALAN POLA NOTASI**

**Muhamad Arifin**

Guru SMK Telkom: SMK Telkom Malang

Email: arifin@smktelkom-mlg.sch.id

**Khoirudin Asfani**

3Dosen Teknik Elektro: Fakultas Teknik, Universitas Negeri Malang

Email: koden\_denko@yahoo.co.id

**Anik Nur Handayani**

3Dosen Teknik Elektro: Fakultas Teknik, Universitas Negeri Malang

Email: handayani.aniknur@gmail.com

**ABSTRAK:**

Jaringan saraf tiruan (JST) merupakan pemroses informasi yang meniru cara kerja otak manusia, yaitu bentuk neuron (sel syaraf). Karena kelebihan inilah, JST dapat dipakai untuk mengenali pola tertentu, pada penelitian ini pola notasi aritmatika. JST yang dipakai dengan metode perceptron. pola notasi aritmatika mulai dari notasi penjumlahan (+), pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=). Penentuan pengenalan pola notasi tersebut berdasarkan 3 inputan yang harus dimasukkan yaitu nilai bobot (w) = 0, nilai alpha (α) = 1, dan nilai threshold (θ) = 0.

Hasil dari penelitian ini bahwa pengenalan pola notasi penjumlahan (+), pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=) dari kedua perhitungan baik secara manual pada program excel dan implementasi pada program Delphi dalam Jaringan Saraf Tiruan (JST) metode Perceptron, didapatkan hasil yang sama dengan selisih 0, sehingga dapat disimpulkan perhitungan manual dengan excel dan implementasi pada program Delphi untuk pengenalan pola notasi penjumlahan (+) adalah presisi. Kemudian untuk pengenalan pola lain seperti pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=) hasilnya antara perhitungan dan program juga sama.

***Kata Kunci***: JST, Perceptron, Notasi Aritmatika.

***ABSTRACT:***

Artificial Neural Network (ANN) is an information processing approach that resembles the function or workings of the human brain, namely the form of neurons (neurons cells). Because of these advantages, ANN can be used to recognize arithmetic notation patterns, then ANN used in this research by perceptron method. The pattern of arithmetic notation starts from the notation of addition (+), subtraction (-), multiplication (x), division (/), and equals (=). The determination of the recognition of the notation pattern is based on 3 inputs that must be entered ie the weight value (w) = 0, the alpha value (α) = 1, and the threshold value (θ) = 0.

The result of this research is that the recognition of the pattern of addition notation (+), subtraction (-), multiplication (x), division (/), and equals (=) of both calculations manually on excel program and implementation in Delphi program in Artificial Neural Network (ANN) Perceptron method, obtained the same results with the difference 0, so it can be concluded manual calculations with excel and implementation in Delphi program for recognition of the pattern of addition notation (+) is precision. Then for the introduction of other patterns such as subtraction (-), multiplication (x), division (/), and equals (=) the result between the calculation and the program is also the same.

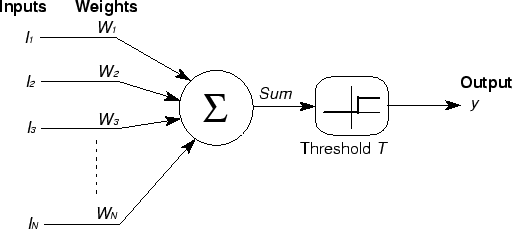
1. **PENDAHULUAN**

Jaringan Syaraf Tiruan (JST) adalah salah satu cabang ilmu dari bidang ilmu kecerdasan buatan. Salah satu model JST yang sering digunakan untuk pembelajaran adalah perceptron. Metode perceptron merupakan metode pembelajaran dengan pengawasan dalam sistem jaringan syaraf. Dalam merancang jaringan neuron yang perlu diperhatikan adalah banyaknya spesifikasi yang akan diidentifikasi. Jaringan neuron terdiri dari sejumlah neuron dan sejumlah masukan.

Pada jurnal ini, penulis akan membuat program untuk mengenali pola notasi yaitu notasi penjumlahan (+), pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=). Untuk mengenali pola notasi, salah satu metode yang dapat dipakai adalah dengan menggunakan (JST). JST merupakan suatu sistem yang bertugas untuk memproses data dengan meniru jaringan saraf biologis. Sehingga sampel notasi yang dijadikan sebagai input dapat dikenali oleh komputer seperti otak yang memproses informasi dan kemudian mengenali pola notasi yang dilihat oleh mata. Karena kelebihan inilah, jaringan saraf tiruan merupakan metode yang tepat untuk mengenali pola notasi. Selain kelebihan-kelebihan di atas, jaringan saraf tiruan juga memiliki kemampuan untuk belajar dan sifat toleransi kesalahan (*fault tolerance*) [1].

Perkembangan Jaringan Saraf Tiruan telah dimulai pada tahun 1940 dengan mengasosiasikan cara kerja otak manusia dengan logika numerik yang diadaptasi peralatan computer [Muis, Saludin, 2006]. Sederhananya, jaringan saraf tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan saraf biologi, di mana jaringan saraf tiruan menyerupai otak manusia dalam mendapatkan pengetahuan yaitu dengan proses *learning* (belajar) dan menyimpan pengetahuan yang didapat di dalam kekuatan koneksi antarneuron [2].

Jaringan Saraf Tiruan (JST) mampu mengenali kegiatan dengan berbasis pada data. Data akan dipelajari oleh JST sehingga memiliki kemampuan untuk memberi keputusan terhadap data yang belum dipelajari []. JST ditentukan oleh 3 hal, yakni: pola hubungan antarneuron (arsitektur jaringan), metode untuk menentukan bobot penghubung (metode *training*/*learning*/ algoritma) dan fungsi aktivasi [3]. Struktur jaringan saraf tiruan dapat dilihat pada Gambar 1.



**Gambar 1.1 Struktur Jaringan Saraf Tiruan** [4]

Neuron Y menerima *input* dari neuron *x1*, *x2*, dan *x3* dengan bobot hubungan masing- masing adalah *w1*, *w2*, dan *w3*. Kemudian ketiga impuls neuron yang ada dijumlahkan, sehingga dapat ditulis:

*net = x1w1 + x2w2 + x3w3*.

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi y = f(net). Apabila nilai aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk mengubah bobot [5][6].

1. **METODE PELAKSANAAN**

**2.1. Penentuan Nilai Inputan (x)**

Pada pola notasi “+” dibuat, seperti pada Gambar 3. Setelah itu vektor masukan (pola notasi yang digunakan) dan target (*t*) yang diinginkan dibentuk, dengan bobot (*w*) awal = 0 serta bias (*b*) awal = 1. Lalu setiap titik pada Gambar 3 (a), diambil sebagai komponen vektor. Setiap vektor masukan (pola 1 sampai 5) mempunyai matrik (3×3) = 9 komponen.

**    **

**Gambar 2.1 Pola notasi (+, -, x, /, dan =)**

Kemudian kepada titik dalam pola yang bertanda “.” diberikan nilai (–1) dan nilai (1) diberikan kepada tiap titik dalam pola yang bertanda “#”. Pola-pola notasi tersebut harus dibaca dari kiri ke kanan, dimulai dari baris yang paling atas. Pada setiap pola notasi memiliki vektor masukan pola (x), dan masukan tersebut seperti pada Tabel 2.1 berikut.

**Tabel 2.1 Nilai Masukan pada Setiap Pola Notasi**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Pola**  **Masukan** | **Nilai (x1)** | **Nilai (x2)** | **Nilai (x3)** | **Nilai (x4)** | **Nilai (x5)** | **Nilai (x6)** | **Nilai (x7)** | **Nilai (x8)** | **Nilai (x9)** |
| **+** | -1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 |
| **-** | -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| **X** | 1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 |
| **/** | -1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 |
| **=** | 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 |

**2.2 Penentuan Nilai Target (t)**

Pada tugas perceptron ini, program yang dibuat untuk mengenali lima pola notasi yaitu “(+, -, x, /, dan =)”. Nilai target (t) pada ke-lima pola notasi ini seperti pada Tabel 2.2.

**Tabel 2.2 Nilai Target (t) pada Setiap Pola Notasi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pola**  **Masukan** | **Nilai (t1)** | **Nilai (t2)** | **Nilai (t3)** | **Nilai (t4)** | **Nilai (t5)** |
| **+** | 1 | -1 | -1 | -1 | -1 |
| **-** | -1 | 1 | -1 | -1 | -1 |
| **x** | -1 | -1 | 1 | -1 | -1 |
| **/** | -1 | -1 | -1 | 1 | -1 |
| **=** | -1 | -1 | -1 | -1 | 1 |

Setelah itu, pemodelan jaringan perceptron dibentuk dan keluaran dari jaringan perceptron (*a* ) yang dibuat pun harus dihitung. Guna mendapatkan bobot (*w*) dan bias (*b*) yang diinginkan, program perceptron harus dilatih. Setelah mendapatkan bobot (*w*) dan bias (*b*) yang diinginkan, *output* yang diperoleh dari pemrograman perceptron (*a*)dibandingkan dengan target (*t*) yang sudah ditentukan.

**2.3 Perhitungan nilai aktivasi**

Perhitungan untuk nilai aktivasi unit masukan *xi* = *si* (*i* = 1, ..., *n*). Dihitung respon unit keluaran: net = + *b*



1. **HASIL DAN PEMBAHASAN**

**3.1 Perhitungan Manual Menggunakan Program *Excel***

Pada bab ini akan dilakukan perhitungan manual untuk mencari iterasi dalam pengenalan pola notasi dengan disesuaikan dengan metode perceptron. Contoh kasus perhitungan pengenalan pola notasi penjumlahan menggunakan metode perceptron, dengan data input:

1. Nilai bobot (w) = 0
2. Nilai alfa (α) = 1
3. Nilai threshold (θ) = 0
4. Nilai bias (b) = 0

**Perhitungan**

1. **Mencari nilai net1, net2, net3, net4, dan net5, pada pola penjumlahan (+) dalam epoch 1**

Net1= (x1.1\*w1.1\_baru)+(x1.2\*w1.2\_baru)+ (x1.3\*w1.3\_baru)+ ... +(x1.9\*w1.9\_baru)+(bias1­\_baru)

Net2= (x2.1\*w2.1\_baru)+(x2.2\*w2.2\_baru)+ (x2.3\*w2.3\_baru)+ ... +(x2.9\*w2.9\_baru)+(bias5\_baru)

...

Net5= (x5.1\*w5.1\_baru)+(x5.2\*w5.2\_baru)+ (x5.3\*w5.3\_baru)+ ... +(x5.9\*w5.9\_baru)+(bias5­\_baru)

1. **Mencari nilai y1=f(net1), y2=f(net2), y3=f(net3), y4=f(net4), dan y5=f(net5), pada pola penjumlahan (+) dalam epoch 1**

y1, y2, y3, y4, dan y5= jika, (net1) > θ, maka bernilai 1; jika, (net1) < -θ, maka bernilai -1; selain itu, maka bernilai 0.

1. **Mencari nilai Δbobot1 & Δbias1, Δbobot2 & Δbias2, Δbobot3 & Δbias3, Δbobot4 & Δbias4, dan Δbobot5 & Δbias5, pada pola penjumlahan (+) dalam epoch 1.**

Δbobot1 & Δbias1:

ΔW1\_1= alpha\*target1\*X1= 1\*1\*-1= -1

ΔW1\_2= alpha\*target1\*X2= 1\*1\*1= 1

...

ΔW1\_9= alpha\*target1\*X9= 1\*1\*-1= -1

ΔB1= alpha\*target1\*bias= 1\*1\*1 = 1

Δbobot2 & Δbias2:

ΔW2\_1= alpha\*target2\*X1= 1\*-1\*-1= 1

ΔW2\_2= alpha\*target2\*X2= 1\*-1\*1= -1

...

ΔW2\_9= alpha\*target2\*X9= 1\*-1\*-1= 1

ΔB2= alpha\*target2\*bias= 1\*-1\*1 = -1

...

Δbobot5 & Δbias5:

ΔW5\_1= alpha\*target5\*X1= 1\*-1\*-1= 1

ΔW5\_2= alpha\*target5\*X2= 1\*-1\*1= -1

...

ΔW5\_9= alpha\*target5\*X9= 1\*-1\*-1= 1

ΔB5= alpha\*target5\*bias= 1\*-1\*1 = -1

1. **Mencari nilai bobot1\_baru & bias1\_baru, bobot2\_baru & bias2\_baru, bobot3\_baru & bias3\_baru, bobot4\_baru & bias4\_baru, dan bobot5\_baru & bias5\_baru, pada pola penjumlahan (+) dalam epoch 1**

bobot1\_baru & bias1\_baru:

Wbaru1\_1= W1\_1 + ΔW1\_1 = 0 + -1 = -1

Wbaru1\_2= W1\_2 + ΔW1\_2 = 0 + 1 = 1

...

Wbaru1\_9= W1\_9 + ΔW1\_9 = 0 + -1 = -1

Bbaru1= B1 + ΔB1 = 0 + 1 = 1

bobot2\_baru & bias2\_baru:

Wbaru2\_1= W2\_1 + ΔW2\_1 = 0 + 1 = 1

Wbaru2\_2= W2\_2 + ΔW2\_2 = 0 + -1 = -1

...

Wbaru2\_9= W2\_9 + ΔW2\_9 = 0 + 1 = 1

Bbaru2= B2 + ΔB2 = 0 + -1 = -1

...

bobot5\_baru & bias5\_baru:

Wbaru5\_1= W5\_1 + ΔW5\_1 = 0 + 1 = 1

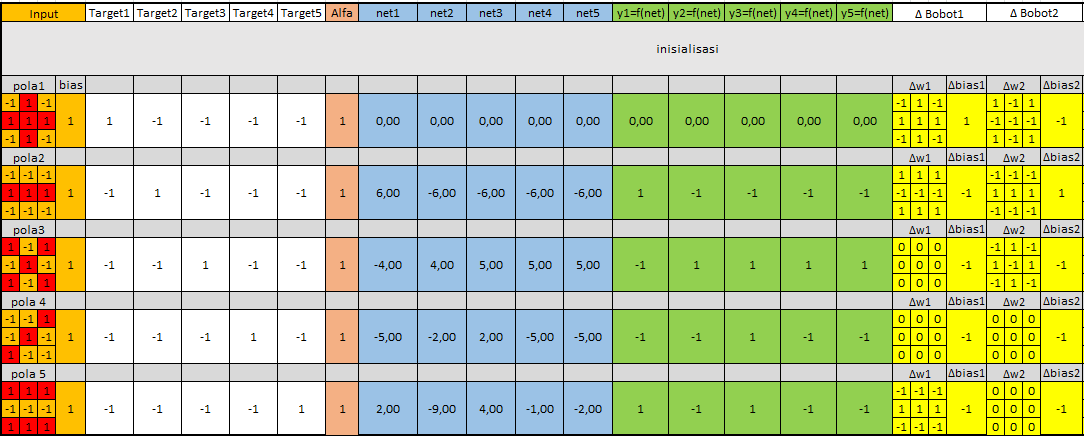
Wbaru5\_2= W5\_2 + ΔW5\_2 = 0 + -1 = -1

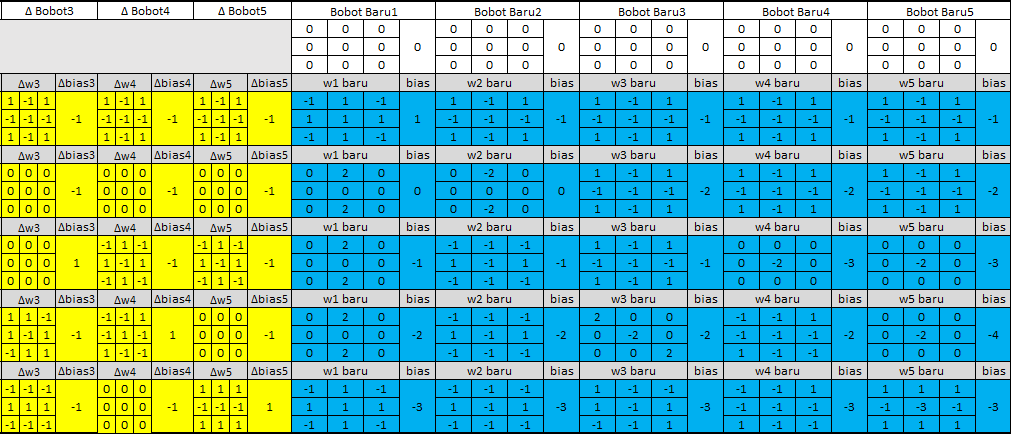
...

Wbaru5\_9= W5\_9 + ΔW5\_9 = 0 + 1 = 1

Bbaru5= B5 + ΔB5 = 0 + -1 = -1

Hasil dari perhitungan untuk rumus diatas ditunjukkan pada Gambar 3.1 berikut ini.

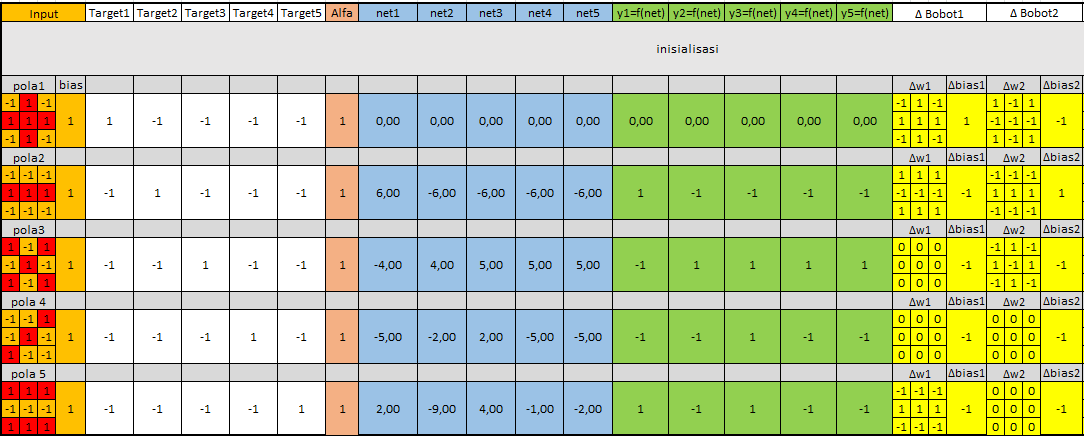


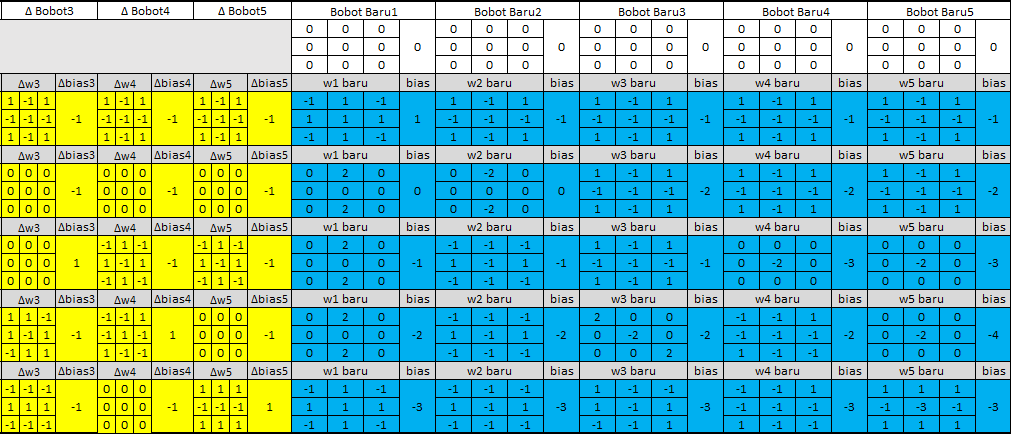


**Gambar 3.1 Hasil Analisis Perhitungan pada Program *Excel***

Kemudian, nilai bobot1\_baru & bias1\_baru hingga bobot5\_baru & bias5\_baru pada perhitungan pola penjumlahan (+), digunakan untuk perhitungan pada pola pengurangan (-) dalam *epoch* 1. Ulangi langkah 1-4 seperti sebelumnya, untuk perhitungan pada pola pengurangan (-) dalam *epoch* 1, hingga nilai bobot1\_baru & bias1\_baru hingga bobot5\_baru & bias5\_baru pada perhitungan pola pengurangan (-) ditemukan. nilai bobot1\_baru & bias1\_baru hingga bobot5\_baru & bias5\_baru pada perhitungan pola pengurangan (-) digunakan untuk perhitungan pada pola perkalian (x) selanjutnya, begitu juga dengan perhitungan pada pola pembagian (/) dan sama dengan (=). Hasil perhitungan secara keseluruhan dalam pengenalan pola pada *epoch* 1 dapat dilihat pada Gambar 3.2.

***Epoch* 1**

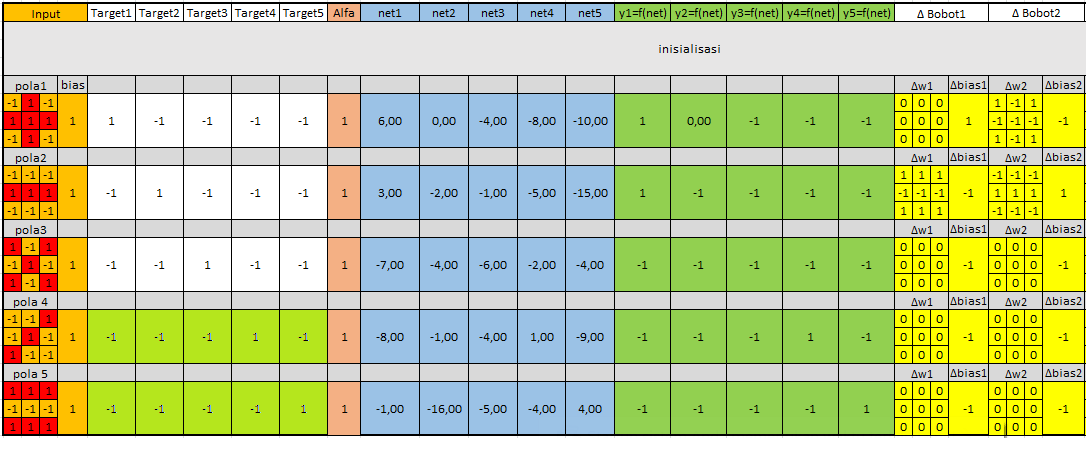


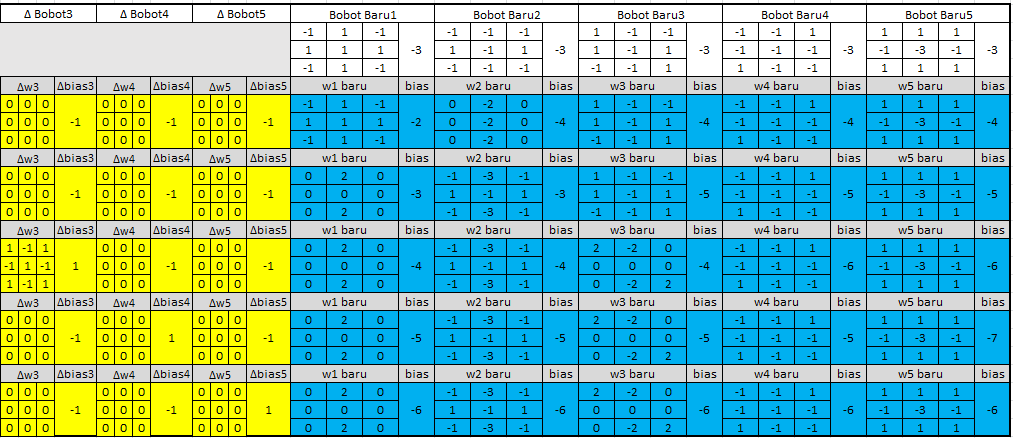


**Gambar 3.2 Hasil Analisis Perhitungan pada Program *Excel Epoch 1***

Pada iterasi 1 pengenalan pola penjumlahan, pengurangan, perkalian, pembagian dan sama dengan nilai target (t) **belum** sesuai dengan nilai f(*net*) atau nilai (y). Sehingga dilakukan iterasi ke 2 dengan hasil perhitungan seperti pada Gambar 3.2.

***Epoch* 2**

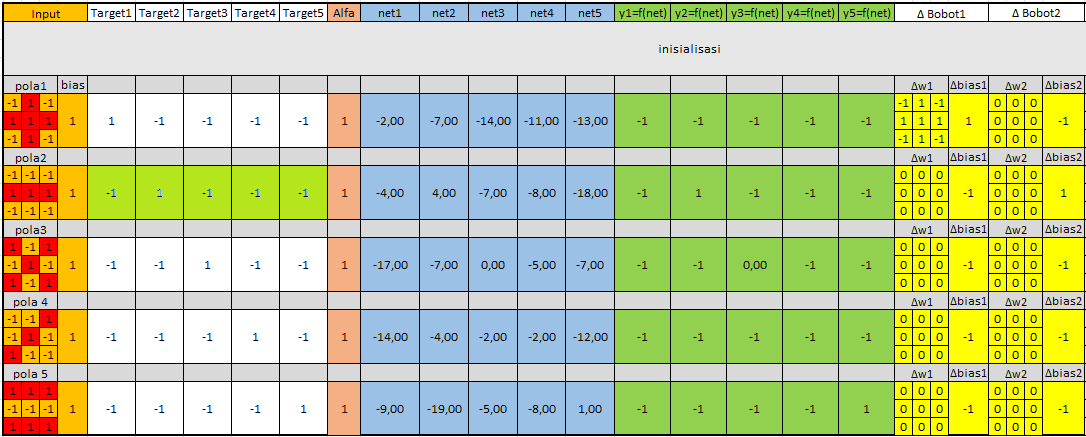
****

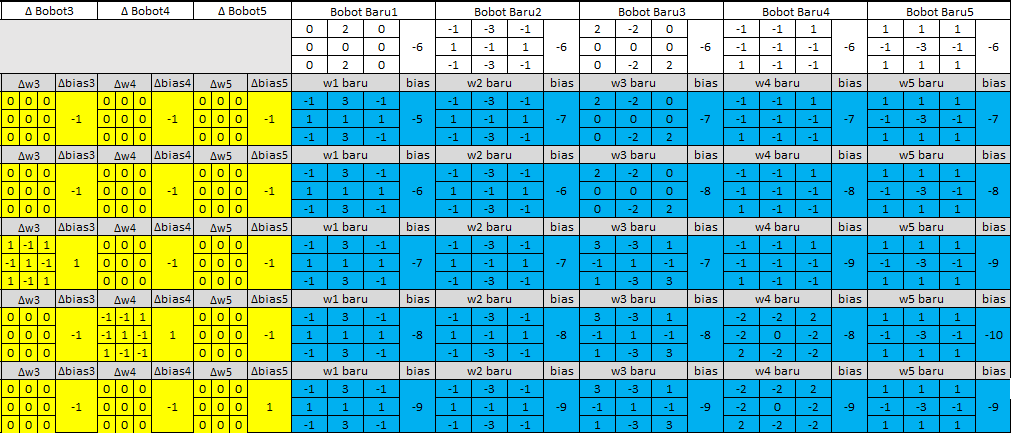
****

**Gambar 3.3 Hasil Analisis Perhitungan pada Program *Excel Epoch* 2**

Pada iterasi kedua seperti pada Gambar 3.3 didapatkan hasil bahwa pada pengenalan pola penjumlahan (+) nilai target (t) **belum** sesuai dengan nilai f(*net*) atau nilai (y), pola pengurangan (-) nilai target (t) **belum** sesuai dengan nilai f(*net*), pengenalan pola perkalian (x) nilai target (t) **belum** sesuai dengan nilai f(*net*), pola pembagian (/) nilai target (t) sudah sesuai dengan nilai f(*net*), kemudian pola sama dengan (=) nilai target (t) sudah sesuai dengan nilai f(*net*). Untuk pola yang belum memenuhi target dilakukan iterasi ketiga.

***Epoch* 3**

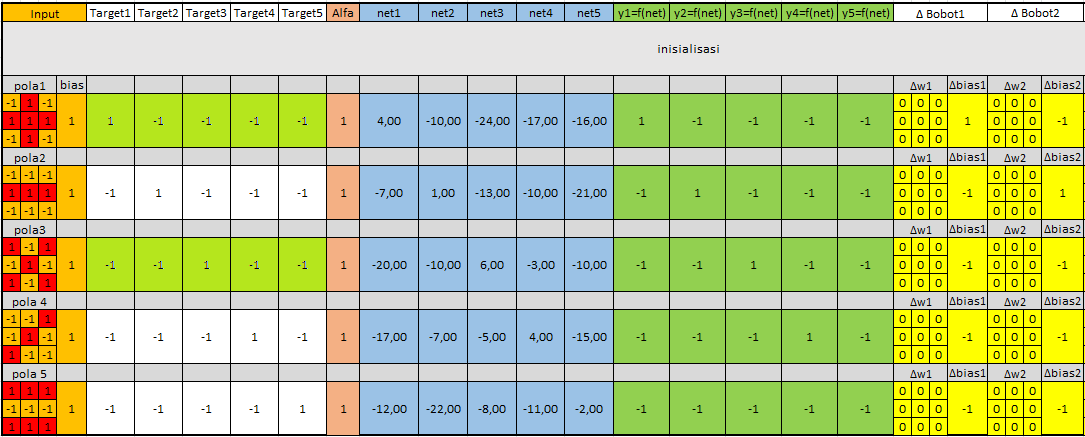
****

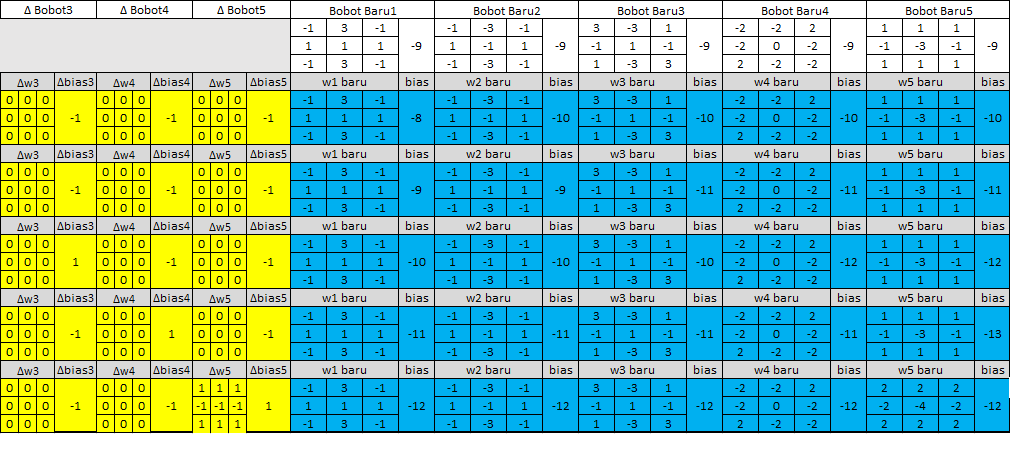
****

**Gambar 3.4 Hasil Analisis Perhitungan pada Program *Excel Epoch* Ketiga**

Pada iterasi ketiga seperti pada Gambar 3.4 didapatkan hasil bahwa pada pengenalan pola penjumlahan (+) nilai target (t) **belum** sesuai dengan nilai f(*net*), pola pengurangan (-) nilai target (t) **sudah** sesuai dengan nilai f(*net*), pengenalan pola perkalian (x) nilai target (t) **belum** sesuai dengan nilai f(*net*). Untuk pola yang belum memenuhi target dilakukan iterasi ketiga.

***Epoch* 4**

****

****

**Gambar 3.5 Hasil Analisis Perhitungan pada Program *Excel Epoch* Keempat**

Pada iterasi keempat seperti pada Gambar 3.5 didapatkan hasil bahwa pada pengenalan pola penjumlahan (+) nilai target (t) **sudah** sesuai dengan nilai f(*net*) dan pengenalan pola perkalian (x) nilai target (t) **sudah** sesuai dengan nilai f(*net*).

Kesimpulan dari keempat iteraksi tersebut, dengan menggunakan model Perceptron dimana **alpha = 1, threshold = 0, bobot** dan **bias awal = 0**, maka didapatkan hasil untuk mengenali pola 1 (+), iterasi dilakukan sampai *epoch* 4, untuk mengenali pola 2 (-), iterasi dilakukan sampai *epoch* 3, untuk mengenali pola 3 (x), iterasi dilakukan sampai *epoch* 4, untuk mengenali pola 4 (/), iterasi dilakukan sampai *epoch* 2, dan untuk mengenali pola 5 (=), iterasi dilakukan sampai *epoch* 2.

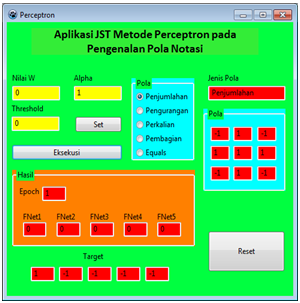
* 1. **Implementasi Antarmuka Program Perceptron**

Tampilan program pada pengenalan pola notasi metode perceptron yang telah dibuat dapat dilihat pada Gambar 3.6.



**Gambar 3.6 Tampilan Awal Program Perceptron untuk Pengenalan Pola Notasi**

Untuk menjalankan program terlebih dahulu memasukkan nila w =0, nilai alpha=0, threshold=0, kemudian pilih pola penjumlahan, kemudian klik tombol set dan lalukan ekseskusi, untuk mereset ulang pilih tombol reset. Setelah dieksekusi pengenalan pola penjumlahan pada *epoch* 1 untuk nilai target (t) belum sesuai dengan nilai f (*net*), sehingga dilakukan eksekusi lagi, hasil lihat Gambar 3.7.



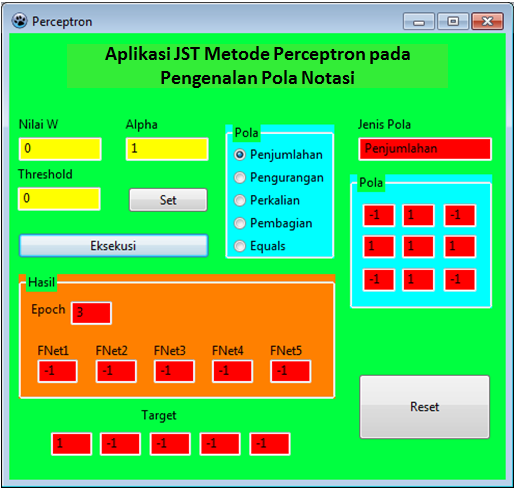
**Gambar 3.7 Tampilan Hasil Program Perceptron untuk Pengenalan Pola Notasi pada *Epoch* 1**

Eksekusi selanjutnya untuk pengenalan pola penjumlahan pada *epoch* 2, ternyata untuk nilai target (t) belum sesuai dengan nilai f (*net*), sehingga dilakukan eksekusi lagi, hasil lihat Gambar 3.8.



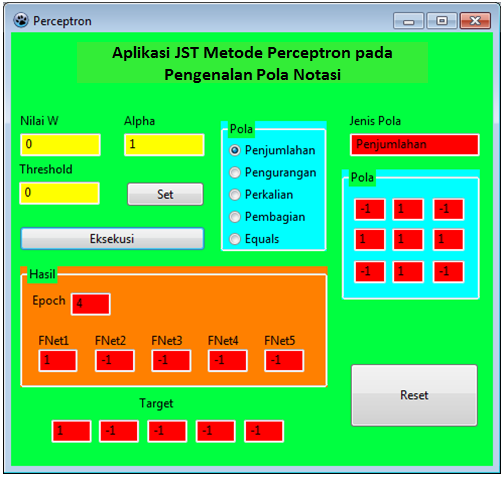
**Gambar 3.8 Tampilan Hasil Program Perceptron untuk Pengenalan Pola Notasi pada *Epoch* 2**

Pengenalan pola penjumlahan pada *epoch* 3, ternyata untuk nilai target (t) belum sesuai dengan nilai f (*net*), sehingga dilakukan eksekusi lagi, hasil lihat Gambar 3.9.

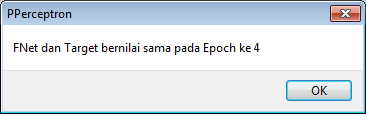


**Gambar 3.9 Tampilan Program Perceptron untuk Pengenalan Pola Notasi pada *Epoch 3***

Kemudian eksekusi selanjutnya pada *epoch* 4, untuk nilai target (t) sudah sesuai dengan nilai f (*net*), dan ini sesuai perhitungan pada excel, sehingga tidak perlu dilakukan eksekusi lagi, hasil *epoch 4* lihat Gambar 3.10. Selain itu pada program ini juga akan memberikan notifikasi atau pemberitahuan bahwa nilai f (*net*) dan target (t) bernilai sama pada *epoch* 4, tampilan bisa dilihat pada Gambar 3.11.



**Gambar 3.10 Tampilan Hasil Program Perceptron untuk Pengenalan Pola Notasi pada *Epoch* 4**



**Gambar 3.11 Tampilan Notifikasi Program Perceptron untuk Pengenalan Pola Notasi pada *Epoch* 4 Telah Berhasil**

**Analisis**

Hasil pengenalan pola notasi penjumlahan (+), pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=) dari kedua perhitungan baik secara manual pada program *excel* dan implementasi pada program Delphi dalam Jaringan Saraf Tiruan (JST) metode Perceptron, didapatkan hasil yang sama dalam pengenalan pola, tabulasi data hasil ditunjukkan pada Tabel 3.1 berikut.

**Tabel 3.1 Tabulasi Data Hasil Pengenalan Pola Notasi pada Metode Perceptron**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Pola yang dipilih** | **Perhitungan** | | | | **Hasil** | | **Selisih** |
| **Pengenalan Pola pada Excel** | | **Pengenalan Pola pada Program** | | **Excel (epoch)** | **Program (epoch)** |
| **+** | Inputan | Nilai (w) = 0  Nilai (α) = 1  Nilai (θ) = 0 | Inputan | Nilai (w) = 0  Nilai (α) = 1  Nilai (θ) = 0 | 4 | 4 | 0 |
| **+** | Inputan | Nilai (w) = 0  Nilai (α) = 0,1  Nilai (θ) = 0 | Inputan | Nilai (w) = 0  Nilai (α) = 0,1  Nilai (θ) = 0 | 9 | 9 | 0 |

Berdasarkan Tabel 3.1 bahwa pada pengenalan pola notasi studi kasus pola penjumlahan didapatkan hasil yang sama antara perhitungan manual dengan *excel* dan implementasi pada program Delphi dengan selisih 0, sehingga dapat disimpulkan perhitungan manual dengan *excel* dan implementasi pada program Delphi untuk pengenalan pola (studi kasus pola penjumlahan) adalah presisi. Kemudian untuk pengenalan pola yang lain seperti pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=) hasilnya antara perhitungan dan program juga sama, namun tidak dijelaskan secara keseluruhan pada laporan ini. Data lebih lengkap mengenai perhitungan manual program *excel* dengan inputan yang berbeda dapat dilihat pada lampiran laporan.

1. **KESIMPULAN**

Penggunaan Jaringan Saraf Tiruan (JST) metode perceptron dapat digunakan untuk penentuan pengenalan pola notasi pola notasi penjumlahan (+), pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=). Penentuan pengenalan pola notasi tersebut berdasarkan 3 inputan yang harus dimasukkan yaitu nilai bobot (w) = 0, nilai *alpha* (α) = 1, dan nilai *threshold* (θ) = 0.

Hasil pengenalan pola notasi penjumlahan (+), pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=) dari kedua perhitungan baik secara manual pada program *excel* dan implementasi pada program Delphi dalam Jaringan Saraf Tiruan (JST) metode Perceptron, didapatkan hasil yang sama dengan selisih 0, sehingga dapat disimpulkan perhitungan manual dengan *excel* dan implementasi pada program Delphi untuk pengenalan pola (studi kasus pola penjumlahan) adalah presisi. Kemudian untuk pengenalan pola yang lain seperti pengurangan (-), perkalian (x), pembagian (/), dan sama dengan (=) hasilnya antara perhitungan dan program juga sama.

**DAFTAR PUSTAKA**

[1] Nugroho, Fx. Henry. *Pengenalan Wajah dengan Jaringan Saraf* *Tiruan Backpropogation.*Yogyakarta: Graha Ilmu.

[2] Islam, M.J, dkk. 2009. *Neural Network Based Handwritten Digits* *Recognition- An Experiment and Analysis*. University of Windsor,Canada.

[3] Siang, Jong Jek. 2005. *Jaringan Saraf Tiruan dan Pemrogramannya* *Menggunakan MATLAB.* Yogyakarta: Penerbit Andi.

[4] Kawaguchi, Kiyoshi, 2000. *A Multithreaded Software Model for Backpropagation Neural Network Applications*. Department of Electrical and Computer Engineering: The University of Texas At El Paso. Thesis Online, diakses tanggal 10 November 2017.

[5] Sri Kusumadewi & Sri Hartati. 2006. *Neuro Fuzzy-Integrasi Sistem Fuzzy dan Jaringan Syaraf.* Yogyakarta: Graha Ilmu.

[6] Muis, Saludin. 2006 . *Teknik Jaringan Saraf Tiruan*. Yogyakarta. Penerbit Andi.