

BAB III

PERANCANGAN SISTEM

3.1 Identifikasi Permasalahan dan Analisa Kebutuhan Sistem

Penerapan sistem belajar untuk dapat membantu penentuan *hero* pada tahap eliminasi dan tahap pemilihan DoTA dengan mode CM memiliki permasalahan yang kompleks, karena tahap eliminasi dan tahap pemilihan *hero* dilakukan secara bergantian antar 2 tim yang bertanding, dengan komposisi 1-1-1-1-1-1-1-1 untuk tahap eliminasi, dan 1-2-2-2-2-1. Dibutuhkan total 7 arsitektur ANN untuk 1 tim, yaitu 4 arsitektur ANN untuk tahap eliminasi, dan 3 arsitektur ANN untuk tahap pemilihan. Pada penelitian tugas akhir ini akan difokuskan pada sistem penentuan *hero* untuk tim 2.

Adanya variasi data yang cukup banyak dan memiliki perbedaan yang besar untuk 1 *hero* pada tahap eliminasi awal, dimana data masukan ANN hanya ada 1, yaitu *hero* yang dieliminasi tim 1, maka sangat dimungkinkan pembelajaran ANN pada tahap ini akan mengalami kegagalan.

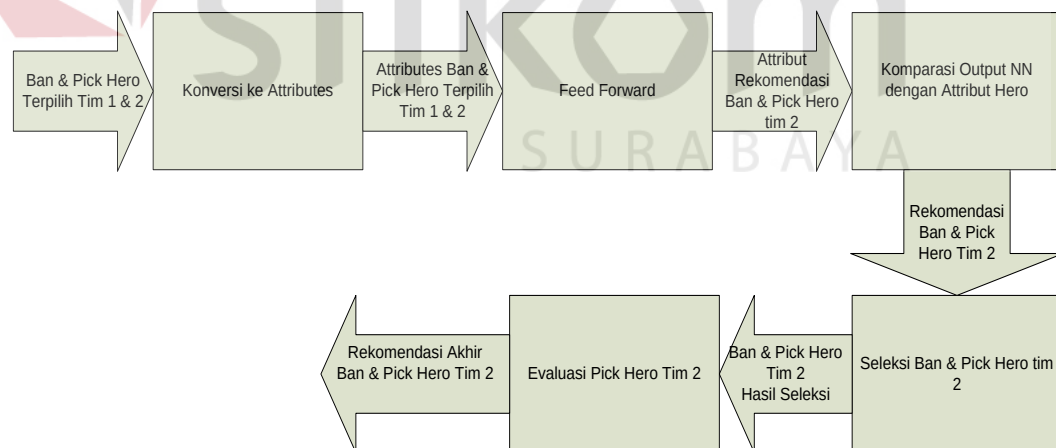
Permasalahan lainnya adalah dibutuhkannya cara untuk mengasosiasikan *hero input* dan *hero output* ANN secara tepat, karena ID *hero* kurang dapat mewakili karakteristik *hero* terkait. Karakteristik tiap *hero* dapat diwakili dengan 13 kemampuan dasar (atribut) yang ada pada tiap *hero*, oleh sebab itu maka *hero input* dan *hero output* ANN akan diasosiasikan terhadap ke 13 atribut yang ada, yaitu *tanker*, *ganker*, *initiator*, *pusher*, *disabler*, *AOE stunner*, *AOE nuker*, *nuker*, *support*, *semi carry*, *carry / DPS*, *stunner*, dan *slower*. *Hero output* ANN yang berupa nilai 13 atribut *hero* perlu diproses lebih lanjut untuk dapat menghasilkan

hero yang berkarakter sesuai, sehingga *output* ANN dijadikan dasar seleksi dari tiap *hero* yang ada berdasarkan pada nilai dari 13 atributnya. *Hero* rekomendasi ANN akan ditampilkan secara urut berdasarkan pada kesesuaian 13 atribut *hero* terkait dengan 13 atribut hasil dari ANN.

Sistem juga perlu memberikan tambahan informasi untuk membantu keputusan pemilihan *hero* dari daftar rekomendasi *hero* yang dihasilkan ANN di tahap pemilihan, yaitu *counter hero*, *combo 2 hero*, dan *combo 3 hero*. Pemberian informasi ini akan dilakukan dengan proses *query* dari tabel *counter hero*, *combo 2 hero*, dan *combo 3 hero* terhadap *hero* rekomendasi ANN dengan *hero* yang telah dipilih oleh tim 2.

3.2 Desain Sistem

Berdasarkan identifikasi permasalahan diatas maka blok diagram ANN pada sistem penentuan *hero* dapat dilihat pada gambar 3.1



Gambar 3.1. Blok Diagram Sistem Penentuan Hero pada DotA

Sistem penentuan hero terdiri dari 5 modul, yaitu:

1. Konversi ke *attributes*

Proses konversi *attributes* adalah proses merubah ID hero yang diinputkan dari tim 1 dan tim 2 menjadi atribut-attribut hero yang telah ditentukan.

2. *Feed Forward*

Proses *Feed forward* adalah proses yang dilakukan ANN untuk menghasilkan rekomendasi atribut hero, dengan menggunakan bobot yang telah dilatihkan.

3. Komparasi Output NN dengan Atribut Hero

Output atribut dari proses ANN akan dicocokkan dengan atribut dari tiap hero yang ada, dan banyaknya kecocokan akan menentukan besar prosentase kecocokan.

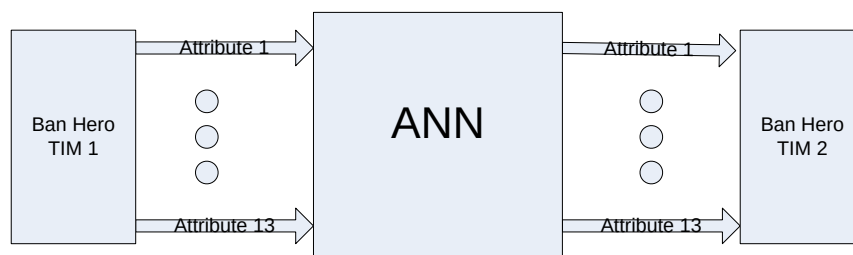
4. Seleksi Ban & Pick hero tim 2

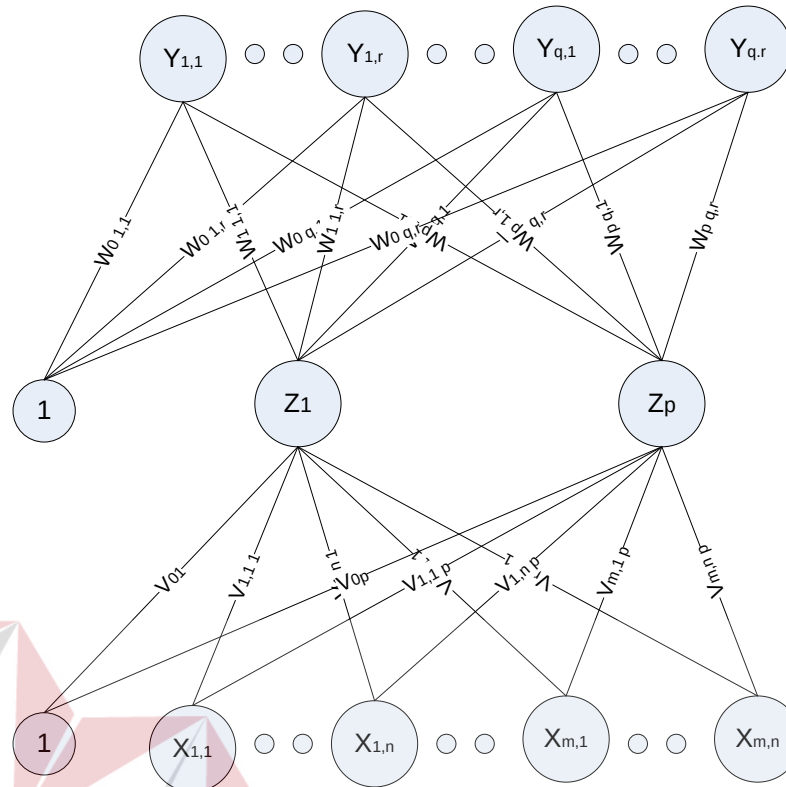
Sebelum hero rekomendasi ditampilkan ke tim 2, proses seleksi dibutuhkan untuk mengeliminasi *hero* yang termasuk dalam daftar *unavailable hero* dan *hero* yang telah dipilih sebelumnya.

5. Evaluasi Pick hero tim 2

Proses evaluasi adalah proses terakhir digunakan untuk menambahkan informasi *combo hero* dan *counter hero* terhadap rekomendasi hero yang diberikan pada tim 2.

3.2.1 Arsitektur ANN





Gambar 3.2. Gambar Arsitektur NN Ban Phase 1

x Input vektor pelatihan : $x = (x_1, \dots, x_i, \dots, x_n)$.

t Output vektor target : $t = (t_1, \dots, t_i, \dots, t_n)$.

$\delta_{q,r}$ Bagian dari penyesuaian bobot koreksi error untuk $w_{q,r}$ yang dikarenakan suatu error dari unit output $Y_{q,r}$; jadi informasi tentang error di unit $Y_{q,r}$ yang dikembalikan ke hidden unit itu dimasukkan kedalam unit $Y_{q,r}$.

$\delta_{m,n}$ Bagian dari penyesuaian bobot koreksi error untuk $v_{m,n}$ yang dikarenakan informasi yang salah backpropagation dari layer output ke unit hidde Z_p .

α Konstanta pembelajaran

$X_{m,n}$ Unit input $x_{m,n}$:

m untuk mewakili jumlah hero input

n untuk mewakili jumlah atribut hero input dengan total 13 atribut

Sinyal input (aktivasi) dari $X_{m,n}$ dinotasikan dengan $x_{m,n}$ dimana menggunakan fungsi aktivasi identity

$$x_{m,n} = f(x_{in\ m,n}).$$

v_{op} Bias pada unit hidden p .

Z_p Unit hidden p : Input net ke Z_p dinotasikan dengan z_{inp} :

$$z_{inp} = v_{op} + \sum x_{m,n} v_{m,n\ p}$$

Sinyal output (aktivasi) dari Z_p dinotasikan dengan z_p dimana menggunakan fungsi aktivasi Bipolar Sigmoid, karena dengan fungsi tersebut dapat memberikan efek non-linearitas dalam perhitungan sehingga mempercepat proses pelatihan.

$$z_p = f(z_{inp})$$

$$f(z_{inp}) = \frac{1 - \exp(-z_{inp})}{1 + \exp(-z_{inp})}$$

f adalah fungsi sigmoid

Z_{inp} Unit hidden

$w_{o\ q,r}$ Bias pada unit output $Y_{q,r}$.

$Y_{q,r}$ Unit output q,r

q untuk mewakili jumlah hero output

r untuk mewakili jumlah atribut hero output dengan total 13 atribut

Input net ke $Y_{q,r}$ dinotasikan dengan $y_{in\ q,r}$:

$$y_{in\ q,r} = w_{o\ q,r} + \sum z_p w_{p\ q,r}$$

Sinyal output (aktivasi) dari $Y_{q,r}$ dinotasikan dengan $y_{q,r}$ dimana menggunakan fungsi aktivasi identity

$$y_{q,r} = f(y_{in\ q,r}).$$

Secara lebih rinci algoritma backpropagation adalah sebagai berikut :

Step 0. Inisialisasi bobot.

(Set ke nilai random kecil).

Step 1. Ketika kondisi berhenti adalah salah, kerjakan langkah 2-9.

Step 2. Untuk tiap pasang pelatihan, kerjakan langkah 3-8.

Step 3. Tiap unit input ($X_{m,n}$, $m = 1, \dots, x$, $n = 1, \dots, 13$) menerima sinyal input $x_{m,n}$ dan meneruskan sinyal ini ke semua unit pada layer yang bersangkutan (unit hidden).

Step 4. Tiap unit hidden (Z , $p = 1, \dots, p$) dijumlahkan dengan bobot sinyal input.

$$z_{in_p} = v_{op} + \sum x_{m,n} v_{m,n\ p}$$

pergunakan fungsi aktivasi ini untuk menghitung sinyal output,

$$z_p = f(z_{in_p}).$$

dan mengirimkan sinyal ini ke seluruh unit pada layer yang bersangkutan (unit output).

Step 5. Tiap unit output ($Y_{q,r}$, $q = 1, \dots, x$, $r = 1, \dots, 13$) dijumlahkan dengan bobot sinyal input,

$$y_{in\ q,r} = w_{o\ q,r} + \sum z_p w_{p\ q,r}$$

dan menggunakan fungsi aktivasi ini untuk menghitung sinyal output,

$$y_{q,r} = f(y_{in_{q,r}}).$$

Error dari backpropagation :

Step 6. Tiap unit output ($Y_{q,r}$, $q = 1, \dots, x$, $r = 1, \dots, 13$) menerima satu pola target yang cocok untuk pola pelatihan input, syarat perhitungan informasi error ini,

$$\delta_{q,r} = (t_{q,r} - y_{q,r}) f'(y_{in_{q,r}}),$$

syarat penghitungan bobot koreksi ini (digunakan untuk meng-update w_{jk} nantinya),

$$\Delta w_{p_{q,r}} = \alpha \delta_{q,r} z_p,$$

syarat penghitungan bias koreksi ini (digunakan untuk meng-update $w_{o_{q,r}}$ nantinya),

$$\Delta w_{o_{q,r}} = \alpha \delta_{q,r},$$

dan mengirimkan $\delta_{q,r}$ ke unit pada layer sebelumnya.

Step 7. Tiap unit hidden ($Y_{q,r}$, $q = 1, \dots, x$, $r = 1, \dots, 13$) menjumlahkan ke input delta (dari unit pada layer sesudahnya),

$$\delta_{in_p} = \sum \delta_{Y_{q,r}} w_{p_{q,r}},$$

kalikan dengan nilai dari fungsi aktivasi untuk menghitungnya syarat informasi error,

$$\delta_p = \delta_{in_p} f'(z_{in_p}),$$

hitung koreksi bobot syaratnya (nanti digunakan untuk meng-update $v_{m,n p}$

$$\Delta v_{m,n p} = \alpha \delta_p x_{m,n}$$

dan hitung koreksi bias syaratnya (nanti digunakan untuk meng-update v_{oj}),

$$\Delta v_{op} = \alpha \delta_p.$$

Meng-update bobot dan bias :

Step 8. Tiap unit output ($Y_{q,r}$, $q = 1, \dots, x$, $r = 1, \dots, 13$) update bias dan bobot ($p = 1, \dots, p$) :

$$w_{p q,r} (\text{baru}) = w_{p q,r} (\text{lama}) + \Delta w_{p q,r}$$

Tiap unit hidden (Z_p , $p = 1, \dots, p$) update bias dan bobot ($i = 0, \dots, n$) :

$$v_{m,n p} (\text{baru}) = v_{m,n p} (\text{lama}) + \Delta v_{m,n p}.$$

Step 9. Kondisi pemberhentian test.

Setelah pelatihan, suatu jaringan neural digunakan hanya pada fase feedforward algoritma pelatihan. Prosedur aplikasinya meliputi antara lain :

Step 0. Inisialisasi bobot (dari algoritma pelatihan)

Step 1. Untuk tiap vektor input, kerjakan step 2-4.

Step 2. Untuk $m, n = 1, \dots, n$; set aktivasi dari unit input $x_{m,n}$;

Step 3. Untuk $p = 1, \dots, p$:

$$z_in_p = v_{op} + \sum x_{m,n} v_{m,n p};$$

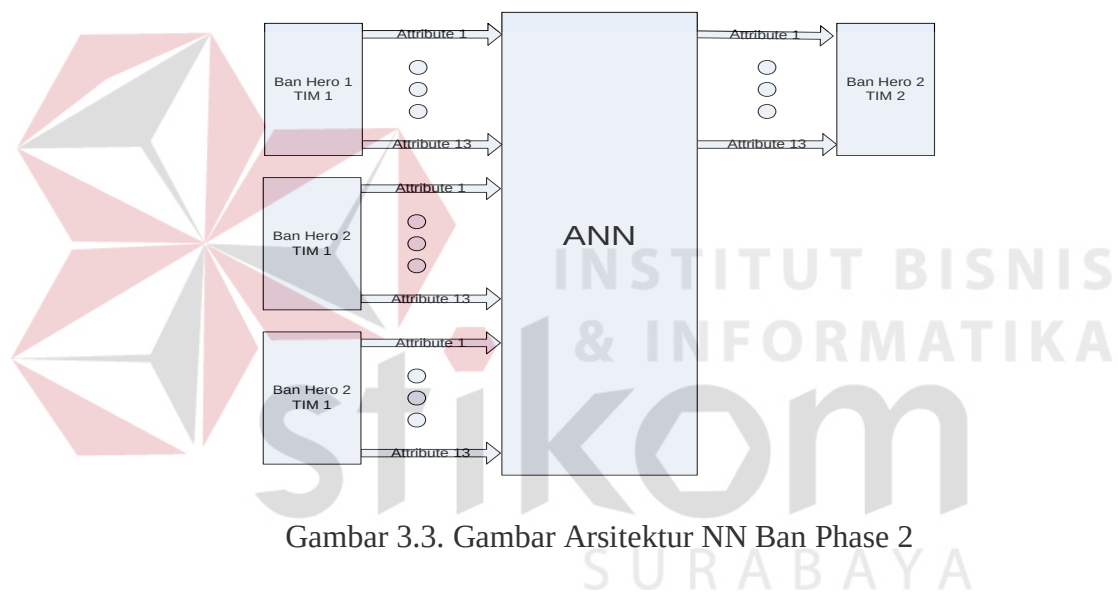
$$z_p = f(z_in_p),$$

Step 4. Untuk q, r , $q = 1, \dots, x$, $r = 1, \dots, 13$:

$$y_{in\ q,r} = w_{o\ q,r} + \sum z_p w_{p\ q,r}$$

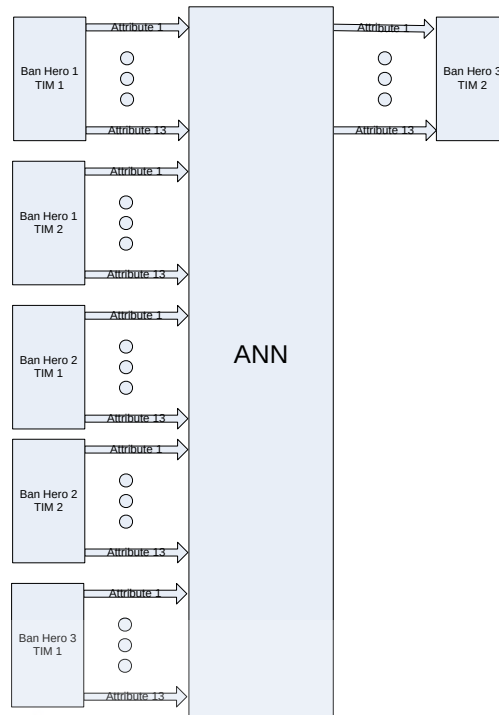
$$y_{q,r} = f(y_{in\ q,r}).$$

Pada ban phase yang pertama, Proses ANN menerima hasil input ban hero dari tim 1 yang sudah di konversikan menjadi atribut. Di dalam proses ANN sistem mencoba memberikan saran penentuan hero yang berupa output untuk diban selanjutnya kepada tim 2



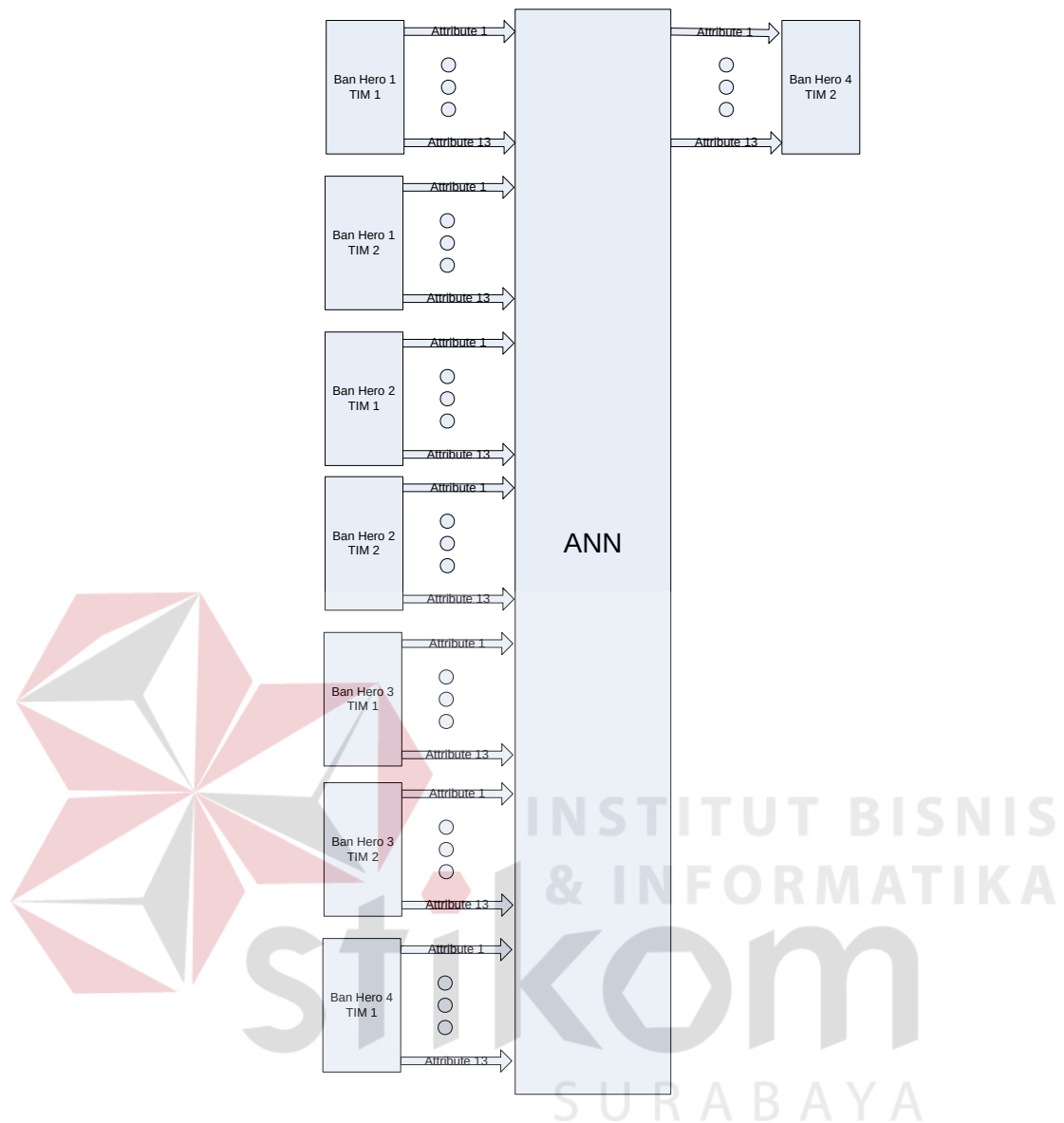
Gambar 3.3. Gambar Arsitektur NN Ban Phase 2

Proses selanjutnya adalah ban phase yang kedua, dimana hasil output saran dari ban phase 1 dijadikan input untuk proses ANN bersama dengan hero yang diban kedua oleh tim 1. Hasilnya adalah saran untuk tim 2 hero yang akan di ban.



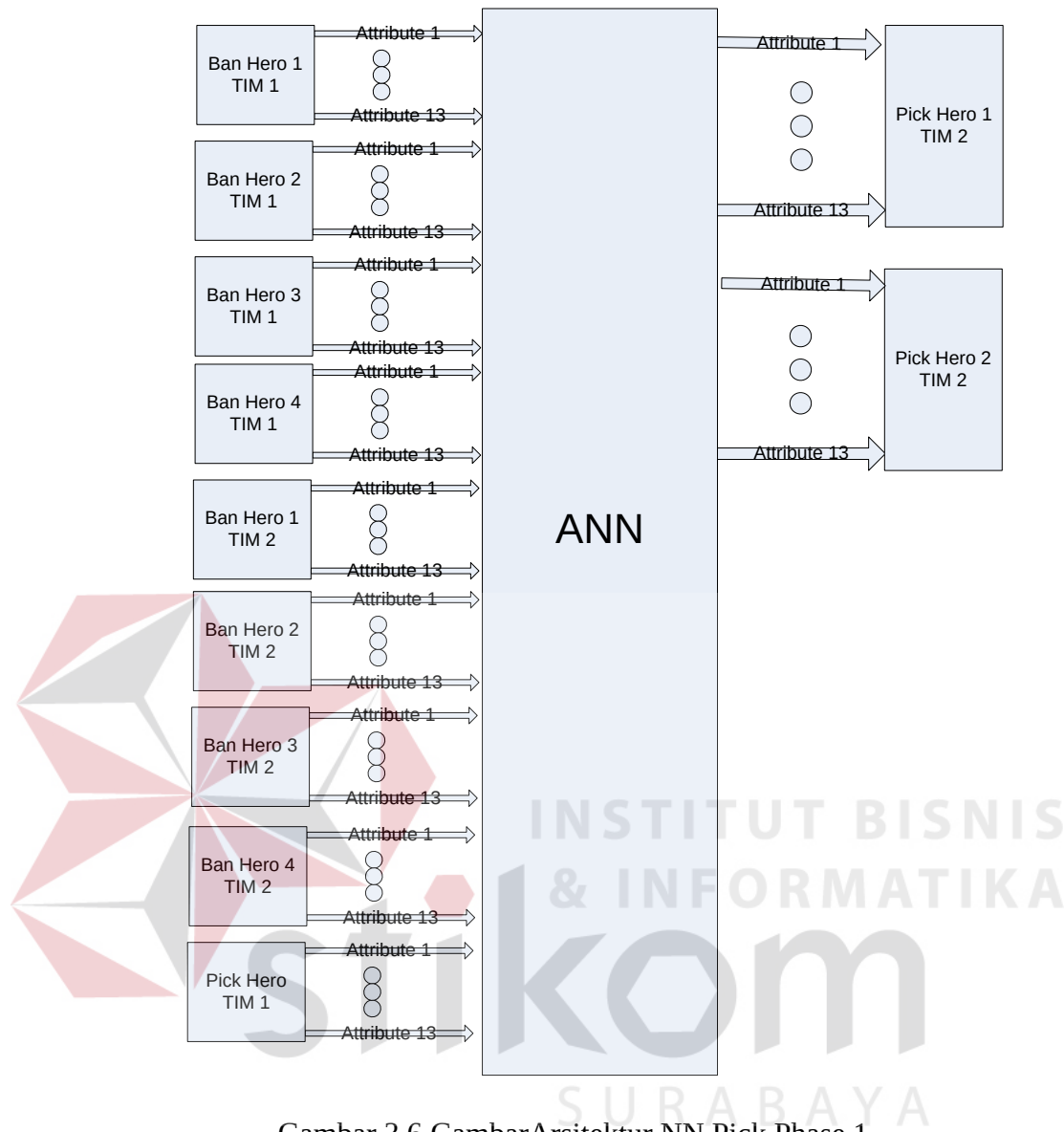
Gambar 3.4. GambarArsitektur NN Ban Phase 3

Pada ban phase ke 3 outputnya adalah saran untuk memilih hero yang akan diban tim 2 setelah di proses ANN dengan 3 inputan hero yang diban tim 1 dan 2 hero yang diban dari tim 2.



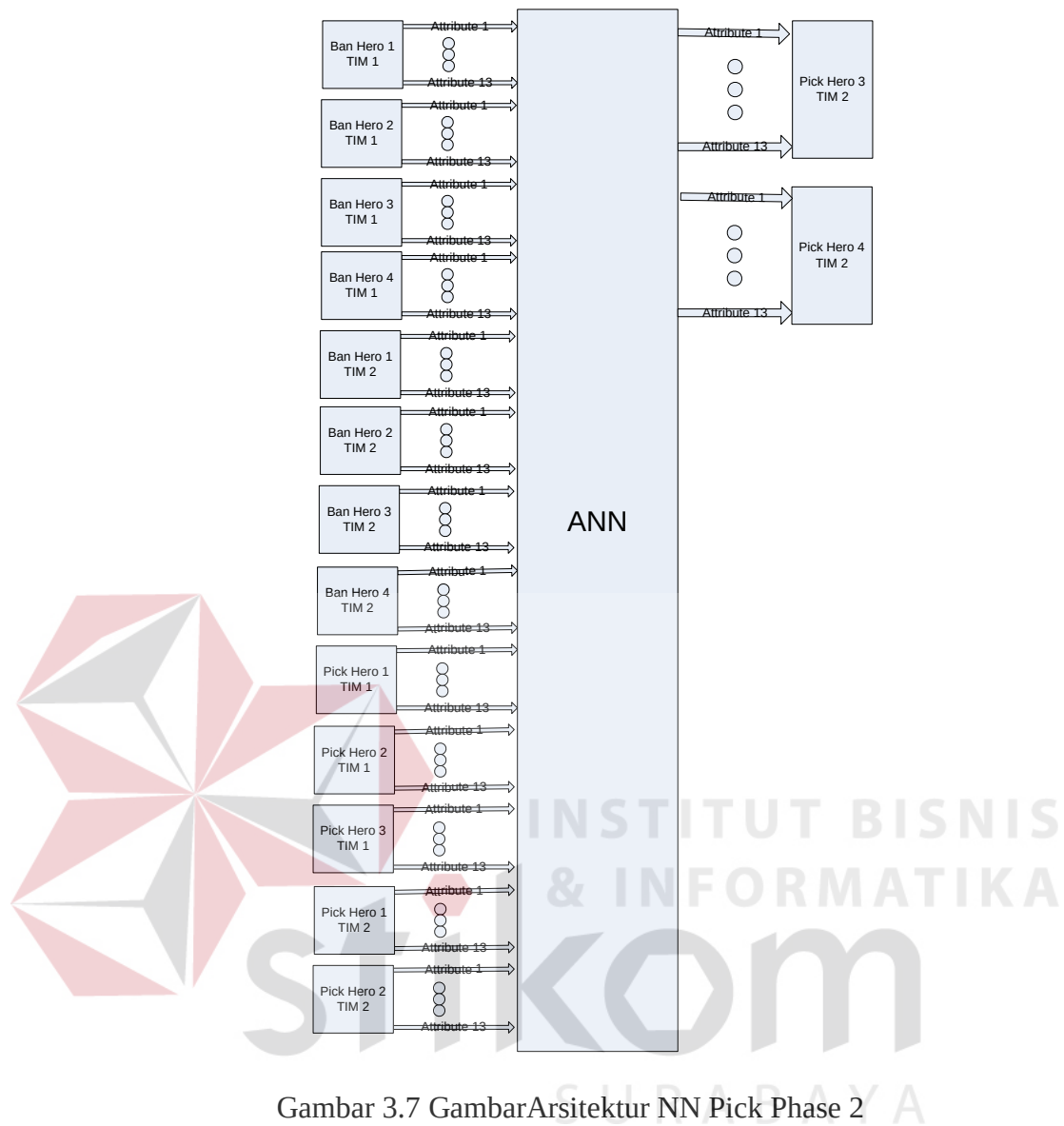
Gambar 3.5. GambarArsitektur NN Ban Phase 4

Proses ban phase ke 4 hampir sama dengan proses ban phase sebelumnya, outputnya adalah saran hero yang akan di ban terakhir oleh tim 2 berdasarkan proses ANN yang telah diolah dengan hasil input dari 4 hero yang diban tim 1 dan 3 hero dari tim 2.



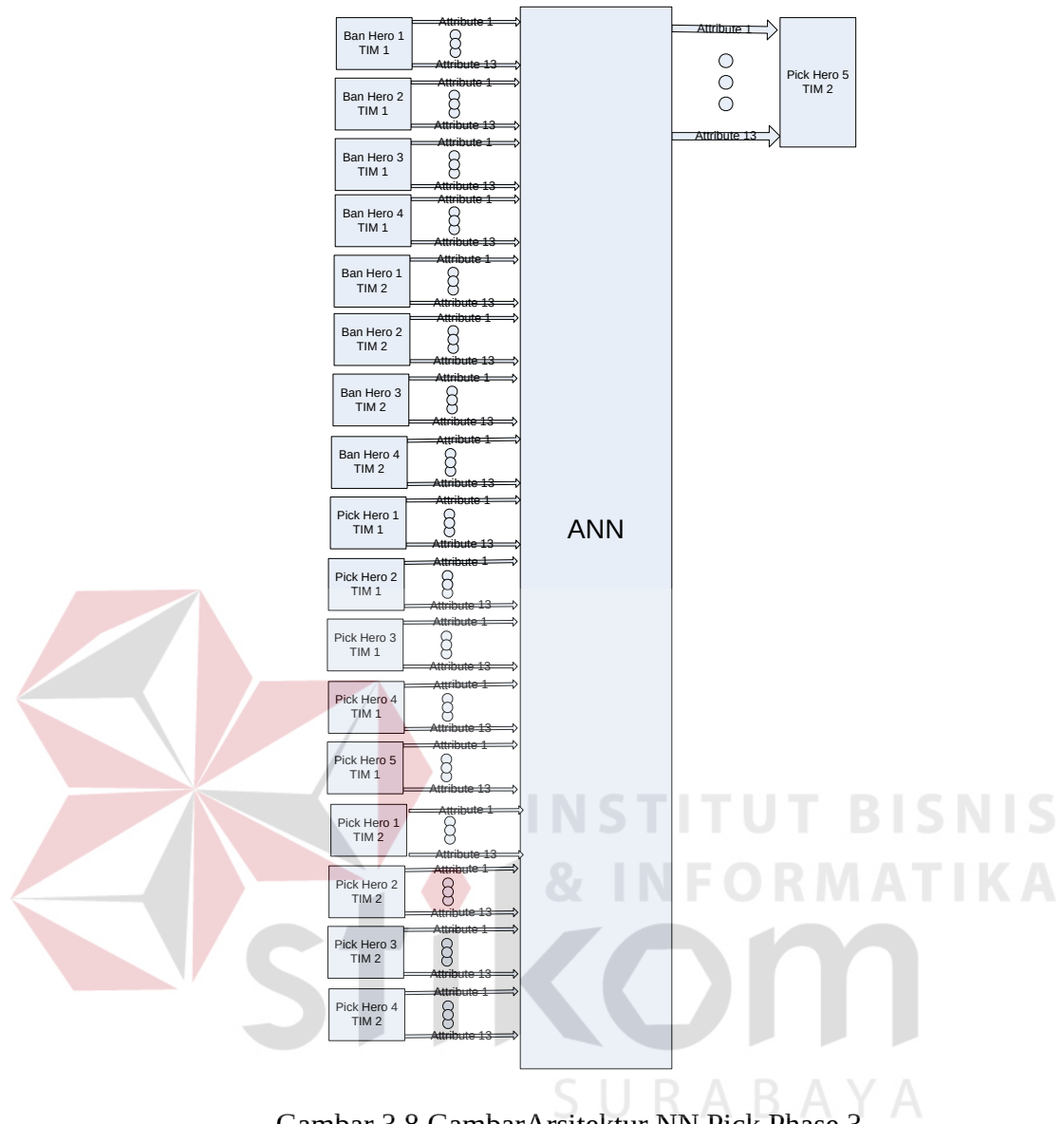
Gambar 3.6 GambarArsitektur NN Pick Phase 1

Proses ANN pada saat pick phase 1 menghasilkan 2 output saran untuk pemilihan hero tim 2, pick hero 1 dan pick hero 2. Proses ANN mengolah semua input hasil ban phase tim 1 dan 2 kemudia ditambahkan pick hero yang pertama oleh tim 1.



Gambar 3.7 GambarArsitektur NN Pick Phase 2

Demikian juga dengan pick phase ke 2, output dari proses ANN ada 2 yaitu saran pemilihan hero pada pick hero 3 dan pick hero 4 untuk tim 2. Data yang diinputkan pada prosesa ANN adalah semua ban hero tim 1 dan tim 2, pick hero tim 1 sampai dengan pick hero yang ke 3, tim 2 pick hero 1 dan pick hero 2.



Gambar 3.8 GambarArsitektur NN Pick Phase 3

Proses NN terakhir mengeluarkan output pick hero yang ke 5 bagi tim 2. Data yang diinput untuk proses NN kali ini dari semua ban hero tim 1 dan tim 2, pick hero tim 1 sampai dengan pick hero yang ke 5 sedangkan tim 2 sampai dengan pick hero 4

3.2.2 Contoh Perhitungan ANN

Sebagai contoh perhitungan ANN pada sistem ini akan diambil proses pelatihan pada Ban Phase 2. Sebagaimana telah dijelaskan sebelumnya pada

mekanisme pemilihan hero DotA untuk *Captains Mode* (sub bab 2.4), Ban Phase 2 akan mendapatkan input masing-masing 1 hero yang dipilih untuk tidak boleh dimainkan oleh tim 1 dan tim 2.

Pelatihan ANN akan bergantung pada konfigurasi ANN yang telah ditetapkan sebelumnya. Misal untuk pelatihan Ban phase 2, ditetapkan konfigurasi ANN sebagai berikut :

NN ID : Ban Phase 2

Input Group Quantity : 3

I/O Nodes Quantity : 13

Output Group Quantity : 1

Hidden Nodes Quantity : 3

Learning Const : 0,001

Treshold : 0

Learning Pattern : 10

Grup ID Input : 1

Neuron Quantity : 13

Activation : Identity

Object ID : B2

Max Value : 1

Min Value : 0

Grup ID Input : 2

Neuron Quantity : 13

Activation : Identity

Object ID : B2

Max Value : 1

Min Value : 0

Grup ID Input : 3

Neuron Quantity : 13

Activation : Identity

Object ID : B2

Max Value : 1

Min Value : 0

Grup ID Output : 1

Neuron Quantity : 13

Activation : Identity

Object ID : B2

Max Value : 1

Min Value : 0

Berdasarkan pada jumlah pola data (Learning Pattern) yang telah ditentukan sebelumnya, misal 1 pola sebagaimana disebutkan di atas, maka sistem akan menyediakan 1 pola data dari permainan sebelumnya yang telah disimpan di database (tabel 3.1)

Tabel 3.1 Pola data input pada Ban Phase 2

Ban Phase 2			
Pola Ke	INPUT		
	Hero 1 tim 1	Hero 1 tim 2	Hero 2 tim 1

	ID	Nama	ID	Nama	ID	Nama
1	51	Vengeful Spirit	61	Spectre	88	Dark Seer

Tabel 3.2 Pola data target pada Ban Phase 2

Pola Ke	TARGET	
	Hero 2 tim 2	
	ID	Nama
1	78	Visage

Saat pelatihan ANN dimulai, pola data pertama akan diumpankan ke ANN, dan sebelum diproses lebih lanjut ke perhitungan ANN, data akan dikonversi terlebih dahulu menjadi 13 atribut berdasarkan pada atribut dari ID hero bersangkutan yang telah ada di tabel MstHeroesAttributes pada database .

Misal ID hero 51 Vengeful Spirit akan memiliki atribut sebagai berikut :

Tanker : 0

Ganker : 1

Initiator : 1

Pusher : 0

Disabler : 0

AOE Stunner : 0

AOE Nuker : 0

Nuker : 0

Support : 1

Semi Carry : 0

Carry / DPS : 0

Stunner : 1

Slower : 0

Demikian seterusnya untuk pola data input dan output dari ID hero yang ada pada Tabel 3.1, akan di konversi menjadi 13 atribut sebagaimana pada tabel 3.2 dan 3.3

Tabel 3.2 Pola Input Attribute Hero Ban Phase 2

Pola input	ID	Attributes												
		1	2	3	4	5	6	7	8	9	10	11	12	13
1	51	0	0	1	0	0	0	0	0	1	0	0	1	0
	61	0	0	1	0	0	0	0	0	0	0	1	0	1
	88	0	1	0	1	1	0	1	0	1	0	0	0	0

Tabel 3.3 Pola Target Attribute Hero Ban Phase 2

Pola Target	ID	Attributes												
		1	2	3	4	5	6	7	8	9	10	11	12	13
1	78	1	0	0	0	0	0	0	1	0	1	0	0	1

Setelah persiapan pola input dan target selesai maka proses pelatihan dimulai dengan melakukan proses forward propagation. Pada awal pelatihan bobot akan di beri nilai random. Perhitungan akan dimulai dengan set iterasi pertama dengan mengumpulkan pola input di tabel 3.2 pada proses forward propagation dan mengakumulasi perubahan bobot yang terjadi pada proses backpropagation hingga iterasi selesai, baru melakukan update bobot.

Berikut ini adalah perhitungan forward propagation pada iterasi awal :

Tiap unit hidden dijumlahkan dengan bobot sinyal input.

$z_in_p = v_{op} + \sum x_{m,n} v_{m,n p}$, dimana pada perhitungan ini tidak ada node bias untuk layer input jadi nilai $v_{op} = 0$.

$z_in_1 = (x_{1,1} v_{1,1 1} + \dots + x_{1,13} v_{1,13 1} + x_{2,1} v_{2,1 1} + \dots + x_{2,13} v_{2,13 1} + x_{3,1} v_{3,1 1} + \dots + x_{3,13} v_{3,13 1})$

$$z_in_1 = (0 \cdot 0.00531129702112896 + \dots + 0 \cdot 0.00504173658580013 + 0 \cdot 0.000875272066938814 + \dots + 1 \cdot 0.0108366005700201 + 0 \cdot 0.00184866986176721 + \dots + 0 \cdot 0.00248305944621586) = -0.0133280446980434$$

$$z_in_2 = (x_{1,1} v_{1,1 \cdot 2} + \dots + x_{1,13} v_{1,13 \cdot 2} + x_{2,1} v_{2,1 \cdot 2} + \dots + x_{2,13} v_{2,13 \cdot 2} + x_{3,1} v_{3,1 \cdot 2} + \dots + x_{3,13} v_{3,13 \cdot 2})$$

$$z_in_2 = (0 \cdot 0.000863668431241366 + \dots + 0 \cdot 1.24031653173033 + 0 \cdot 0.0101713105683963 + \dots + 1 \cdot 0.00340419159498733 + 0 \cdot -0.0103882572794006 + \dots + 0 \cdot 0.00655523787931578) = 0.0418224324556603$$

$$z_in_2 = (x_{1,1} v_{1,1 \cdot 2} + \dots + x_{1,13} v_{1,13 \cdot 2} + x_{2,1} v_{2,1 \cdot 2} + \dots + x_{2,13} v_{2,13 \cdot 2} + x_{3,1} v_{3,1 \cdot 2} + \dots + x_{3,13} v_{3,13 \cdot 2})$$

$$z_in_3 = (x_{1,1} v_{1,1 \cdot 3} + \dots + x_{1,13} v_{1,13 \cdot 3} + x_{2,1} v_{2,1 \cdot 3} + \dots + x_{2,13} v_{2,13 \cdot 3} + x_{3,1} v_{3,1 \cdot 2} + \dots + x_{3,13} v_{3,13 \cdot 3})$$

$$z_in_3 = (0 \cdot 0.00205474143716503 + \dots + 0 \cdot -0.00661675070200232 + 0 \cdot 0.0129047491811626 + \dots + 1 \cdot 0.00329823872658614 + 0 \cdot 0.00157726122324792 + \dots + 0 \cdot 0.00248305944621586) = -0.00359495388729084$$

Nilai z dihitung dari nilai z_in berdasarkan pada fungsi aktivasi yang telah ditetapkan untuk tiap z .

$$z_1 = f(z_in_1) = (1 - \exp(-z_in_1)) / (1 + \exp(-z_in_1)) = -0.00675333529307017$$

$$z_2 = f(z_in_2) = (1 - \exp(-z_in_2)) / (1 + \exp(-z_in_2)) = 0.0200517714084377$$

$$z_3 = f(z_in_3) = (1 - \exp(-z_in_3)) / (1 + \exp(-z_in_3)) = -0.00180394847651682$$

Secara ringkas perhitungan Z dapat dilihat pada tabel 3.4

Tabel 3.4 Perhitungan forward propagation nilai hidden layer z

i		x_i	h	$v_{i,h}$	$x_i \cdot v_{i,h}$	z_{in_h}	z_h
1	1	0	1	5.31E-03	0.00E+00	-1.33E-02	-6.75E-03
1	2	0	1	-5.44E-03	0.00E+00		
1	3	1	1	-1.26E-02	-1.26E-02		
1	4	0	1	5.40E-03	0.00E+00		
1	5	0	1	9.37E-03	0.00E+00		
1	6	0	1	1.19E-02	0.00E+00		
1	7	0	1	1.16E-02	0.00E+00		
1	8	0	1	6.90E-03	0.00E+00		
1	9	1	1	-8.09E-04	-8.09E-04		
1	10	0	1	3.82E-03	0.00E+00		
1	11	0	1	8.52E-03	0.00E+00		
1	12	1	1	1.26E-02	1.26E-02		
1	13	0	1	5.04E-03	0.00E+00		
2	1	0	1	8.75E-04	0.00E+00		
2	2	0	1	4.55E-03	0.00E+00		
2	3	1	1	-1.03E-02	-1.03E-02		
2	4	0	1	-5.57E-03	0.00E+00		
2	5	0	1	-3.05E-03	0.00E+00		
2	6	0	1	1.24E-02	0.00E+00		
2	7	0	1	-8.77E-03	0.00E+00		
2	8	0	1	-2.32E-03	0.00E+00		
2	9	0	1	-4.49E-03	0.00E+00		
2	10	0	1	-8.11E-03	0.00E+00		
2	11	1	1	-1.09E-03	-1.09E-03		
2	12	0	1	7.37E-03	0.00E+00		
2	13	1	1	1.08E-02	1.08E-02		
3	1	0	1	-1.85E-03	0.00E+00		
3	2	1	1	5.03E-03	5.03E-03		
3	3	0	1	-1.23E-02	0.00E+00		
3	4	1	1	-1.80E-03	-1.80E-03		
3	5	1	1	3.55E-04	3.55E-04		
3	6	0	1	-2.46E-03	0.00E+00		
3	7	1	1	-6.62E-03	-6.62E-03		
3	8	0	1	-2.83E-03	0.00E+00		
3	9	1	1	-8.90E-03	-8.90E-03		
3	10	0	1	3.33E-03	0.00E+00		
3	11	0	1	1.13E-02	0.00E+00		
3	12	0	1	-2.83E-03	0.00E+00		
3	13	0	1	-1.04E-03	0.00E+00		
1	1	0	2	8.64E-04	0.00E+00	4.18E-02	2.01E-02
1	2	0	2	-5.12E-03	0.00E+00		
1	3	1	2	6.74E-03	6.74E-03		
1	4	0	2	-1.17E-02	0.00E+00		
1	5	0	2	7.51E-03	0.00E+00		
1	6	0	2	9.60E-03	0.00E+00		
1	7	0	2	-3.51E-03	0.00E+00		

Tabel 3.4 Perhitungan forward propagation nilai hidden layer z (lanjutan)

i		x_i	h	$v_{i,h}$	$x_i \cdot v_{i,h}$	z_{in_h}	z_h
1	8	0	2	-1.15E-02	0.00E+00	4.18E-02	2.01E-02
1	9	1	2	-5.22E-03	-5.22E-03		
1	10	0	2	-6.10E-03	0.00E+00		
1	11	0	2	8.39E-03	0.00E+00		
1	12	1	2	1.06E-02	1.06E-02		
1	13	0	2	1.24E+00	0.00E+00		
2	1	0	2	-1.02E-02	0.00E+00		
2	2	0	2	-1.25E-02	0.00E+00		
2	3	1	2	-1.03E-02	-1.03E-02		
2	4	0	2	-1.17E-02	0.00E+00		
2	5	0	2	-5.14E-03	0.00E+00		
2	6	0	2	-2.55E-03	0.00E+00		
2	7	0	2	-8.71E-03	0.00E+00		
2	8	0	2	-2.25E-03	0.00E+00		
2	9	0	2	3.44E-03	0.00E+00		
2	10	0	2	2.15E-03	0.00E+00		
2	11	1	2	1.05E-02	1.05E-02		
2	12	0	2	-3.13E-03	0.00E+00		
2	13	1	2	3.40E-03	3.40E-03		
3	1	0	2	-1.04E-02	0.00E+00		
3	2	1	2	1.07E-02	1.07E-02		
3	3	0	2	1.12E-03	0.00E+00		
3	4	1	2	4.60E-03	4.60E-03		
3	5	1	2	-9.57E-04	-9.57E-04		
3	6	0	2	-5.95E-03	0.00E+00		
3	7	1	2	1.24E-02	1.24E-02		
3	8	0	2	-3.49E-03	0.00E+00		
3	9	1	2	-6.60E-04	-6.60E-04		
3	10	0	2	1.09E-03	0.00E+00		
3	11	0	2	3.99E-03	0.00E+00		
3	12	0	2	-1.01E-02	0.00E+00		
3	13	0	2	6.56E-03	0.00E+00		
1	1	0	3	2.05E-03	0.00E+00	-3.59E-03	-1.80E-03
1	2	0	3	7.10E-03	0.00E+00		
1	3	1	3	8.13E-03	8.13E-03		
1	4	0	3	-2.22E-03	0.00E+00		
1	5	0	3	-3.27E-03	0.00E+00		
1	6	0	3	-1.15E-02	0.00E+00		
1	7	0	3	6.43E-04	0.00E+00		
1	8	0	3	2.39E-03	0.00E+00		
1	9	1	3	3.17E-03	3.17E-03		
1	10	0	3	-5.70E-03	0.00E+00		
1	11	0	3	2.30E-03	0.00E+00		
1	12	1	3	-7.06E-03	-7.06E-03		
1	13	0	3	-6.62E-03	0.00E+00		
2	1	0	3	1.29E-02	0.00E+00		

Tabel 3.4 Perhitungan forward propagation nilai hidden layer z (lanjutan)

i	x_i	h	$v_{i,h}$	$x_i \cdot v_{i,h}$	z_{in_h}	z_h
---	-------	---	-----------	---------------------	------------	-------

2	2	0	3	1.94E-03	0.00E+00	-3.59E-03	-1.80E-03
2	3	1	3	7.72E-03	7.72E-03		
2	4	0	3	-5.28E-03	0.00E+00		
2	5	0	3	1.16E-02	0.00E+00		
2	6	0	3	-5.73E-03	0.00E+00		
2	7	0	3	3.79E-03	0.00E+00		
2	8	0	3	5.50E-03	0.00E+00		
2	9	0	3	-7.56E-03	0.00E+00		
2	10	0	3	-1.08E-02	0.00E+00		
2	11	1	3	-6.17E-03	-6.17E-03		
2	12	0	3	-5.44E-03	0.00E+00		
2	13	1	3	3.30E-03	3.30E-03		
3	1	0	3	1.58E-03	0.00E+00	-3.59E-03	-1.80E-03
3	2	1	3	8.65E-03	8.65E-03		
3	3	0	3	1.08E-02	0.00E+00		
3	4	1	3	6.34E-05	6.34E-05		
3	5	1	3	-3.79E-03	-3.79E-03		
3	6	0	3	-1.15E-02	0.00E+00		
3	7	1	3	-1.13E-02	-1.13E-02		
3	8	0	3	-2.61E-04	0.00E+00		
3	9	1	3	-6.27E-03	-6.27E-03		
3	10	0	3	-8.88E-03	0.00E+00		
3	11	0	3	1.57E-04	0.00E+00		
3	12	0	3	7.34E-03	0.00E+00		
3	13	0	3	2.48E-03	0.00E+00		

$$y_{in1} = 1 \cdot w_{0,1} + z_{1,1} w_{1,1} + z_{2,1} w_{2,1} + z_{3,1} w_{3,1} = -1 \cdot -0.426046669483185 + -0.00675333529307017 \cdot -0.394547343254089 + 0.0200517714084377 \cdot -0.168305575847626 + -0.00180394847651682 \cdot -0.371750116348267 = -0.42609$$

$y_1 = f(y_{in1})$ dimana fungsi aktivasi dari y adalah fungsi linier, jadi

$$y_1 = -0.42609$$

Demikian seterusnya sampai y_{13} sebagaimana pada tabel 3.5

Tabel 3.5 Perhitungan forward propagation nilai output layer y

h	o	z_h	$w_{h,o}$	$w_{h,o} \cdot z_h$	y_{in_o}	y_o
0	1	1	-0.42605	-0.426046669	-0.42609	-0.42609
1	1	-6.75E-03	-0.39455	2.66E-03		
2	1	2.01E-02	-0.16831	-3.37E-03		
3	1	-1.80E-03	-0.37175	6.71E-04		
0	2	1	-0.49976	-0.49975878	-0.49694	-0.49694
1	2	-6.75E-03	3.68E-02	-2.48E-04		
2	2	2.01E-02	0.157055	3.15E-03		
3	2	-1.80E-03	4.40E-02	-7.94E-05		
0	3	1	0.327412	0.32741183	0.323735	0.323735
1	3	-6.75E-03	-0.41811	2.82E-03		
2	3	2.01E-02	-0.30808	-6.18E-03		
3	3	-1.80E-03	0.178913	-3.23E-04		
0	4	1	-4.58E-02	-0.045792162	-0.05221	-0.05221
1	4	-6.75E-03	-0.14298	9.66E-04		
2	4	2.01E-02	-0.35002	-7.02E-03		
3	4	-1.80E-03	0.204396	-3.69E-04		
0	5	1	0.428786	0.42878598	0.419889	0.419889
1	5	-6.75E-03	3.02E-02	-2.04E-04		
2	5	2.01E-02	-0.41036	-8.23E-03		
3	5	-1.80E-03	0.257729	-4.65E-04		
0	6	1	-9.82E-02	-0.098157942	-0.09753	-0.09753
1	6	-6.75E-03	-3.81E-02	2.57E-04		
2	6	2.01E-02	-7.83E-03	-1.57E-04		
3	6	-1.80E-03	-0.29237	5.27E-04		
0	7	1	-0.17026	-0.170264065	-0.16514	-0.16514
1	7	-6.75E-03	-0.40457	2.73E-03		
2	7	2.01E-02	8.98E-02	1.80E-03		
3	7	-1.80E-03	-0.33013	5.96E-04		
0	8	1	0.427617	0.427616537	0.429616	0.429616
1	8	-6.75E-03	-0.40207	2.72E-03		
2	8	2.01E-02	-5.61E-02	-1.13E-03		
3	8	-1.80E-03	-0.22705	4.10E-04		
0	9	1	0.372547	0.372546852	0.365988	0.365988
1	9	-6.75E-03	0.250688	-1.69E-03		
2	9	2.01E-02	-0.22706	-4.55E-03		
3	9	-1.80E-03	0.173647	-3.13E-04		
0	10	1	-0.24337	-0.243371069	-0.24969	-0.24969
1	10	-6.75E-03	-0.4101	2.77E-03		
2	10	2.01E-02	-0.46905	-9.41E-03		
3	10	-1.80E-03	-0.17728	3.20E-04		
0	11	1	0.290129	0.290128887	0.286225	0.286225
1	11	-6.75E-03	-0.20274	1.37E-03		
2	11	2.01E-02	-2.65E-01	-5.31E-03		
3	11	-1.80E-03	-1.95E-02	3.52E-05		

Tabel 3.5 Perhitungan forward propagation nilai output layer y (lanjutan)

h	o	z_h	$w_{h,o}$	$w_{h,o} \cdot z_h$	y_{in_o}	y_o
0	12	1	-0.2454	-0.245398104	-0.25341	-0.25341
1	12	-6.75E-03	-0.15939	1.08E-03		
2	12	2.01E-02	-0.45507	-9.12E-03		
3	12	-1.80E-03	-1.76E-02	3.17E-05		
0	13	1	-0.29398	-0.293982804	-0.29513	-0.29513
1	13	-6.75E-03	0.364534	-2.46E-03		
2	13	2.01E-02	8.86E-02	1.78E-03		
3	13	-1.80E-03	0.254908	-4.60E-04		

Backpropagation dimulai dengan menghitung δ output NN yang merupakan hasil dari perkalian error NN, yaitu target (t) – output (y), dengan turunan fungsi aktivasi output $f'(y_{in})$. Karena fungsi aktivasi dari output adalah linier, maka turunan dari fungsi aktivasi output adalah 1.

Pada contoh perhitungan ini:

$\delta y_1 = (t_1 - y_1) \times f'(y_{in_1}) = (1 - 0.4260863) \times 1 = 1.426086366$, demikian seterusnya untuk semua output(y) yang ada, sebagaimana ditampilkan pada tabel 3.6

Tabel 3.6 Perhitungan turunan fungsi aktivasi linear untuk output (δy)

o	y_o	t_o	e_o	$f'(y_o)$	δy_o
1	-0.42609	1	1.426086366	1	1.426086366
2	-0.49694	0	0.496937431	1	0.496937431
3	0.323735	0	-0.323735193	1	-0.323735193
4	-0.05221	0	0.05221381	1	0.05221381
5	0.419889	0	-0.419888597	1	-0.419888597
6	-0.09753	0	0.097530135	1	0.097530135
7	-0.16514	0	0.165135831	1	0.165135831
8	0.429616	0	-0.429615787	1	-0.429615787
9	0.365988	1	0.634012284	1	0.634012284
10	-0.24969	0	0.249686966	1	0.249686966
11	0.286225	0	-0.286225234	1	-0.286225234
12	-0.25341	0	0.25341485	1	0.25341485
13	-0.29513	1	1.295127309	1	1.295127309

Setelah nilai δy didapatkan perhitungan dilanjutkan dengan mencari nilai dari perubahan bobot hidden output Δw dengan rumus $\Delta w_{h,o} = \alpha \delta y_o z_h$, dimana

$\Delta w_{0,1} = 0.0001 \times 1.426086366 \times 1 = 0.000142609$, demikian seterusnya untuk perubahan bobot hidden output lainnya, sebagaimana dapat dilihat pada tabel 3.7

Tabel 3.7 Perhitungan perubahan bobot hidden output

h	o	z_h	d_o	a	$\Delta w_{h,o}$
0	1	1	1.426086366	0.0001	0.000142609
1	1	-6.75E-03	1.426086366	0.0001	-9.63084E-07
2	1	2.01E-02	1.426086366	0.0001	2.85956E-06
3	1	-1.80E-03	1.426086366	0.0001	-2.57259E-07
0	2	1	0.496937431	0.0001	4.96937E-05
1	2	-6.75E-03	0.496937431	0.0001	-3.35599E-07
2	2	2.01E-02	0.496937431	0.0001	9.96448E-07
3	2	-1.80E-03	0.496937431	0.0001	-8.9645E-08
0	3	1	-0.323735193	0.0001	-3.23735E-05
1	3	-6.75E-03	-0.323735193	0.0001	2.18629E-07
2	3	2.01E-02	-0.323735193	0.0001	-6.49146E-07
3	3	-1.80E-03	-0.323735193	0.0001	5.84002E-08
0	4	1	0.05221381	0.0001	5.22138E-06
1	4	-6.75E-03	0.05221381	0.0001	-3.52617E-08
2	4	2.01E-02	0.05221381	0.0001	1.04698E-07
3	4	-1.80E-03	0.05221381	0.0001	-9.4191E-09
0	5	1	-0.419888597	0.0001	-4.19889E-05
1	5	-6.75E-03	-0.419888597	0.0001	2.83565E-07
2	5	2.01E-02	-0.419888597	0.0001	-8.41951E-07
3	5	-1.80E-03	-0.419888597	0.0001	7.57457E-08
0	6	1	0.097530135	0.0001	9.75301E-06
1	6	-6.75E-03	0.097530135	0.0001	-6.58654E-08
2	6	2.01E-02	0.097530135	0.0001	1.95565E-07
3	6	-1.80E-03	0.097530135	0.0001	-1.75939E-08
0	7	1	0.165135831	0.0001	1.65136E-05
1	7	-6.75E-03	0.165135831	0.0001	-1.11522E-07
2	7	2.01E-02	0.165135831	0.0001	3.31127E-07
3	7	-1.80E-03	0.165135831	0.0001	-2.97897E-08
0	8	1	-0.429615787	0.0001	-4.29616E-05
1	8	-6.75E-03	-0.429615787	0.0001	2.90134E-07
2	8	2.01E-02	-0.429615787	0.0001	-8.61456E-07
3	8	-1.80E-03	-0.429615787	0.0001	7.75005E-08
0	9	1	0.634012284	0.0001	6.34012E-05
1	9	-6.75E-03	0.634012284	0.0001	-4.2817E-07
2	9	2.01E-02	0.634012284	0.0001	1.27131E-06
3	9	-1.80E-03	0.634012284	0.0001	-1.14373E-07
0	10	1	0.249686966	0.0001	2.49687E-05
1	10	-6.75E-03	0.249686966	0.0001	-1.68622E-07
2	10	2.01E-02	0.249686966	0.0001	5.00667E-07
3	10	-1.80E-03	0.249686966	0.0001	-4.50422E-08
0	11	1	-0.286225234	0.0001	-2.86225E-05
1	11	-6.75E-03	-0.286225234	0.0001	1.93297E-07

Tabel 3.7 Perhitungan perubahan bobot hidden output (lanjutan)

h	o	z_h	d_o	a	$\Delta w_{h,o}$
2	11	2.01E-02	-0.286225234	0.0001	-5.73932E-07
3	11	-1.80E-03	-0.286225234	0.0001	5.16336E-08
0	12	1	0.25341485	0.0001	2.53415E-05
1	12	-6.75E-03	0.25341485	0.0001	-1.7114E-07
2	12	2.01E-02	0.25341485	0.0001	5.08142E-07
3	12	-1.80E-03	0.25341485	0.0001	-4.57147E-08
0	13	1	1.295127309	0.0001	0.000129513
1	13	-6.75E-03	1.295127309	0.0001	-8.74643E-07
2	13	2.01E-02	1.295127309	0.0001	2.59696E-06
3	13	-1.80E-03	1.295127309	0.0001	-2.33634E-07

Setelah nilai Δw didapatkan, perhitungan dilanjutkan untuk mencari perubahan bobot input hidden Δv , yang dimulai dengan mencari masukan backpropagation dari hidden ke input, dengan rumus $\delta_{in_h} = \sum \delta_{y_o} w_{h,o}$, dan dilanjutkan dengan mencari nilai $\delta_h = \delta_{in_h} f'(z_{in_h})$, dimana turunan dari fungsi aktivasi node hidden yang menggunakan bipolar sigmoid dapat dicari dengan rumus $f'(z_{in_h}) = 0.5(1+f(z_{in_h}))(1-f(z_{in_h}))$, sebagaimana dapat dilihat pada tabel 3.8

Tabel 3.8 Perhitungan turunan fungsi aktivasi bipolar sigmoid untuk bobot (δ_h)

h	o	z_h	$w_{h,o}$	δy_o	$\delta y_o \cdot w_{h,o}$	δ_{in_h}	$f'(z_h)$	δ_h
1	1	-0.007	-0.395	1.426	-0.563	0.219	0.500	0.110
1	2	-0.007	0.037	0.497	0.018	0.219	0.500	0.110
1	3	-0.007	-0.418	-0.324	0.135	0.219	0.500	0.110
1	4	-0.007	-0.143	0.052	-0.007	0.219	0.500	0.110
1	5	-0.007	0.030	-0.420	-0.013	0.219	0.500	0.110
1	6	-0.007	-0.038	0.098	-0.004	0.219	0.500	0.110
1	7	-0.007	-0.405	0.165	-0.067	0.219	0.500	0.110
1	8	-0.007	-0.402	-0.430	0.173	0.219	0.500	0.110
1	9	-0.007	0.251	0.634	0.159	0.219	0.500	0.110
1	10	-0.007	-0.410	0.250	-0.102	0.219	0.500	0.110
1	11	-0.007	-0.203	-0.286	0.058	0.219	0.500	0.110
1	12	-0.007	-0.159	0.253	-0.040	0.219	0.500	0.110
1	13	-0.007	0.365	1.295	0.472	0.219	0.500	0.110
2	1	0.020	-0.168	1.426	-0.240	-0.056	0.500	-0.028
2	2	0.020	0.157	0.497	0.078	-0.056	0.500	-0.028
2	3	0.020	-0.308	-0.324	0.100	-0.056	0.500	-0.028

Tabel 3.8 Perhitungan δ_h (lanjutan)

h	o	z_h	$w_{h,o}$	δy_o	$\delta y_o \cdot w_{h,o}$	δ_{in_h}	$f'(z_h)$	δ_h
2	4	0.020	-0.350	0.052	-0.018	-0.056	0.500	-0.028
2	5	0.020	-0.410	-0.420	0.172	-0.056	0.500	-0.028
2	6	0.020	-0.008	0.098	-0.001	-0.056	0.500	-0.028
2	7	0.020	0.090	0.165	0.015	-0.056	0.500	-0.028
2	8	0.020	-0.056	-0.430	0.024	-0.056	0.500	-0.028
2	9	0.020	-0.227	0.634	-0.144	-0.056	0.500	-0.028
2	10	0.020	-0.469	0.250	-0.117	-0.056	0.500	-0.028
2	11	0.020	-0.265	-0.286	0.076	-0.056	0.500	-0.028
2	12	0.020	-0.455	0.253	-0.115	-0.056	0.500	-0.028
2	13	0.020	0.089	1.295	0.115	-0.056	0.500	-0.028
3	1	-0.002	-0.372	1.426	-0.530	-0.252	0.500	-0.126
3	2	-0.002	0.044	0.497	0.022	-0.252	0.500	-0.126
3	3	-0.002	0.179	-0.324	-0.058	-0.252	0.500	-0.126
3	4	-0.002	0.204	0.052	0.011	-0.252	0.500	-0.126
3	5	-0.002	0.258	-0.420	-0.108	-0.252	0.500	-0.126
3	6	-0.002	-0.292	0.098	-0.029	-0.252	0.500	-0.126
3	7	-0.002	-0.330	0.165	-0.055	-0.252	0.500	-0.126
3	8	-0.002	-0.227	-0.430	0.098	-0.252	0.500	-0.126
3	9	-0.002	0.174	0.634	0.110	-0.252	0.500	-0.126
3	10	-0.002	-0.177	0.250	-0.044	-0.252	0.500	-0.126
3	11	-0.002	-0.020	-0.286	0.006	-0.252	0.500	-0.126
3	12	-0.002	-0.018	0.253	-0.004	-0.252	0.500	-0.126
3	13	-0.002	0.255	1.295	0.330	-0.252	0.500	-0.126

Setelah nilai masukan backpropagation untuk node hidden δ_h didapatkan, perhitungan dilanjutkan dengan mencari nilai perubahan bobot input hidden $\Delta v_{i,h}$ dengan rumus $\Delta v_{i,h} = \alpha \delta_h x_i$, dimana konstanta belajar α bernilai 0.0001, dan hasil perhitungan dapat dilihat pada tabel 3.9

Tabel 3.9 Perhitungan perubahan bobot input hidden

i	x_i	h	a	δ_h	$\Delta v_{i,h}$
1	1	0	1	0.0001	0.1096629
1	2	0	1	0.0001	0.1096629
1	3	1	1	0.0001	1.09663E-05
1	4	0	1	0.0001	0.1096629
1	5	0	1	0.0001	0.1096629
1	6	0	1	0.0001	0.1096629
1	7	0	1	0.0001	0.1096629
1	8	0	1	0.0001	0.1096629
1	9	1	1	0.0001	1.09663E-05

Tabel 3.9 Perhitungan perubahan bobot input hidden (lanjutan)

i		x_i	h	a	δ_h	$\Delta v_{i,h}$
1	10	0	1	0.0001	0.1096629	0
1	11	0	1	0.0001	0.1096629	0
1	12	1	1	0.0001	0.1096629	1.09663E-05
1	13	0	1	0.0001	0.1096629	0
2	1	0	1	0.0001	0.1096629	0
2	2	0	1	0.0001	0.1096629	0
2	3	1	1	0.0001	0.1096629	1.09663E-05
2	4	0	1	0.0001	0.1096629	0
2	5	0	1	0.0001	0.1096629	0
2	6	0	1	0.0001	0.1096629	0
2	7	0	1	0.0001	0.1096629	0
2	8	0	1	0.0001	0.1096629	0
2	9	0	1	0.0001	0.1096629	0
2	10	0	1	0.0001	0.1096629	0
2	11	1	1	0.0001	0.1096629	1.09663E-05
2	12	0	1	0.0001	0.1096629	0
2	13	1	1	0.0001	0.1096629	1.09663E-05
3	1	0	1	0.0001	0.1096629	0
3	2	1	1	0.0001	0.1096629	1.09663E-05
3	3	0	1	0.0001	0.1096629	0
3	4	1	1	0.0001	0.1096629	1.09663E-05
3	5	1	1	0.0001	0.1096629	1.09663E-05
3	6	0	1	0.0001	0.1096629	0
3	7	1	1	0.0001	0.1096629	1.09663E-05
3	8	0	1	0.0001	0.1096629	0
3	9	1	1	0.0001	0.1096629	1.09663E-05
3	10	0	1	0.0001	0.1096629	0
3	11	0	1	0.0001	0.1096629	0
3	12	0	1	0.0001	0.1096629	0
3	13	0	1	0.0001	0.1096629	0
1	1	0	2	0.0001	-0.027921224	0
1	2	0	2	0.0001	-0.027921224	0
1	3	1	2	0.0001	-0.027921224	-2.79212E-06
1	4	0	2	0.0001	-0.027921224	0
1	5	0	2	0.0001	-0.027921224	0
1	6	0	2	0.0001	-0.027921224	0
1	7	0	2	0.0001	-0.027921224	0
1	8	0	2	0.0001	-0.027921224	0
1	9	1	2	0.0001	-0.027921224	-2.79212E-06
1	10	0	2	0.0001	-0.027921224	0
1	11	0	2	0.0001	-0.027921224	0
1	12	1	2	0.0001	-0.027921224	-2.79212E-06
1	13	0	2	0.0001	-0.027921224	0
2	1	0	2	0.0001	-0.027921224	0
2	2	0	2	0.0001	-0.027921224	0
2	3	1	2	0.0001	-0.027921224	-2.79212E-06

Tabel 3.9 Perhitungan perubahan bobot input hidden (lanjutan)

i		x_i	h	a	δ_h	$\Delta v_{i,h}$
2	4	0	2	0.0001	-0.027921224	0
2	5	0	2	0.0001	-0.027921224	0
2	6	0	2	0.0001	-0.027921224	0
2	7	0	2	0.0001	-0.027921224	0
2	8	0	2	0.0001	-0.027921224	0
2	9	0	2	0.0001	-0.027921224	0
2	10	0	2	0.0001	-0.027921224	0
2	11	1	2	0.0001	-0.027921224	-2.79212E-06
2	12	0	2	0.0001	-0.027921224	0
2	13	1	2	0.0001	-0.027921224	-2.79212E-06
3	1	0	2	0.0001	-0.027921224	0
3	2	1	2	0.0001	-0.027921224	-2.79212E-06
3	3	0	2	0.0001	-0.027921224	0
3	4	1	2	0.0001	-0.027921224	-2.79212E-06
3	5	1	2	0.0001	-0.027921224	-2.79212E-06
3	6	0	2	0.0001	-0.027921224	0
3	7	1	2	0.0001	-0.027921224	-2.79212E-06
3	8	0	2	0.0001	-0.027921224	0
3	9	1	2	0.0001	-0.027921224	-2.79212E-06
3	10	0	2	0.0001	-0.027921224	0
3	11	0	2	0.0001	-0.027921224	0
3	12	0	2	0.0001	-0.027921224	0
3	13	0	2	0.0001	-0.027921224	0
1	1	0	3	0.0001	-0.12606122	0
1	2	0	3	0.0001	-0.12606122	0
1	3	1	3	0.0001	-0.12606122	-1.26061E-05
1	4	0	3	0.0001	-0.12606122	0
1	5	0	3	0.0001	-0.12606122	0
1	6	0	3	0.0001	-0.12606122	0
1	7	0	3	0.0001	-0.12606122	0
1	8	0	3	0.0001	-0.12606122	0
1	9	1	3	0.0001	-0.12606122	-1.26061E-05
1	10	0	3	0.0001	-0.12606122	0
1	11	0	3	0.0001	-0.12606122	0
1	12	1	3	0.0001	-0.12606122	-1.26061E-05
1	13	0	3	0.0001	-0.12606122	0
2	1	0	3	0.0001	-0.12606122	0
2	2	0	3	0.0001	-0.12606122	0
2	3	1	3	0.0001	-0.12606122	-1.26061E-05
2	4	0	3	0.0001	-0.12606122	0
2	5	0	3	0.0001	-0.12606122	0
2	6	0	3	0.0001	-0.12606122	0
2	7	0	3	0.0001	-0.12606122	0
2	8	0	3	0.0001	-0.12606122	0
2	9	0	3	0.0001	-0.12606122	0
2	10	0	3	0.0001	-0.12606122	0

Tabel 3.9 Perhitungan perubahan bobot input hidden (lanjutan)

i	x_i	h	a	δ_h	$\Delta v_{i,h}$
2	11	1	3	0.0001	-0.12606122
2	12	0	3	0.0001	-0.12606122
2	13	1	3	0.0001	-0.12606122
3	1	0	3	0.0001	-0.12606122
3	2	1	3	0.0001	-0.12606122
3	3	0	3	0.0001	-0.12606122
3	4	1	3	0.0001	-0.12606122
3	5	1	3	0.0001	-0.12606122
3	6	0	3	0.0001	-0.12606122
3	7	1	3	0.0001	-0.12606122
3	8	0	3	0.0001	-0.12606122
3	9	1	3	0.0001	-0.12606122
3	10	0	3	0.0001	-0.12606122
3	11	0	3	0.0001	-0.12606122
3	12	0	3	0.0001	-0.12606122
3	13	0	3	0.0001	-0.12606122

Setelah perubahan bobot input hidden Δv dan perubahan bobot hidden output Δw didapatkan, proses perhitungan dilanjutkan dengan mengubah nilai bobot lama sebesar perubahan bobot hasil perhitungan, dengan rumus:

$w_{h,o}$ (baru) = $w_{h,o}$ (lama) + $\Delta w_{h,o}$ untuk bobot hidden output, dengan hasil perhitungan sebagaimana dapat dilihat pada tabel 3.10, dan $v_{i,h}$ (baru) = $v_{i,h}$ (lama) + $\Delta v_{i,h}$ untuk bobot input hidden, dengan hasil perhitungan sebagaimana dapat dilihat pada tabel 3.11. Nilai bobot baru akan digunakan untuk melakukan perhitungan pada iterasi berikutnya.

Tabel 3.10 Perhitungan update nilai bobot hidden output

h	o	$w_{h,o}$ lama	$\Delta w_{h,o}$	$w_{h,o}$ baru
0	1	-0.426046669	0.000142609	-0.425904061
1	1	-0.394547343	-9.63084E-07	-0.394548306
2	1	-0.168305576	2.85956E-06	-0.168302716
3	1	-0.371750116	-2.57259E-07	-0.371750374
0	2	-0.49975878	4.96937E-05	-0.499709086
1	2	3.68E-02	-3.35599E-07	0.036793731
2	2	0.157054961	9.96448E-07	0.157055957
3	2	4.40E-02	-8.9645E-08	0.044013887

Tabel 3.10 Perhitungan update nilai bobot hidden output (lanjutan)

h	o	$w_{h,o}$ lama	$\Delta w_{h,o}$	$w_{h,o}$ baru
0	3	0.32741183	-3.23735E-05	0.327379457
1	3	-0.418106437	2.18629E-07	-0.418106218
2	3	-0.308077514	-6.49146E-07	-0.308078163
3	3	0.178913355	5.84002E-08	0.178913413
0	4	-4.58E-02	5.22138E-06	-0.045786941
1	4	-0.14297688	-3.52617E-08	-0.142976915
2	4	-0.350018919	1.04698E-07	-0.350018814
3	4	0.204395771	-9.4191E-09	0.204395762
0	5	0.42878598	-4.19889E-05	0.428743991
1	5	3.02E-02	2.83565E-07	0.030212805
2	5	-0.410358608	-8.41951E-07	-0.41035945
3	5	0.257729292	7.57457E-08	0.257729368
0	6	-9.82E-02	9.75301E-06	-0.098148189
1	6	-3.81E-02	-6.58654E-08	-0.0381257
2	6	-7.83E-03	1.95565E-07	-0.007834179
3	6	-0.292372704	-1.75939E-08	-0.292372721
0	7	-0.170264065	1.65136E-05	-0.170247552
1	7	-0.404570937	-1.11522E-07	-0.404571049
2	7	8.98E-02	3.31127E-07	0.089792881
3	7	-0.330126524	-2.97897E-08	-0.330126554
0	8	0.427616537	-4.29616E-05	0.427573575
1	8	-0.402070165	2.90134E-07	-0.402069875
2	8	-5.61E-02	-8.61456E-07	-0.056138364
3	8	-0.227053165	7.75005E-08	-0.227053088
0	9	0.372546852	6.34012E-05	0.372610253
1	9	0.250687718	-4.2817E-07	0.25068729
2	9	-0.227057636	1.27131E-06	-0.227056364
3	9	0.173646688	-1.14373E-07	0.173646574
0	10	-0.243371069	2.49687E-05	-0.243346101
1	10	-0.41010344	-1.68622E-07	-0.410103609
2	10	-0.469049394	5.00667E-07	-0.469048893
3	10	-0.177282333	-4.50422E-08	-0.177282378
0	11	0.290128887	-2.86225E-05	0.290100264
1	11	-0.202741981	1.93297E-07	-0.202741787
2	11	-2.65E-01	-5.73932E-07	-0.264718332
3	11	-1.95E-02	5.16336E-08	-0.019525238
0	12	-0.245398104	2.53415E-05	-0.245372763
1	12	-0.15939343	-1.7114E-07	-0.159393601
2	12	-0.455066144	5.08142E-07	-0.455065636
3	12	-1.76E-02	-4.57147E-08	-0.017571972
0	13	-0.293982804	0.000129513	-0.293853291
1	13	0.364534497	-8.74643E-07	0.364533623
2	13	8.86E-02	2.59696E-06	0.088631187
3	13	0.254908323	-2.33634E-07	0.25490809

Tabel 3.11 Perhitungan update nilai bobot input hidden

i	h	$v_{i,h}$ lama	$\Delta v_{i,h}$	$v_{i,h}$ baru
1	1	5.31E-03	0	5.31E-03
1	2	-5.44E-03	0	-5.44E-03
1	3	-1.26E-02	1.09663E-05	-1.25E-02
1	4	5.40E-03	0	5.40E-03
1	5	9.37E-03	0	9.37E-03
1	6	1.19E-02	0	1.19E-02
1	7	1.16E-02	0	1.16E-02
1	8	6.90E-03	0	6.90E-03
1	9	-8.09E-04	1.09663E-05	-7.98E-04
1	10	3.82E-03	0	3.82E-03
1	11	8.52E-03	0	8.52E-03
1	12	1.26E-02	1.09663E-05	1.26E-02
1	13	5.04E-03	0	5.04E-03
2	1	8.75E-04	0	8.75E-04
2	2	4.55E-03	0	4.55E-03
2	3	-1.03E-02	1.09663E-05	-1.03E-02
2	4	-5.57E-03	0	-5.57E-03
2	5	-3.05E-03	0	-3.05E-03
2	6	1.24E-02	0	1.24E-02
2	7	-8.77E-03	0	-8.77E-03
2	8	-2.32E-03	0	-2.32E-03
2	9	-4.49E-03	0	-4.49E-03
2	10	-8.11E-03	0	-8.11E-03
2	11	-1.09E-03	1.09663E-05	-1.08E-03
2	12	7.37E-03	0	7.37E-03
2	13	1.08E-02	1.09663E-05	1.08E-02
3	1	-1.85E-03	0	-1.85E-03
3	2	5.03E-03	1.09663E-05	5.04E-03
3	3	-1.23E-02	0	-1.23E-02
3	4	-1.80E-03	1.09663E-05	-1.79E-03
3	5	3.55E-04	1.09663E-05	3.66E-04
3	6	-2.46E-03	0	-2.46E-03
3	7	-6.62E-03	1.09663E-05	-6.61E-03
3	8	-2.83E-03	0	-2.83E-03
3	9	-8.90E-03	1.09663E-05	-8.89E-03
3	10	3.33E-03	0	3.33E-03
3	11	1.13E-02	0	1.13E-02
3	12	-2.83E-03	0	-2.83E-03
3	13	-1.04E-03	0	-1.04E-03
1	1	8.64E-04	0	8.64E-04
1	2	-5.12E-03	0	-5.12E-03
1	3	6.74E-03	-2.79212E-06	6.73E-03
1	4	-1.17E-02	0	-1.17E-02
1	5	7.51E-03	0	7.51E-03
1	6	9.60E-03	0	9.60E-03
1	7	-3.51E-03	0	-3.51E-03

Tabel 3.11 Perhitungan update nilai bobot input hidden (lanjutan)

i		h	$v_{i,h}$ lama	$\Delta v_{i,h}$	$v_{i,h}$ baru
1	8	2	-0.011537333	0	-1.15E-02
1	9	2	-0.005215355	-2.79212E-06	-5.22E-03
1	10	2	-0.006103532	0	-6.10E-03
1	11	2	0.008387639	0	8.39E-03
1	12	2	0.010619216	-2.79212E-06	1.06E-02
1	13	2	1.240316532	0	1.24E+00
2	1	2	-0.010171311	0	-1.02E-02
2	2	2	-0.012514091	0	-1.25E-02
2	3	2	-0.010257797	-2.79212E-06	-1.03E-02
2	4	2	-0.011740313	0	-1.17E-02
2	5	2	-0.005142873	0	-5.14E-03
2	6	2	-0.002548463	0	-2.55E-03
2	7	2	-0.008712607	0	-8.71E-03
2	8	2	-0.002254084	0	-2.25E-03
2	9	2	0.00344131	0	3.44E-03
2	10	2	0.002153976	0	2.15E-03
2	11	2	0.010483959	-2.79212E-06	1.05E-02
2	12	2	-0.003129128	0	-3.13E-03
2	13	2	0.003404192	-2.79212E-06	3.40E-03
3	1	2	-0.010388257	0	-1.04E-02
3	2	2	0.01069036	-2.79212E-06	1.07E-02
3	3	2	0.001120427	0	1.12E-03
3	4	2	0.004598125	-2.79212E-06	4.60E-03
3	5	2	-0.000956587	-2.79212E-06	-9.59E-04
3	6	2	-0.005950079	0	-5.95E-03
3	7	2	0.012379256	-2.79212E-06	1.24E-02
3	8	2	-0.003488485	0	-3.49E-03
3	9	2	-0.000659969	-2.79212E-06	-6.63E-04
3	10	2	0.001087084	0	1.09E-03
3	11	2	0.003992227	0	3.99E-03
3	12	2	-0.010145325	0	-1.01E-02
3	13	2	0.006555238	0	6.56E-03
1	1	3	0.002054741	0	2.05E-03
1	2	3	0.007099216	0	7.10E-03
1	3	3	0.008126345	-1.26061E-05	8.11E-03
1	4	3	-0.002221374	0	-2.22E-03
1	5	3	-0.003267794	0	-3.27E-03
1	6	3	-0.01146673	0	-1.15E-02
1	7	3	0.000642594	0	6.43E-04
1	8	3	0.002389098	0	2.39E-03
1	9	3	0.003170452	-1.26061E-05	3.16E-03
1	10	3	-0.005701747	0	-5.70E-03
1	11	3	0.00230395	0	2.30E-03
1	12	3	-0.007057715	-1.26061E-05	-7.07E-03
1	13	3	-0.006616751	0	-6.62E-03
2	1	3	0.012904749	0	1.29E-02

Tabel 3.11 Perhitungan update nilai bobot input hidden (lanjutan)

i		h	$v_{i,h}$ lama	$\Delta v_{i,h}$	$v_{i,h}$ baru
2	2	3	0.001942731	0	1.94E-03
2	3	3	0.007723099	-1.26061E-05	7.71E-03
2	4	3	-0.005277179	0	-5.28E-03
2	5	3	0.011590966	0	1.16E-02
2	6	3	-0.005729191	0	-5.73E-03
2	7	3	0.003787775	0	3.79E-03
2	8	3	0.005496903	0	5.50E-03
2	9	3	-0.007556548	0	-7.56E-03
2	10	3	-0.01083423	0	-1.08E-02
2	11	3	-0.006166185	-1.26061E-05	-6.18E-03
2	12	3	-0.005435003	0	-5.44E-03
2	13	3	0.003298239	-1.26061E-05	3.29E-03
3	1	3	0.001577261	0	1.58E-03
3	2	3	0.008651593	-1.26061E-05	8.64E-03
3	3	3	0.010753575	0	1.08E-02
3	4	3	6.34088E-05	-1.26061E-05	5.08E-05
3	5	3	-0.003786231	-1.26061E-05	-3.80E-03
3	6	3	-0.011483355	0	-1.15E-02
3	7	3	-0.011345816	-1.26061E-05	-1.14E-02
3	8	3	-0.000261117	0	-2.61E-04
3	9	3	-0.006272144	-1.26061E-05	-6.28E-03
3	10	3	-0.008881066	0	-8.88E-03
3	11	3	0.000157296	0	1.57E-04
3	12	3	0.007338368	0	7.34E-03
3	13	3	0.002483059	0	2.48E-03

3.2.2 Alur Proses

Terdapat 2 proses yang dilakukan oleh sistem penentuan hero yaitu pelatihan NN dan captain mode sebagai hasil dari pelatihan NN yang berupa rekomendasi ban dan pick hero untuk team 2.

A. Pelatihan NN

Didalam pelatihan NN ini dapat dijelaskan tahapan setiap proses sebagai berikut:

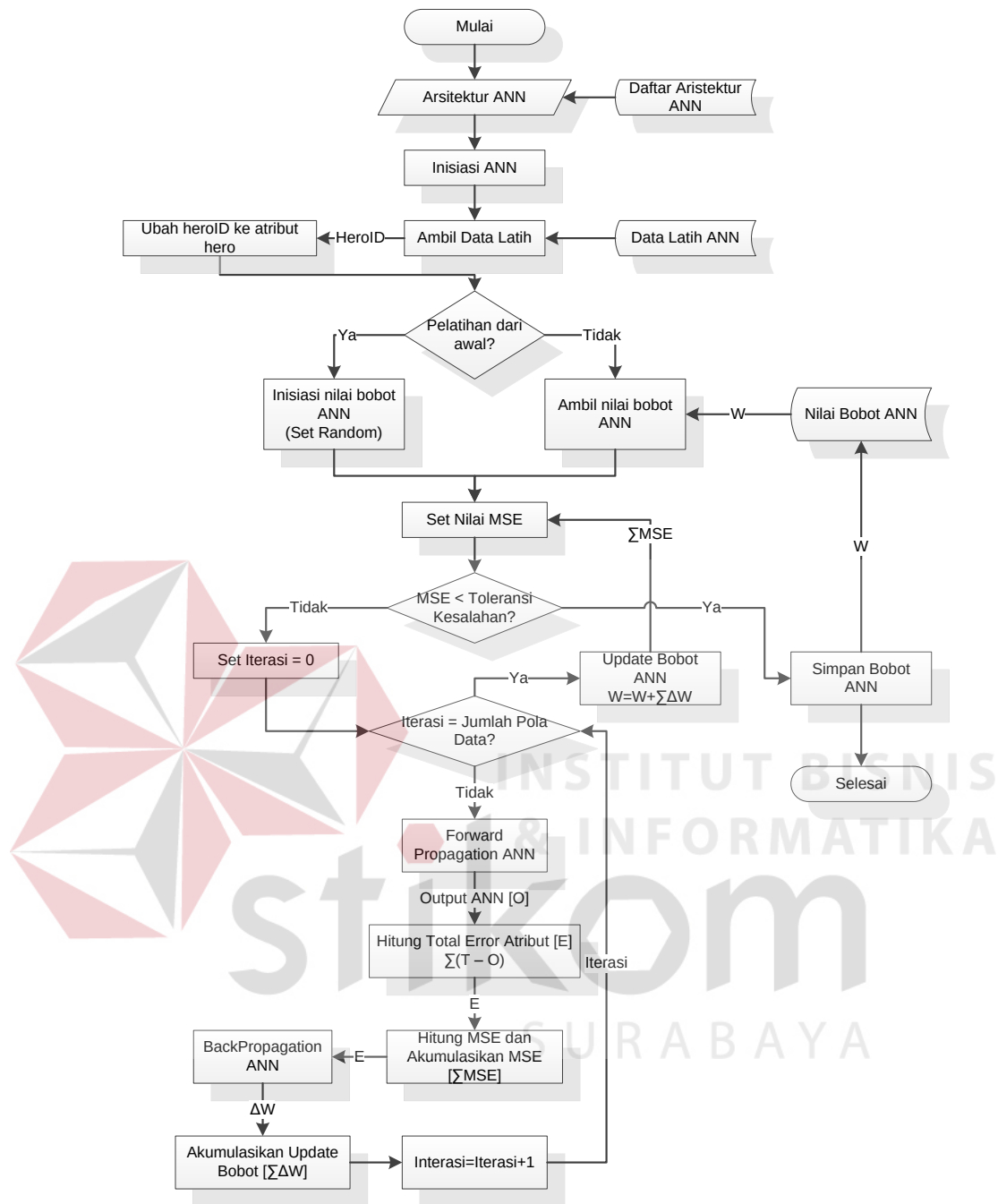
1. Memilih setting konfigurasi NN dalam pelatihan NN.
2. Proses inisiasi NN dimana dalam proses ini pembentukan grup input, node input, grup output, node output, node hidden, konstanta belajar, pola data yang akan digunakan, jumlah pola data yang akan dilatihkan, jumlah iterasi dan nilai

MSE dari pelatihan sebelumnya, dan nilai toleransi pelatihan. Dibentuk berdasarkan dari setting konfigurasi NN.

3. Pengambilan data latih yang didapatkan dari data latih didalam database.
4. Merubah HeroID yang didapatkan dari data latih menjadi atribut hero.
5. Pemilihan jenis pelatihan apakah pelatihan dimulai dari awal atau tidak.
Inisiasi nilai bobot yang penentuannya secara random dilakukan apabila pelatihan dimulai dari awal, sedangkan pengambilan nilai bobot NN yang diambil dari database dilakukan apabila pelatihannya melanjutkan dari pelatihan yang sudah dilakukan.
6. Penentuan nilai Mean Square Error (MSE) yang bernilai lebih besar daripada toleransi untuk pilihan pelatihan dari awal dan bernilai sama dengan nilai MSE pelatihan sebelumnya untuk melanjutkan pelatihan sebelumnya.
7. Pengecekan apakah nilai MSE lebih kecil dari toleransi, apabila tidak maka iterasi diberi nilai 0, sedangkan bila iterasi lebih besar sama dengan toleransi maka dilakukan proses penyimpanan bobot NN.
8. Setelah iterasi diberi nilai 0 maka proses selanjutnya adalah pengecekan apakah iterasi sama dengan jumlah pola data, apabila sama maka dilakukan update bobot NN dimana bobot didapatkan dari akumulasi bobot ditambahkan dengan nilai bobot yang baru. Bila jumlah iterasi tidak sama maka dilakukan *forward propagation NN*.
9. Berdasarkan output NN yang didapatkan dari *forward propagation NN* maka dilakukan perhitungan akumulasi error yang didapatkan dari pengurangan output dengan target.

1. Perhitungan MSE dan akumulasi MSE dilakukan setelah mendapatkan nilai error dari proses diatas.
2. Hasil dari perhitungan MSE dan akumulasi MSE yang berupa nilai error akan dilakukan proses back propagation yang menghasilkan akumulasi bobot.
3. Proses selanjutnya adalah mengakumulasikan update bobot dan dilakukan penambahan nilai iterasi untuk menandakan iterasi baru.
4. Setelah proses diatas selesai maka akan dilakukan kembali tahapan proses 8, dimana proses 9 sampai 13 akan berhenti apabila nilai MSE sama dengan 0 atau seminimal mungkin sesuai dengan keinginan pengguna.





Gambar 3.9 Proses Pelatihan NN

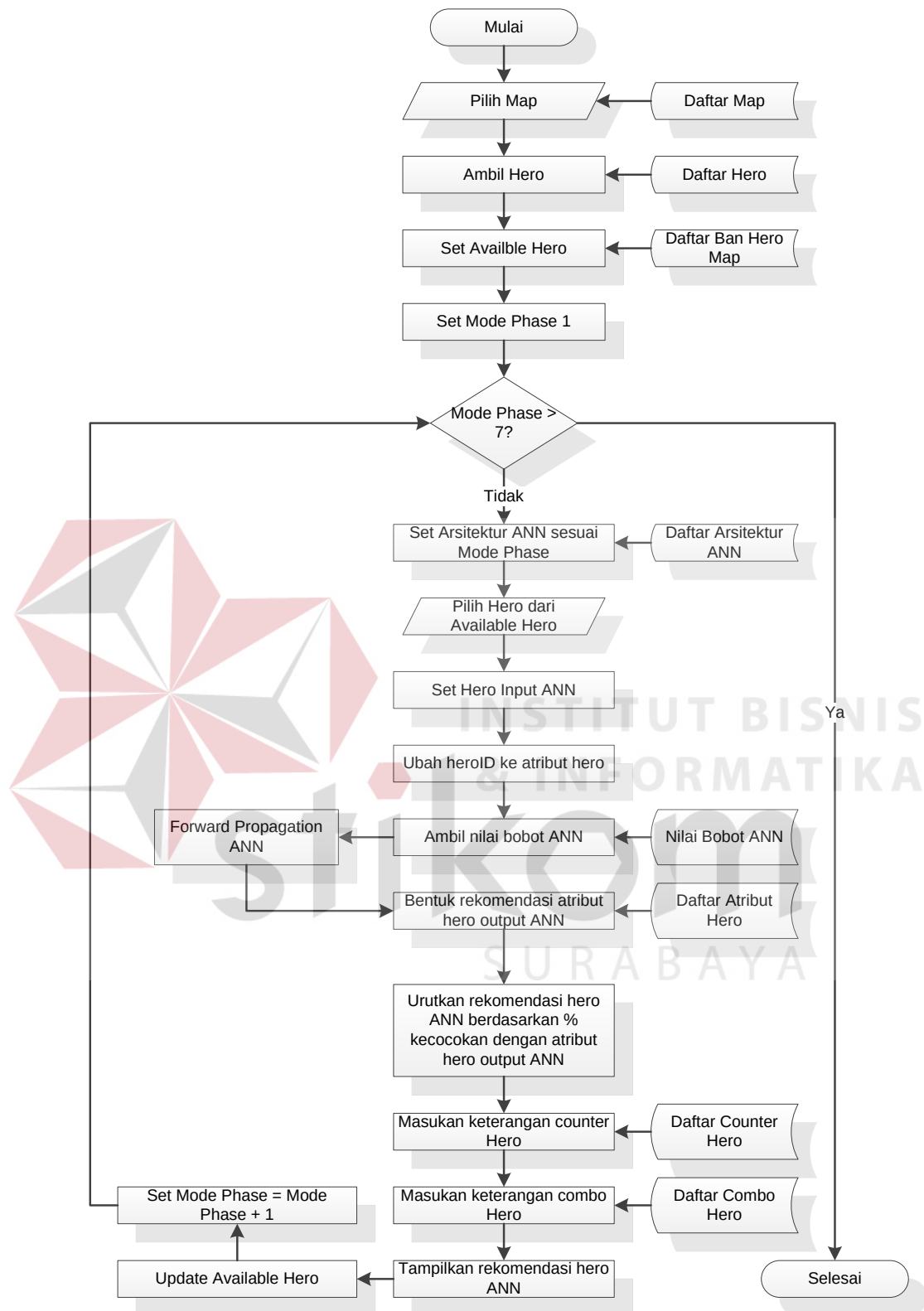
B. Captain Mode

Proses dalam captain mode ini dapat dijabarkan sebagai berikut:

1. Pemilihan map yang didapatkan dari daftar map.
2. Pemilihan hero berdasarkan daftar hero yang ada.
3. Melakukan setting available hero yang didapatkan dari ban phase dan secara default setiap versi map memiliki hero yang tidak dapat digunakan (ban).
4. Melakukan pemilihan mode phase pertama.
5. Dilakukan pengecekan apakah mode phase sudah ketujuh? Apabila sudah sesuai maka proses selesai.
6. Apabila proses diatas belum sesuai maka dilakukan proses setting arsitektur NN yang sesuai dengan mode phase dimana arsitektur ini didapat dari daftar arsitektur NN.
7. Proses selanjutnya adalah pemilihan hero berdasarkan available hero.
8. Dilakukan setting hero sebagai input kedalam NN.
9. Merubah HeroID menjadi atribut hero dilakukan setelah proses input hero di dalam NN.
10. Melakukan pengambilan nilai bobot dari nilai bobot NN yang didapatkan dari pelatihan dan dilakukan *forward propagation* berdasarkan dari atribut hero dan nilai bobot.
11. Bentuk rekomendasi atribut hero output NN yang nantinya akan dibandingkan berdasarkan atribut yang dimiliki setiap hero.
12. proses selanjutnya adalah pengurutan rekomendasi hero NN berdasarkan prosentase kecocokan dengan atribut hero output.
13. Memasukkan keterangan counter hero yang didapatkan dari daftar counter hero.
14. Memasukkan keterangan combo hero yang didapatkan dari daftar combo hero.

15. Setelah proses diatas selesai maka sistem akan menampilkan rekomendasi hero dan melakukan update available hero dan mode phase akan ditambahkan 1 untuk menandakan phase berikutnya.





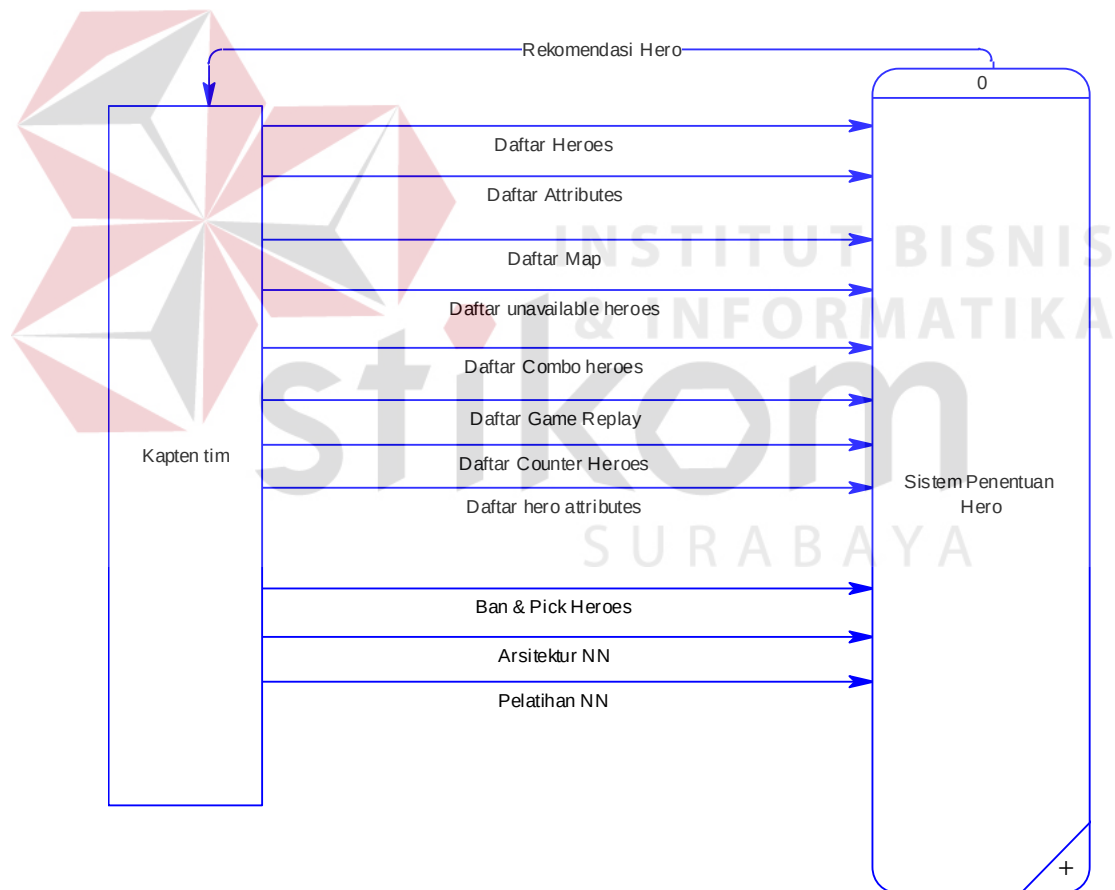
Gambar 3.10 Proses Captain Mode

3.2.3 Data Flow Diagram (DFD)

DFD merupakan representasi grafik dalam menggambarkan arus data sistem secara terstruktur dan jelas sehingga dapat menjadi sarana dokumentasi yang baik.

A. Context Diagram

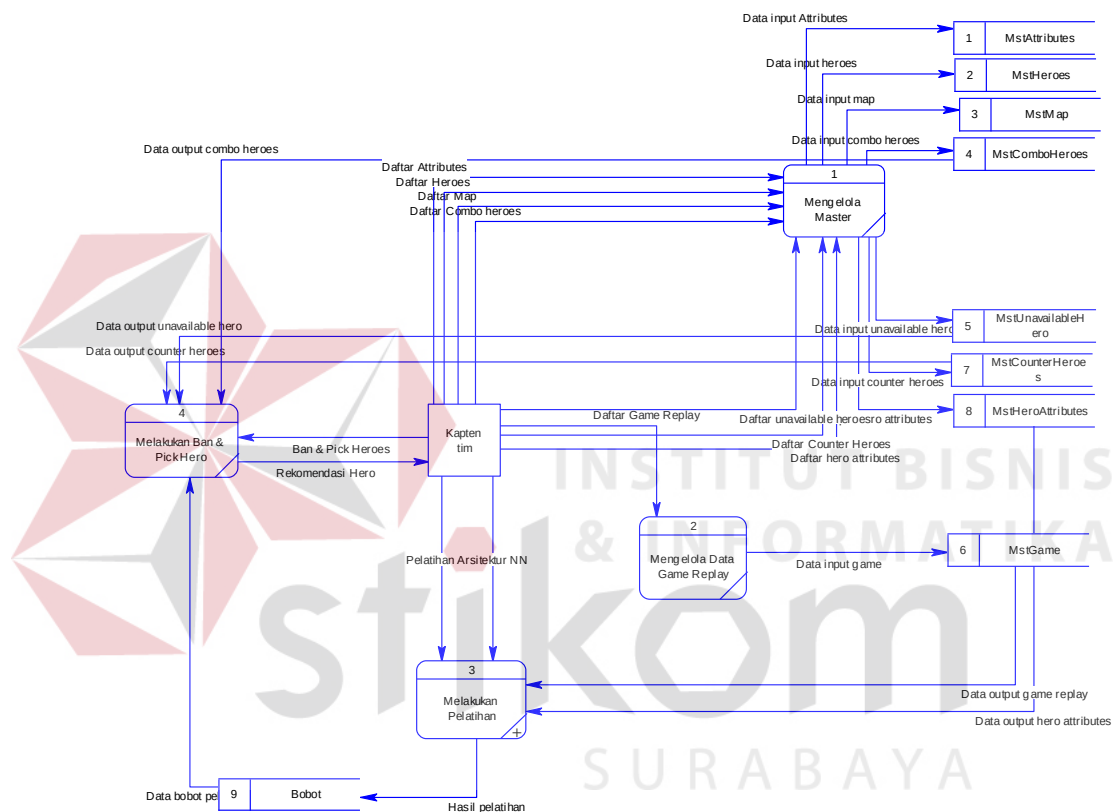
Context diagram ini menggambarkan rancangan global/ keseluruhan dari proses yang ada pada DFD. Gambar 3.9 berikut ini merupakan tampilan dari context diagram sistem yang dirancang.



Gambar 3.11 Context Diagram dari DFD

B. DFD Level 0

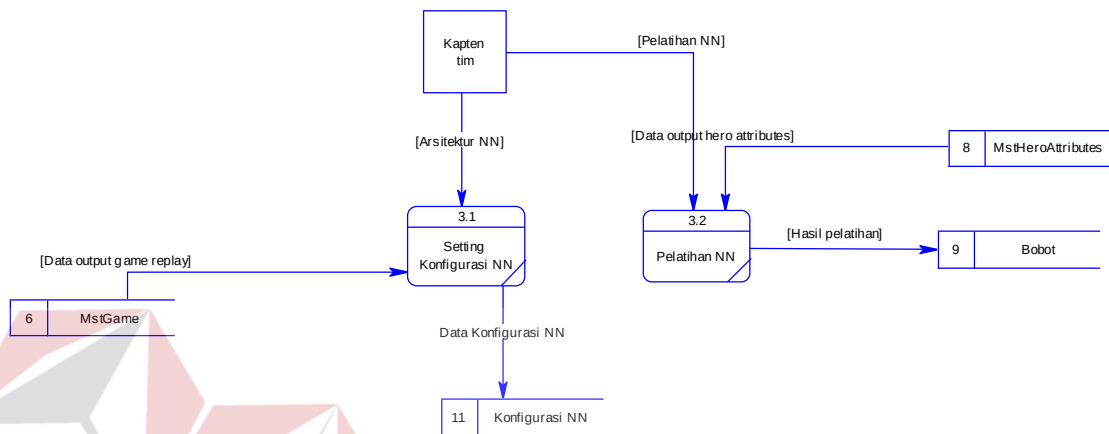
Dari *context diagram* yang ada, sistem yang terjadi dapat dipecah lagi menjadi beberapa proses, yaitu proses Mengelola Master, proses Mengelola Data Replay, proses Melakukan Pelatihan, dan proses Melakukan Ban & Pick Hero



Gambar 3.12 Level 0 dari Data Flow Diagram (DFD)

C. DFD Level 1 Melakukan Pelatihan

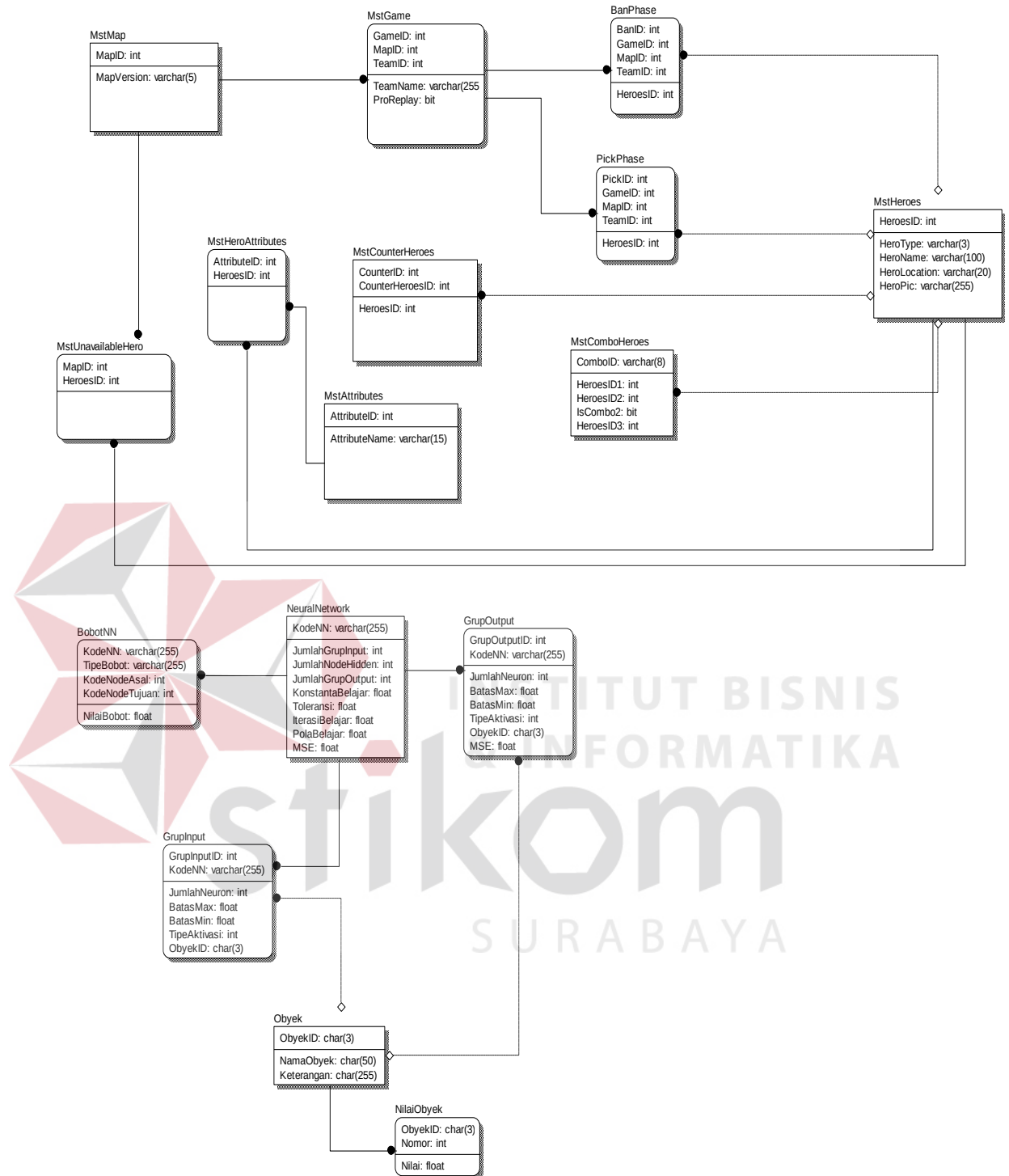
Dari DFD Level 1 Melakukan Pelatihan, proses yang terjadi dapat dipecah lagi menjadi beberapa subproses, yaitu subproses Setting Konfigurasi NN dan subproses Pelatihan NN.



Gambar 3.13 DFD Level 1 dari Melakukan Pelatihan

3.2.4 Entity Relationship Diagram (ERD)

ERD Merupakan suatu desain sistem yang digunakan untuk merepresentasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ERD juga menyediakan bentuk untuk menunjukkan struktur keseluruhan data dari pemakai. Dalam perancangan sistem ini telah dibuat ERD yang merupakan lanjutan dari pembuatan desain dengan menggunakan DFD.



Gambar 3.14 Entity Relationship Diagram (ERD)

3.3 Struktur Database

Struktur *database* menggambarkan data-data yang ada dalam *database* beserta tipe dan kegunaannya.

3.3.1 Tabel MstGame

Nama Tabel : MstGame

Primary Key : GameID

Foreign Key : MapID, TeamID

Fungsi : Menyimpan informasi data Master Game Replay

Tabel 3.12. Struktur Tabel MstGame

Field	Tipe	Ukuran	Keterangan
GameID	Int	-	
MapID	Int	-	
TeamID	Int	-	
TeamName	Varchar	255	
ProReplay	bit	-	

3.3.2 Tabel BanPhase

Nama Tabel : BanPhase

Primary Key : BanID

Foreign Key : MapID, TeamID, Game ID

Fungsi : Menyimpan informasi data Ban Phase

Tabel 3.13. Struktur Tabel BanPhase

Field	Tipe	Ukuran	Keterangan
BanID	Int	-	
MapID	Int	-	
GameID	Int	-	
TeamID	Int	-	
HeroesID	Int	-	

3.3.3 Tabel PickPhase

Nama Tabel : PickPhase

Primary Key : PickID

Foreign Key : MapID, TeamID, Game ID

Fungsi : Menyimpan informasi data PickPhase

Tabel 3.14. Struktur Tabel PickPhase

Field	Type	Ukuran	Keterangan
PickID	Int	-	
MapID	Int	-	
GameID	Int	-	
TeamID	Int	-	
HeroesID	Int	-	

3.3.4 Tabel MstHeroes

Nama Tabel : MstHeroes

Primary Key : HeroesID

Foreign Key : -

Fungsi : Menyimpan informasi data Heroes

Tabel 3.15. Struktur Tabel MstHeroes

Field	Type	Ukuran	Keterangan
HeroesID	Int	-	
HeroType	Varchar	3	
HeroName	Varchar	100	
HeroLocation	Varchar	20	
HeroPic	Varchar	255	

3.3.5 Tabel MstCounterHeroes

Nama Tabel : MstCounterHeroes

Primary Key : CounterID, CounterHeroesID

Foreign Key : HeroesID

Fungsi : Menyimpan informasi data Counter Heroes

Tabel 3.16. Struktur Tabel MstCounterHeroes

Field	Tipe	Ukuran	Keterangan
CounterID	Int	-	
CounterHeroesID	Int	-	
HeroesID	Int	-	

3.3.6 Tabel MstComboHeroes

Nama Tabel : MstComboHeroes

Primary Key : ComboID

Foreign Key : -

Fungsi : Menyimpan informasi data ComboHeroes

Tabel 3.17. Struktur Tabel MstComboHeroes

Field	Tipe	Ukuran	Keterangan
ComboID	Varchar	8	
HeroesID1	Int	-	
HeroesID2	Int	-	
isCombo2	bit	-	
HeroesID3	Int	-	

3.3.7 Tabel MstAttributes

Nama Tabel : MstAttributes

Primary Key : AttributeID

Foreign Key : -

Fungsi : Menyimpan informasi data Attribute

Tabel 3.18. Struktur Tabel MstAttributes

Field	Type	Ukuran	Keterangan
AttributeID	Int	-	
AttributeName	Varchar	15	

3.3.8 Tabel MstMap

Nama Tabel : MstMap

Primary Key : MapID

Foreign Key : -

Fungsi : Menyimpan data MstMap

Tabel 3.19. Struktur Tabel MstMap

Field	Type	Ukuran	Keterangan
MapID	Int	-	
MapVersion	Varchar	5	

3.3.9 Tabel MstUnavailableHero

Nama Tabel : MstUnavailableHero

Primary Key : MapID, HeroesID

Foreign Key : -

Fungsi : Menyimpan informasi data MstUnavailableHero

Tabel 3.20. Struktur Tabel MstUnavailableHero

Field	Type	Ukuran	Keterangan
MapID	Int	-	
HeroesID	Int	-	

3.3.10 Tabel NeuralNetwork

Nama Tabel : NeuralNetwork

Primary Key : KodeNN

Foreign Key : -

Fungsi : Menyimpan informasi data Neural Network

Tabel 3.21. Struktur Tabel NeuralNetwork

Field	Tipe	Ukuran	Keterangan
KodeNN	Varchar	255	
JumlahGroupInput	Int	-	
JumlahNodeHidden	Int	-	
JumlahGroupOutput	Int	-	
KonstantaBelajar	Float	-	
Toleransi	Float	-	
IterasiBelajar	Float	-	
PolaBelajar	Float	-	
MSE	Float	-	

3.3.11 Tabel GroupOutput

Nama Tabel : GroupOutput

Primary Key : GroupOutputID

Foreign Key : KodeNN

Fungsi : Menyimpan data informasi Group Output

Tabel 3.22. Struktur Tabel GroupOutput

Field	Tipe	Ukuran	Keterangan
GroupOutputID	Int	-	
KodeNN	Varchar	255	
JumlahNeuron	Int	-	
BatasMax	Float	-	
BatasMin	Float	-	

TipeAktivasi	Int	-	
ObyekID	Char	3	
MSE	Float	-	

3.3.12 Tabel GroupInput

Nama Tabel : GroupInput

Primary Key : GroupInputID

Foreign Key : KodeNN

Fungsi : Menyimpan data informasi Group Input

Tabel 3.23. Struktur Tabel GroupInput

Field	Type	Ukuran	Keterangan
GroupInputID	Int	-	
KodeNN	Varchar	255	
JumlahNeuron	Int	-	
BatasMax	Float	-	
BatasMin	Float	-	
TipeAktivasi	Int	-	
ObyekID	Char	3	

3.3.13 Tabel BobotNN

Nama Tabel : BobotNN

Primary Key : TipeBobot, KodeNodeAsal, KodeNodeTujuan

Foreign Key : KodeNN

Fungsi : Menyimpan data informasi Bobot NN

Tabel 3.24. Struktur Tabel BobotNN

Field	Type	Ukuran	Keterangan
KodeNN	Varchar	255	
TipeBobot	Varchar	255	
KodeNodeAsal	Int	-	
KodeNodeTujuan	Int	-	
NilaiBobot	Float	-	

3.3.14 Tabel Obyek

Nama Tabel : Obyek

Primary Key : ObyekID

Foreign Key : -

Fungsi : Menyimpan data Obyek

Tabel 3.25. Struktur Tabel Obyek

Field	Tipe	Ukuran	Keterangan
ObyekID	Char	3	
NamaObyek	Char	50	
Keterangan	Char	255	

3.3.15 Tabel NilaiObyek

Nama Tabel : NilaiObyek

Primary Key : ObyekID

Foreign Key : -

Fungsi : Menyimpan data informasi NilaiObyek

Tabel 3.26. Struktur Tabel NilaiObyek

Field	Tipe	Ukuran	Keterangan
ObyekID	Char	3	
Nomor	Int	-	
Nilai	Float	-	

3.4 Desain User Interface

Pada tahap ini akan dibagi menjadi tiga aspek desain yang dibutuhkan oleh aplikasi agar dapat berjalan dengan baik yaitu Master Data, Konfigurasi dan Pelatihan NN, dan DotA CM

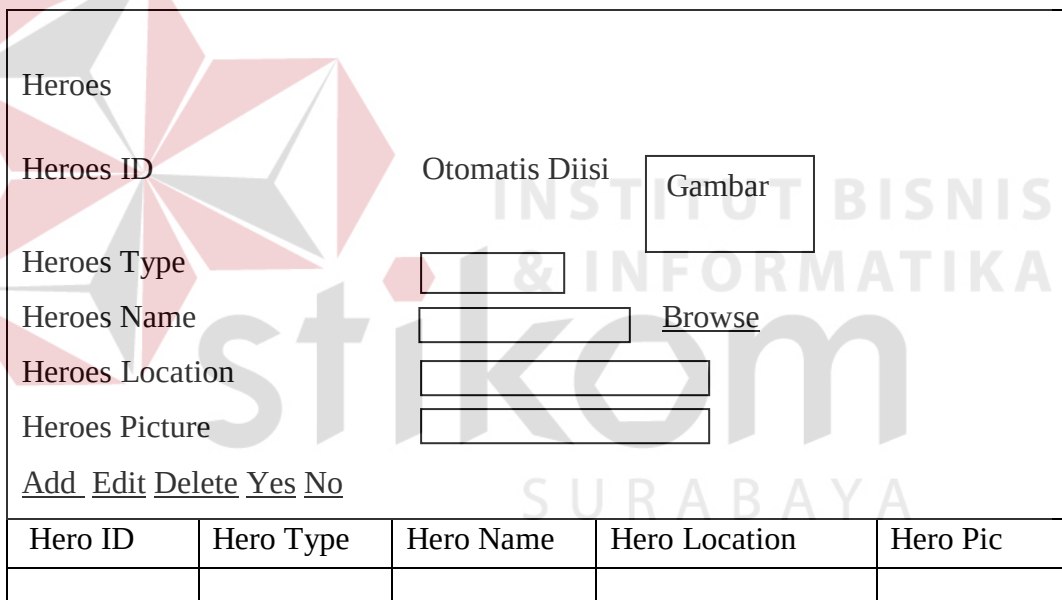
3.4.1 Master Data

A. Master Heroes

Form ini berfungsi untuk mengisi, mengubah, menghapus informasi data hero, counter, combo dan atribut setiap hero pada DotA. Form ini memiliki 5 buah tab yaitu heroes, counter heroes, 2 combo, 3 combo, dan hero attributes.

A.1 Tab Heroes

Tab ini digunakan untuk mengisi informasi, mengubah, menghapus, memberi gambar pada hero di DotA Contoh form Master Heroes dapat dilihat pada Gambar 3.15.



Hero ID	Hero Type	Hero Name	Hero Location	Hero Pic

Gambar 3.15. Contoh Tampilan Master Heroes

Fungsi-fungsi obyek dalam desain form Master Heroes sebagai berikut:

Tabel 3.27. Fungsi obyek form Master Heroes

Nama Obyek	Tipe Obyek	Fungsi
List Heroes	Gridview	Digunakan untuk Memilih hero yang akan di ubah atau dihapus

Add Edit Delete Yes No

Gambar 3.16 Contoh Tampilan form Counter Heroes

Fungsi-fungsi obyek dalam desain form Counter Heroes sebagai berikut:

Tabel 3.28. Fungsi obyek form Counter Heroes

Nama Obyek	Tipe Obyek	Fungsi
List Heroes	<i>Gridview</i>	Digunakan untuk memilih hero yang akan di ubah atau dihapus
Browse	<i>Button</i>	Digunakan untuk mencari gambar wajah dan nama counter hero yang kemudian otomatis akan terisi pada gambar dan textbox
Add	<i>Button</i>	Digunakan untuk menambah data counter hero baru
Edit	<i>Button</i>	Digunakan untuk mengubah data counter hero
Delete	<i>Button</i>	Digunakan untuk menghapus data counter hero
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridView
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

A.3 Tab 2 Combo

Tab ini digunakan untuk mengisi, mengubah, dan menghapus macam-macam combo menggunakan 2 hero. Contoh form 2 combo heroes dapat dilihat pada Gambar 3.17.

2 Combo Heroes

Gambar

browse

browse

☐		<u>browse</u>	☐		<u>browse</u>
☐		<u>browse</u>	☐		<u>browse</u>
<u>Add</u> <u>Edit</u> <u>Delete</u> <u>Yes</u> <u>No</u>					

Gambar3.17 Contoh Tampilan form 2 Combo Heroes

Fungsi-fungsi obyek dalam desain form 2 Combo Heroes sebagai berikut:

Tabel 3.29. Fungsi obyek form 2 Combo Heroes

Nama Obyek	Tipe Obyek	Fungsi
List Heroes	<i>Gridview</i>	Digunakan untuk memilih hero yang akan di ubah atau dihapus combonya
Browse	<i>Button</i>	Digunakan untuk mencari gambar wajah dan nama combo hero yang kemudian otomatis akan terisi pada gambar dan textbox
Add	<i>Button</i>	Digunakan untuk menambah data combo hero baru
Edit	<i>Button</i>	Digunakan untuk mengubah data combo hero
Delete	<i>Button</i>	Digunakan untuk menghapus data combo hero

Tabel 3.29 Fungsi Objek form 2 Combo Heroes (Lanjutan)

Nama Obyek	Tipe Obyek	Fungsi
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridView
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

A.4 Tab 3 Combo

Tab ini digunakan untuk mengisi, mengubah, dan menghapus macam-macam combo menggunakan 3 hero. Contoh form 2 combo heroes dapat dilihat pada Gambar 3.18.

3 Combo Heroes

Gambar

☐ browse ☐ browse

☐ browse ☐ browse

☐ browse ☐ browse

Add Edit Delte Yes No

Gambar 3.18 Contoh Tampilan form 3 Combo Heroes

Fungsi-fungsi obyek dalam desain form 3 Combo Heroes sebagai berikut:

Tabel 3.30. Fungsi obyek form 3 Combo Heroes

Nama Obyek	Tipe Obyek	Fungsi
List Heroes	<i>Gridview</i>	Digunakan untuk memilih hero yang akan di ubah atau dihapus combonya

Browse	<i>Button</i>	Digunakan untuk mencari gambar wajah dan nama combo hero yang kemudian otomatis akan terisi pada gambar dan textbox
Add	<i>Button</i>	Digunakan untuk menambah data combo hero baru
Edit	<i>Button</i>	Digunakan untuk mengubah data combo hero
Delete	<i>Button</i>	Digunakan untuk menghapus data combo hero
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridView
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

A.5 Tab Hero Attributes

Tab ini digunakan untuk mengisi, mengubah, dan menghapus macam-macam atribut pada setiap hero. Contoh form hero attributes dapat dilihat pada Gambar 3.19.

Hero Attributes				Heroes List	
Gambar					
Heroes Task ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼ ▼					
<u>Add</u> <u>Edit</u> <u>Delete</u> <u>Yes</u> <u>No</u>				Heroes ID	Heroes Name

Gambar 3.19 Contoh Tampilan Input Data Attribute Hero

Fungsi-fungsi obyek dalam desain form Input Data Attribute Heroes sebagai berikut

Tabel 3.31. Fungsi obyek form Hero Attribute

Nama Obyek	Tipe Obyek	Fungsi
List Heroes	<i>Gridview</i>	Digunakan untuk memilih hero yang akan di ubah attributnya atau dihapus
-	<i>ComboBox</i>	Digunakan untuk mencari nama attribut yang sesuai dengan karakteristik hero
Add	<i>Button</i>	Digunakan untuk menambah data attribute hero baru
Edit	<i>Button</i>	Digunakan untuk mengubah data attribute pada hero
Delete	<i>Button</i>	Digunakan untuk menghapus data attribute pada hero
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridView
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

B. Master Map dan Attributes

Form ini berfungsi untuk mengisi, mengubah, menghapus informasi data map, unavailable hero pada map, dan atribut. Form ini memiliki 3 buah tab yaitu Maps, Unavailable heroes, dan Attributes.

B.1 Tab Maps

Tab ini digunakan untuk mengisi informasi, mengubah, menghapus Map version DotA. Contoh form Master Map dapat dilihat pada Gambar 3.20.

Map ID	Otomatis Diisi	Map ID	Map Version
Map Version			
Add	Edit Delete Yes No		

Gambar 3.20 Contoh Tampilan Master Map

Fungsi-fungsi obyek dalam desain form Map sebagai berikut:

Tabel 3.32. Fungsi obyek form Map

Nama Obyek	Tipe Obyek	Fungsi
Map List	<i>Gridview</i>	Digunakan untuk memilih map yang akan di ubah atau dihapus
Map ID	<i>Textbox</i>	Di <i>generate</i> otomatis oleh sistem. Menunjukan Kode hero untuk Map baru
Map Version	<i>Textbox</i>	Diisi dengan versi map
Add	<i>Button</i>	Digunakan untuk menambah data map baru
Edit	<i>Button</i>	Digunakan untuk mengubah data map
Delete	<i>Button</i>	Digunakan untuk menghapus data map
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridview
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

B.2 Tab Unavailable heroes

Tab ini digunakan untuk mengisi informasi, mengubah, menghapus data-data hero yang tidak bisa dimainkan pada map tertentu. Contoh form unavailable hero dapat dilihat pada Gambar 3.21.

Banned Heroes for Map					
☐		browse	☐		browse
☐		browse	☐		browse
☐		browse	☐		browse
Add Edit Delete Yes No					
Map ID		Map Version			

Gambar 3.21 Contoh Tampilan Input Data Unavailable Hero

Fungsi-fungsi obyek dalam desain form Unavailable hero sebagai berikut:

Tabel 3.33. Fungsi obyek form Unavailable hero

Nama Obyek	Tipe Obyek	Fungsi
Unavailable Heroes	<i>Gridview</i>	Digunakan untuk memilih map yang akan diisikan hero yang tidak boleh dimainkan pada versi tersebut
Browse	<i>Button</i>	Digunakan untuk mencari gambar wajah dan nama hero yang kemudian otomatis akan terisi pada gambar dan textbox
Add	<i>Button</i>	Digunakan untuk menambah data unavailable hero pada map tersebut
Edit	<i>Button</i>	Digunakan untuk mengubah data unavailable hero pada map tersebut
Delete	<i>Button</i>	Digunakan untuk menghapus data unavailable hero pada map tersebut
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridview
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

B.3 Tab Attributes

Tab ini digunakan untuk mengisi informasi, mengubah, menghapus macam-macam jenis attribute yang dimiliki oleh hero. Contoh form attributes dapat dilihat pada Gambar 3.22.

Attribute ID	Otomatis Diisi	Att ID	Attribute Name
Attribute Name			
Add Edit Delete Yes No			

Gambar 3.22 Contoh Tampilan form Master Attributes

Fungsi-fungsi obyek dalam desain form Attributes sebagai berikut:

Tabel 3.34. Fungsi obyek form Attributes

Nama Obyek	Tipe Obyek	Fungsi
------------	------------	--------

Attribute ID	<i>Textbox</i>	Di <i>generate</i> otomatis oleh sistem. Menunjukkan Kode atribut untuk input atribut baru
Attribute Name	<i>Textbox</i>	Diisikan dengan nama atribut
Add	<i>Button</i>	Digunakan untuk menambah data atribut
Edit	<i>Button</i>	Digunakan untuk mengubah data atribut
Delete	<i>Button</i>	Digunakan untuk menghapus data atribut
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridView
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

C. Master Games

Form ini berfungsi untuk mengisi, mengubah, menghapus data hero apa saja dari replay pertandingan DotA. Form ini memiliki 3 buah tab yaitu Game, Ban Phase dan Pick Phase dimana tab ini diisikan secara berurutan.

C.1 Tab Games

Tab ini digunakan untuk mengisi, mengubah dan menghapus nama tim dan map pada pertandingan. Contoh form Master Map dapat dilihat pada Gambar 3.23.

Game ID		Otomatis Diisi		
Map			▼	
Team A				
Team B				
		☐ Pro Replay		
<u>Add</u> <u>Edit</u> <u>Delete</u> <u>Yes</u> <u>No</u>				
Game ID	Map ID	Team A	Team B	Pro Replay

<u>Add</u> <u>Edit</u> <u>Delete</u> <u>Yes</u> <u>No</u>			
Game ID	Map ID	Team A	Team B
			Pro Replay

Gambar 15. Contoh Tampilan form Pick Phase pada Master Game

Fungsi-fungsi obyek dalam desain form Pick Phase sebagai berikut:

Tabel 3.37. Fungsi obyek form Pick Phase

Nama Obyek	Tipe Obyek	Fungsi
List Games	<i>Gridview</i>	Digunakan untuk memilih game yang akan di ubah atau dihapus
Browse	<i>Button</i>	Digunakan untuk mencari gambar wajah dan nama hero pada pick phase yang kemudian otomatis terisi pada gambar dan textbox
Add	<i>Button</i>	Digunakan untuk menambah data hero pada saat pick phase
Edit	<i>Button</i>	Digunakan untuk mengubah data hero pada saat pick phase
Delete	<i>Button</i>	Digunakan untuk menghapus data hero pada saat pick phase
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data ke DataGridview
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data

D. NN Configuration dan NN Training

Form ini berfungsi untuk menentukan data latih yang akan dilatihkan dan menkonfigurasi NN. Form ini memiliki 2 buah tab yaitu Object Data dan ANN Configuration.

D.1 Tab Object Data

Pada form ini berfungsi untuk menentukan data latih yang akan digunakan untuk pelatihan NN dimana data latih ini akan di alokasikan

untuk tiap input NN pada tab arsitektur NN. Contoh form tab objek data dapat dilihat pada Gambar 3.26

Object Data									
Object Data ID									
View Name									
Note									
Map Versi on	Game ID	Team ID	Team Name	Pro Repl ay	X ID	Hero ID	Hero Name	Hero Typ e	Loca tion
Previous Next Add Edit Delete Simpan Batal Refresh									

Gambar 3.26. Contoh Tampilan Object Data

Fungsi-fungsi obyek dalam desain form Object Data sebagai berikut:

Tabel 3.38. Fungsi objek form Object Data

Nama Obyek	Tip e Obyek	Fungsi
Object Data ID	<i>Textbox</i>	Digunakan untuk memberi nama id pada objek data latih yang dipilih
View Name	<i>Textbox</i>	Digunakan untuk memilih data latih dari view database
Note	<i>Textbox</i>	Digunakan untuk memberikan keterangan tambahan
Previous	<i>Button</i>	Digunakan untuk memilih objek data latih selanjutnya
Next	<i>Button</i>	Digunakan untuk memilih objek data latih sebelumnya
Add	<i>Button</i>	Digunakan untuk menambah objek data latih

Edit	<i>Button</i>	Digunakan untuk mengubah objek data latih
Delete	<i>Button</i>	Digunakan untuk menghapus objek data latih
Simpan	<i>Button</i>	Digunakan untuk menyimpan perubahan data
Batal	<i>Button</i>	Digunakan untuk membatalkan perubahan data
Refresh	<i>Button</i>	Untuk <i>Refresh</i> Data pada gridview

D.2 Tab NN Configuration

Tab ini digunakan untuk mengatur konfigurasi NN yang akan dilatihkan.

Tab konfigurasi NN memiliki 2 tab yaitu Input Group dan Output Group. Contoh form NN Configuration dapat dilihat pada Gambar 3.27 dan 3.28.

ANN Configuration

NN ID

Input Group Qty

I/O Nodes Qty

Output Group Qty

Hidden Node Qty

Learning Const

Learning Iteration

Threshold

Learning Pattern

Input Group

Grup ID

Object ID

Neuron Qty

Max Value

Activation

Min Value

Simpan

Update

Detailed Input Group

Group ID	Object ID	Neuron Qty	Min. Value	Max. Value	Activation

[Data Sebelumnya](#)
[Data Selanjutnya](#)
[Data Baru](#)
[Edit](#)
[Hapus](#)
[Simpan](#)
[Batal](#)
[Refresh](#)

Gambar 3.27. Contoh Tampilan Konfigurasi NN

Fungsi-fungsi obyek dalam desain form Konfigurasi NN sebagai berikut:

Tabel 3.39. Fungsi obyek form Konfigurasi NN

Nama Obyek	Tipe Obyek	Fungsi
NN ID	<i>Textbox</i>	Untuk menunjukkan NN ID
Input Group Qty	<i>Textbox</i>	Untuk menunjukkan jumlah input group
I/O Nodes Qty	<i>Textbox</i>	Otomatis terisi menunjukkan jumlah node setiap group
Output Group Qty	<i>Textbox</i>	Menunjukkan jumlah group output
Hidden Nodes Qty	<i>Textbox</i>	Menunjukkan jumlah node dikalikan dengan jumlah input group yang ada
Learning Const	<i>Textbox</i>	Menunjukkan konstanta pembelajaran



Tabel 3.39. Fungsi obyek form Konfigurasi NN (lanjutan)

Nama Obyek	Tipe Obyek	Fungsi
Learning Iteration	<i>Textbox</i>	Digunakan untuk menunjukkan iterasi / perulangan data yang dilakukan NN
Threshold	<i>Textbox</i>	Untuk menunjukkan batas
Learning Pattern	<i>Textbox</i>	Untuk menunjukkan pola pembelajaran sampai data ke berapa
Update	<i>Button</i>	Digunakan untuk mengubah tipe aktivasi dan object id pada input / output group
Group ID	<i>Textbox</i>	Menunjukkan Group ID input group
Neuron Qty	<i>Textbox</i>	Menunjukkan jumlah node group tersebut
Activation	<i>Combobox</i>	Digunakan untuk memilih fungsi tipe aktivasi pada group input tersebut
Object ID	<i>Combobox</i>	Digunakan untuk menunjukkan Object data ID
Max Value	<i>Textbox</i>	Digunakan untuk menunjukkan nilai maksimal perhitungan
Min Value	<i>Textbox</i>	Digunakan untuk menunjukkan nilai minimal perhitungan
Detailed Input Group	<i>Gridview</i>	Untuk menunjukkan jumlah output dan memilih tipe aktivasi group ID yang akan di ubah
Add	<i>Button</i>	Digunakan untuk menambah data konfigurasi
Edit	<i>Button</i>	Digunakan untuk mengubah data konfigurasi
Delete	<i>Button</i>	Digunakan untuk menghapus data konfigurasi
Yes	<i>Button</i>	Digunakan untuk menyimpan perubahan data konfigurasi
No	<i>Button</i>	Digunakan untuk membatalkan perubahan data konfigurasi

Output Group						
Grup ID		Object ID		▼		
				<u>Simpan</u>		
Neuron Qty		Max Value				
				<u>Update</u>		
Activation		▼	Min Value			
Detailed Output Group						
Group ID	Object ID	Neuron Qty	Min. Value	Max. Value	Activation	MSE

Gambar 3.28. Contoh Tampilan Output Grup dalam Konfigurasi NN

Fungsi-fungsi obyek dalam desain form Output Grup dalam Konfigurasi NN sebagai berikut:

Tabel 3.40. Fungsi form Output Grup dalam Konfigurasi NN

Nama Obyek	Tipe Obyek	Fungsi
Group ID	<i>Textbox</i>	Menunjukkan Group ID output group
Neuron Qty	<i>Textbox</i>	Menunjukkan jumlah node group tersebut
Activation	<i>Combobox</i>	Digunakan untuk memilih fungsi tipe aktivasi pada group output tersebut
Object ID	<i>Combobox</i>	Digunakan untuk menunjukkan Object data ID
Max Value	<i>Textbox</i>	Digunakan untuk menunjukkan nilai maksimal perhitungan
Min Value	<i>Textbox</i>	Digunakan untuk menunjukkan nilai minimal perhitungan
Update	<i>Button</i>	Untuk memilih group ID yang akan di ubah
Detailed Output Group	<i>Gridview</i>	Untuk menunjukkan jumlah output dan memilih tipe aktivasi group ID yang akan di ubah

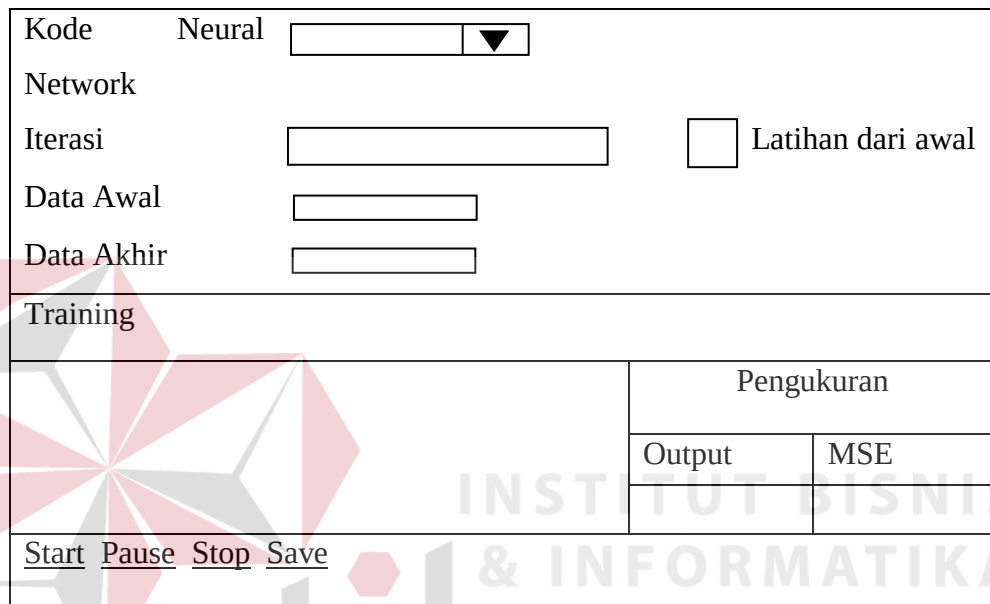
E. NN Training

Form ini berfungsi untuk melihat dan melakukan pelatihan NN. Form ini memiliki 3 buah tab yaitu Training, Detailed Neural Network dan Data.

E.1 Tab NN Training

Tab ini digunakan untuk menjalankan dan melihat proses pelatihan NN.

Contoh form Training dapat dilihat pada Gambar 3.29



Kode Neural		▼
Network		
Iterasi		☐ Latihan dari awal
Data Awal		
Data Akhir		
Training		
		Pengukuran
		Output
		MSE
Start Pause Stop Save		

Gambar 3.29. Contoh Tampilan Training NN

Fungsi-fungsi obyek dalam desain form Training NN sebagai berikut:

Tabel 3.41. Fungsi form Training NN

Nama Obyek	Tipe Obyek	Fungsi
Kode NN	<i>Combobox</i>	Untuk memilih kode NN
Iterasi	<i>Textbox</i>	Untuk menunjukka sudah berapa kali iterasi yang sudah dilakukan
Data Awal	<i>Textbox</i>	Untuk menunjukkan data awal dari data yang dilatihkan

Tabel 3.41. Fungsi form Training NN (lanjutan)

Nama Obyek	Tipe Obyek	Fungsi
Data Akhir	<i>Textbox</i>	Untuk menunjukkan batas data akhir dari data yang dilatihkan
Latihan dari awal	<i>Checkbox</i>	Untuk memilih apakah proses pelatihan dimulai dari awal atau melanjutkan yang sebelumnya.
Pengukuran	<i>Gridview</i>	Untuk menunjukkan berapa jumlah output dan MSE (Mean Square Error)
	<i>Picture box</i>	Untuk menampilkan grafik Actual data dan Neural network selama pelatihan
Play	<i>Button</i>	Untuk memulai jalanya pelatihan
Pause	<i>Button</i>	Untuk pause jalanya pelatihan
Stop	<i>Button</i>	Untuk memberhentikan jalanya pelatihan
Save	<i>Button</i>	Untuk menyimpan hasil pelatihan (bobot)

E.2 Tab Detailed Neural Network

Tab ini digunakan untuk melihat proses pelatihan NN yang dipilih secara lebih detail. Contoh form Detailed Neural Network dapat dilihat pada Gambar

3.30

Detailed Neural Network							
Kode NN							
Jml Grup Input				Iterasi Belajar			
Jml Grup Output				Toleransi			
Jml Node Hidden				Pola Belajar			
Konst. Belajar							
Input Group							
Group ID	Object ID	Obyek ID	Batas Max	Batas Min	Tipe Aktivasi		
Output Group							
Group ID	Object ID	Obyek ID	Batas Max	Batas Min	Tipe Aktivasi	MSE	MAPE

Gambar 3.30. Contoh Tampilan Detailed Neural Network

Fungsi-fungsi obyek dalam desain form Detailed Neural Network sebagai berikut:

Tabel 3.42. Fungsi form Detailed Neural Network

Nama Obyek	Tipe Obyek	Fungsi
Kode NN	<i>Textbox</i>	Untuk menunjukkan NN yang dilatihkan
Jml Grup Input	<i>Textbox</i>	Untuk menunjukkan total jumlah group input
Jml Grup Output	<i>Textbox</i>	Untuk menunjukkan jumlah group output
Jml Node Hidden	<i>Textbox</i>	Untuk menunjukkan total jumlah node hidden
Konst. Belajar	<i>Textbox</i>	Untuk menunjukkan konstanta belajar
Iterasi Belajar	<i>Textbox</i>	Untuk menunjukkan sudah berapa kali iterasi yang sudah dilakukan
Toleransi	<i>Textbox</i>	Untuk menunjukkan toleransi pembelajaran
Pola Belajar	<i>Textbox</i>	Untuk menunjukkan pola pembelajaran sampai data ke berapa
Input Group	<i>Gridview</i>	Untuk menampilkan berapa jumlah Input group dari kode NN yang dipilih
Output Group	<i>Gridview</i>	Untuk menampilkan jumlah hasil Output group dari kode NN yang dipilih

F.DotA CM

Form terakhir ini berfungsi untuk melakukan simulasi DotA CM. Contoh form DotA CM dapat dilihat pada Gambar 3.31

Tim 1 Ban Phase

[Browse](#) [Browse](#) [Browse](#) [Browse](#)

Pick Phase

[Browse](#) [Browse](#) [Browse](#) [Browse](#) [Browse](#)

Map ▼ Ne

Tim 2 Ban Phase

Pick Phase

Process

No.	Hero Name	% Suitability	Information	No.	Hero Name	% Suitability	Information

Select Hero

Select Hero

Gambar 3.31. Contoh Tampilan form DotA CM

Fungsi-fungsi obyek dalam desain form Dota CM Network sebagai berikut:

Tabel 3.43. Fungsi form DotA CM

Nama Obyek	Tipe Obyek	Fungsi
Map	<i>Combobox</i>	Digunakan untuk memilih map yang akan digunakan
New Game	<i>Button</i>	Digunakan untuk memulai sebuah simulasi baru
Browse	<i>Button</i>	Digunakan untuk memilih hero
Process	<i>Button</i>	Digunakan untuk mengkonfirmasi pilihan hero dari tim 1
Hero Recommendation	<i>Datagrid</i>	Untuk menampilkan list rekomendasi hero untuk tim 2

