BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Masalah yang diangkat dalam penulisan tugas akhir ini adalah penerapan algoritma *neural network backpropagation* untuk mencari solusi pada permainan dazzle. Dalam permainan dazzle terdapat minimal 1 pemilihan jalur, dan maksimal memiliki 4 pemilihan jalur untuk menggabungkan karakter yang memiliki *property* yang sama. Semakin banyak kombinasi karakter yang terdapat pada area permainan maka semakin banyak permasalahan dalam pemilihan jalur untung penggabungan karakter. Maka dibutuhkan algoritma yang dapat menentukan pemilihan jalur dalam penggabungan karakter pada permainan dazzle. Algoritma *neural network backpropagation* adalah algoritma yang dapat mengatasi masalah pemilihan jalur pada permainan dazzle. Dengan banyaknya jalur yang tersedia untuk digabungkan maka, akan membentuk sebuah *pattern* dalam *state* tertentu. Algoritma *neural network backpropagation* ini akan melatih komputer untuk menentukan jalur penggabungan karakter dari data yang sudah ada sebelum, karena algoritma *neural network backpropagation* termasuk kedalam algoritma yang menggunakan metode pembelajaran terawasi (*Supervised Learning)* sehingga algoritma tersebut membutkan data training yang digunakan untuk mengenali *pattern* dari state permainan dazzle agar dapat menentukan jalur penggabungan karakter.

3.1.1 Analisis Permainan Dazzle

Permainan dazzle (dagelan puzzle) adalah game bergenre *endless puzzle* yang di kembangkan oleh Touch Ten game dazzle tersedia untuk perangkat mobile pada flatform ios dan android. Pada game ini terdapat sebuah matrik untuk area permainan berukuran 4x4. Tantangan pada game ini adalah menggabungkan karakter yang memiliki jenis yang sama sehingga muncul karakter baru dari penggabungan karakter sebelumnya. Adapun aturan dalam permainan dazzle adalah sebagai berikut :

1. Pemain hanya dapat menggeserkan satu karakter saja dan karakter tersebut hanya dapat digeserkan ke kiri, ke kanan, ke bawah, ke atas. Dimana posisi matrik yang akan digeserken kosong atau tidak diisi oleh karakter lain.
2. Pemain hanya dapat menggabungkan dua karakter yang sejenis dengan status karakter yang sama.
3. Jika karakter digeserkan maka pada setiap baris atau kolom dari arah pergeseran akan ikut bergeser dan akan muncul karakter baru yang memiliki status 1 muncul pada area permainan, berlawanan dengan arah pergeseran yang dilakukan.
4. Setiap karakter yang memiliki status 4 digabungkan maka karakter tersebut akan hilang dari area permainan atau dan menjadi matriks kosong yang tidak ada isi karakter yang digabungkan.
5. Jika pemain menggabungkan dua karakter secara berurutan makan pada urutan yang ke dua akan muncul karakter baru pada ujung matrik dari karakter yang geserkan.
6. Jika semua area matriks sudah terisi oleh karakter yang tidak dapat digabungkan maka permaian telah selesai.
7. Skor permainan didapat dari seberanya banyak pemain dapat menggabungkan karakter yang memiliki jenis yang sama. Berikut adalah peraturan mendapatkan skor dari permainan dazzle

   1 + 1 = 10 point

   Jika melakukan penggabungan dua kali secara berurutan skor akan di kalikan 2. Begitupun jika penggabungan dilakukan secara tiga kali berurutan skor akan dikalikan 3

   2 + 2 = 20 point

   Jika melakukan penggabungan dua kali secara berurutan skor akan di kalikan 2. Begitupun jika penggabungan dilakukan secara tiga kali berurutan skor akan dikalikan 3

   3 + 3 = 30 point

Jika melakukan penggabungan dua kali secara berurutan skor akan di kalikan 2. Begitupun jika penggabungan dilakukan secara tiga kali berurutan skor akan dikalikan 3

4 + 4 = 40 point

Jika melakukan penggabungan dua kali secara berurutan skor akan di kalikan 2. Begitupun jika penggabungan dilakukan secara tiga kali berurutan skor akan dikalikan 3.

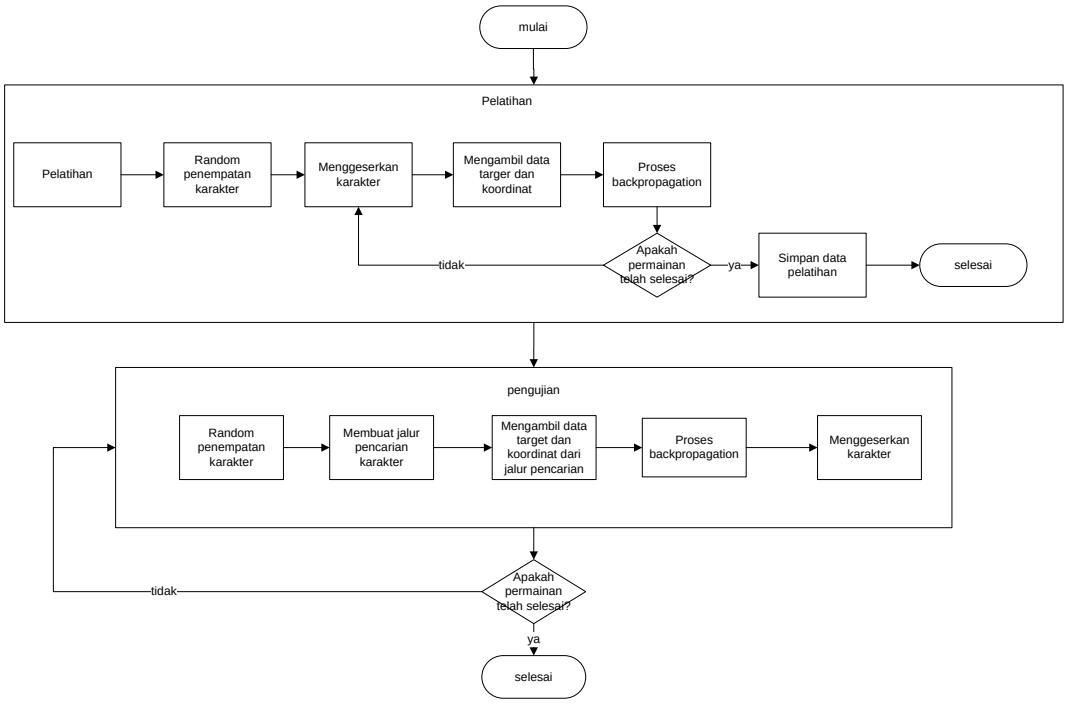
Pada permainan ini memiliki ukuran 4x4 dan terdapat 4 karakter saat memulai permainan. 4 karakter tersebut terdiri dari 2 pasang jenis karakter yang berbeda. Penggabungan karekter yang dilakukan terdapat 4 tahap penggabungan, dimana setiap penggabungan karakter diwakili oleh angka yaitu :

Tabel 3.1 Daftar Keterangan Penggabunan Karakter

Status	Jenis karakter	Penjelasan
1a	1	Status awal jenis karakter ke 1a
2a	1	Hasil penggabungan karakter status 1a
3a	1	Hasil penggabungan karakter status 2a
4a	1	Hasil penggabungan karakter status 3a
1b	2	Status awal jenis karakter ke 1b
2b	2	Hasil penggabungan karakter status 1b
3b	2	Hasil penggabungan karakter status 2b
4b	2	Hasil penggabungan karakter status 3b

3.1.2 Analisis sistem

Pada bagian ini akan dijelaskan bagaimana proses pada sistem yang akan dibangun. berikut adalah proses yang akan dibuat :



Gambar 3.1 *Block Diagram* Sistem Yang Akan Dibangun

3.1.3 Analisis Data Masukan

terdapat nilai input pada awal permainan pada program yang akan dibuat yaitu menggunakan titik awal penempatan 2 pasang karakter yang memiliki jenis yang berbeda pada area permainan, 2 pasang karakter ini terdiri dari empat karakter. Untuk memasukan data titik awal penempatan 2 pasang karakter ditempatkan secara acak pada area permainan.

Pada permainan dazzle ini pemilihan jalur penggabungan karakter dipengaruhi oleh nilai dari karakter yang ada pada area permaianan, sehingga data masukan untuk arsitektur jaringan yang digunakan adalah sebagai berikut :

1. Input 1 : nilai dari karakter yang akan digeserkan
2. Input 2 : nilai dari karakter yang ada disebelah kiri karakter yang akan digeserkan.
3. Input 3 : nilai dari karakter yang ada disebelah atas karakter yang akan digeserkan.
4. Input 4 : nilai dari karakter yang ada disebelah bawah karakter yang akan digeserkan.
5. Input 5 : nilai dari karakter yang ada disebelah kanan karakter yang akan digeserkan.

3.1.4 Analisis Metode

Pada bagian ini dijelaskan sebuah kasus dengan nilai parameter yang telah ditetapkan, kemudian diproses secara manual untuk menghasilkan output. Parameter yang digunakan dalam kasus yang digunakan adalah sebagai berikut :

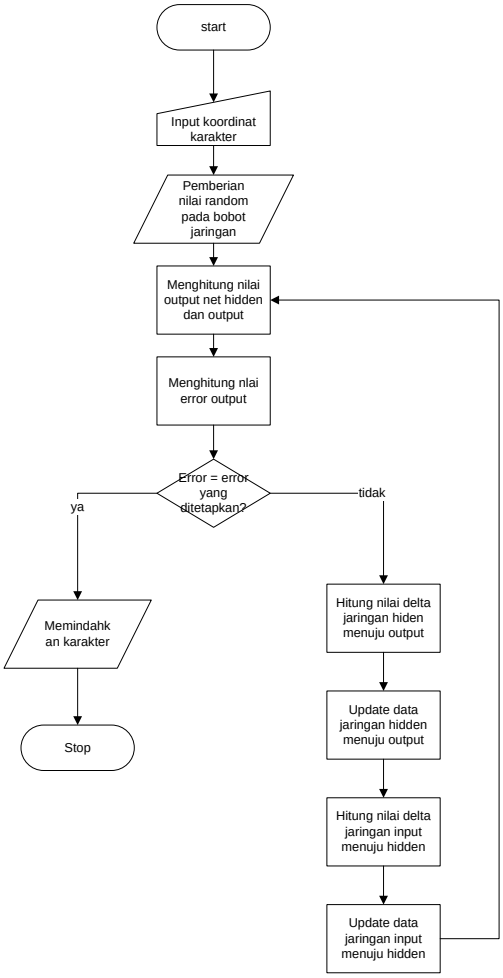
α (learning rate) = 0.5
error = 0.0001
jumlah hidden layer = 2

3.1.5 Analisis Algoritma Neural Network Backpropagation

Algotirma *neural network backpropagation* adalah algoritma yang menggunakan metode *survevised learning* (pembelajaran terawasi) oleh karena itu algoritma ini membutuhkan data-data yang ada untuk dilakukan *training* pada data tersebut sehingga menghasilkan *output* yang mendekati atau sama dengan output yang diinginkan. Algoritma ini akan menghasilkan nilai output pada proses *feed forward* untuk mengetahui *error* yang dihasilkan dari *data output* tersebut. Selanjutnya akan dilakukan proses *backpropagation* berdasarkan *error* dari *data output*, proses dari *backpropagation* ini akan menghasilkan nilai yang digunakan untuk mengubah nilai bobot-bobot pada jaringan neuron. Tahapan ini akan dilakukan berulang-ulang sampai nilai *error* dari *output* mencapai nilai *error* yang telah

ditentukan sehingga nilai dari data *output* dapat mendekati atau sama dengan nilai data output yang diharapkan.

Berdasarkan langkah-langkah proses algoritma *neural network backpropagation* dapat dilihat dalam bentuk flowchart pada gambar , berikut merupakan *flowchart* proses algoritma *neural netwrok backpropagatio*



Gambar 3.2 Flowchart Algoritma Neural Network Bakcpropagation

3.1.6 Analisis Pembuatan Pohon Pencarian Jalur Penggabungan Karakter

Pada analisis jaringan *neural network backpropagation* ini terdapat proses tambahan sebelum menggunakan algoritma *backpropagation* dalam pemilihan jalur untuk penggabungan karakter. Proses pertama adalah menggunakan metode *greedy* yang dimodifikasi, serta dilakukan *pruning* saat membuat pohon pencarian untuk menghemat memory yang digunakan dan untuk menghemat waktu proses pembuatan pohon pencarian. Proses pertama dilakukan bertujuan untuk mendapatkan posisi karakter yang akan digeserkan berserta nilai target pergeseran yang akan dilakukan. Algoritma *neural network backpropagation* ini adalah jenis algoritma *supervised learning,* sehingga algoritma *backpropagation* akan optimal ketika memiliki banyak data training. Proses pertama ini juga bertujuan untuk mengoptimalkan pemilihan jalur penggabungan karakter ketika data training yang digunakan masih sedikit. Adapun contoh kasus pada permainan dazzle dapat digambarkan sebagai berikut :

	1	2	3	4
1	0	0	0	1b
2	0	1a	1a	0
3	0	0	0	0
4	0	0	0	1b

Gambar 3.3 Contoh Kasus Permainan Dazzle

Proses pembuatan pohon pencarian ini akan menggunakan data pada gambar 3.3. proses yang digunakan untuk membuat pohon pencarian ini adalah sebagai berikut :

Memilih Jalur terpendek untuk menggabungkan karakter, didapatkan dengan cara seperti berikut :

Jarak penggabungan = |(X1-X2)|+|(Y1-Y2)|

X1: posisi karakter yang akan digeserkan

X2: tujuan karakter yang akan digubungkan dengan x1

Y1: posisi karakter yang akan digeserkan

Y2: tujuan karakter yang akan digubungkan dengan y1

Jalur terpendek = |(2-2)|+|(2-3)|

= |0|+|-1|

= 1

Selanjutnya dilakukan perhitungan yang sama pada setiap karakter yang ada pada area permainan. Sehingga didapkan hasil sebagai berikut :

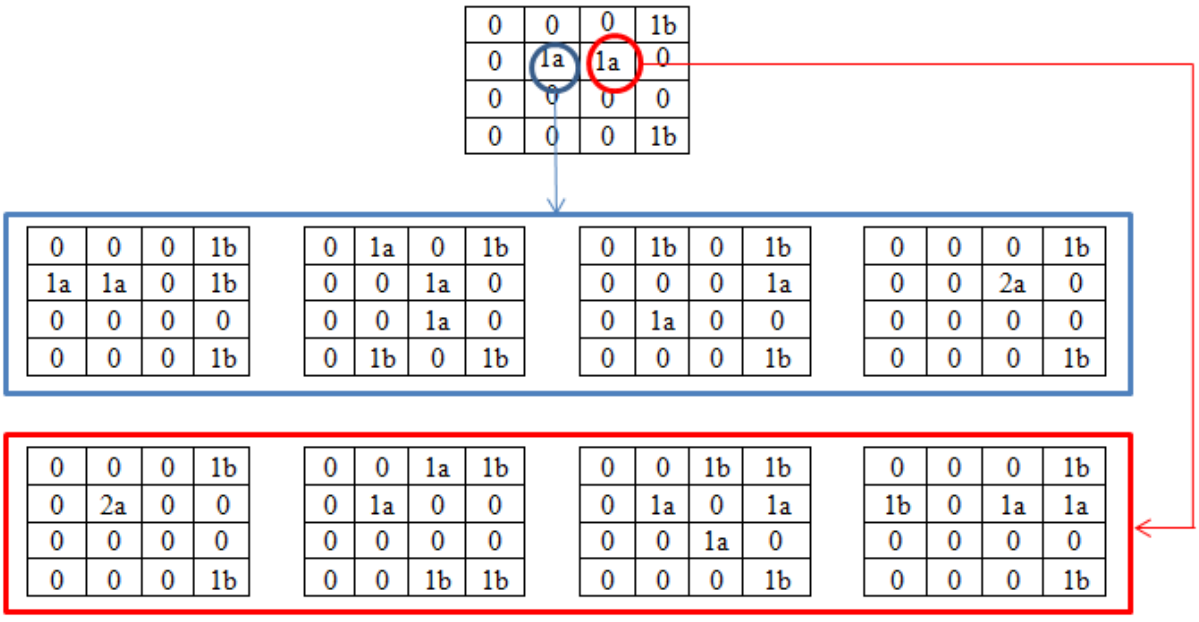
1.4 → 4.4 = 3

2.2 → 2.3 = 1

2.3 → 2.2 = 1

4.4 → 1.4 = 3

Dari hasil tersebut didapatkan kandidat yang akan dipilih adalah karakter yang berada pada koordinat 2.2 dan 2.3. selanjutnya akan dilakukan pemilihan karakter yang mana yang akan digeserkan. Maka akan dilakukan pohon pencarian untuk setiap kemungkinan pergeseran dari karakter pada koordinat 2.2 dan 2.3.



Gambar 3.4 Pohon kemungkinan jalur yang optimal

Dari proses tersebut didapatkan nilai yang optimal untuk penggabungan ada pada koordinat 2.2 digeser ke arah kanan dan pada korrdinat 2.3 digeser ke arah kiri.

Dari proses ini akan dilakukan proses selanjutnya untuk pembuatan pohon pencarian. proses selanjutnya adalah menghitung jumlah kemungkinan karakter yang dapat digabungkan. Proses ini dapat dilihat pada tabel 3.2

Tabel 3.2 Kemungkinan Kemunculan Karakter Yang Memiliki Jalur Terpendek

Koordinat karakter terpilih	Nilai karakter pada koordinat x,y																Kemungkinan kemunculan karakter yang memiliki jalur terpendek untuk penggabungan
	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4.1	4.2	4.3	4.4	
2.2	0	0	0	1b	0	0	2a	1b	0	0	0	0	0	0	0	1b	0
2.2	0	0	2a	1b	0	0	0	0	0	0	0	0	0	0	1b	1b	1
2.2	1b	0	0	1b	0	0	0	2a	0	0	0	0	0	0	0	1b	1
2.2	0	0	1b	1b	0	0	0	0	0	0	2a	0	0	0	0	1b	1
2.3	0	0	0	1b	2a	0	0	1b	0	0	0	0	0	0	0	1b	1
2.3	0	2a	0	1b	0	0	0	0	0	0	0	0	0	1b	0	1b	0
2.3	0	1b	0	1b	0	0	0	0	0	2a	0	0	0	0	0	1b	0
2.3	0	0	0	1b	1b	0	2a	0	0	0	0	0	0	0	0	1b	0

Hasil pada tabel 3.6 menunjukan bahwa karakter pada koordinat 2.3 memiliki 3 kemungkinan pergeseran yang memiliki jalur terpendek untuk penggabungan karakter. 3 kemungkinan tersebut adalah ketika karakter pada koordinat 2.3 di geserkan ke kanan, ke atas dan kebawah. Sehingga karakter pada koordinat 2.2 akan dipruning dan tidak dilanjutkan untuk membuat pohon pencarian. proses selanjutnya adalah melakukan pohon pencarian pada hasil pergeseran yang dilakukan sebelumnya.

3.1.7 Penerapan Algoritma Neural Network *Backpropagation*

Pada bagian ini akan menjelaskan proses proses yang ada pada bagian algoritma *backpropagation*. Tahap yang dilakukan adalah merubah data pohon pencarian menjadi *input* untuk proses *backpropagation*. Pada permainan dazzle

terdapat beberapa karakter yang berada pada area permainan, seperti pada gambar 3.3 menunjukan posisi karakter tersebut ditempatkan. Pada proses pemilihan jalur terpendek dapat dilihat nilai input yang digunakan adalah sebagai berikut :

Nilai input 1 : nilai karakter yang akan digeserkan
: 1

Nilai input 2 : nilai dari karakter yang ada dibagian kiri karakter yang akan digeserkan
: 1

Kemungkinan pergeseran akan diberikan nilai jika karakter tersebut dapat digeserkan pada matriks kosong atau digeserkan pada karakter yang memiliki status yang sama untuk digabungkan dan akan diberi nilai sesuai dengan nilai dari karakter yang beradapa pada posisi tersebut, jika kondisi tersebut tidak terpenuhi maka akan diberi nilai 0.

Nilai input 3 : nilai dari karakter yang ada dibagian atas karakter yang akan digeserkan
:0

Nilai input 4 : nilai dari karakter yang ada dibagian bawah karakter yang akan digeserkan
: 0

Nilai input 5 : nilai dari karakter yang ada dibagian kanan karakter yang akan digeserkan
: 0

Nilai output1 : target pergeseran karakter
: 1000 (kanan)

Pada nilai output terdapat nilai untuk menentukan pergeseran karakter. Nilai normalisasi pergeseran didapatkan menggunakan persamaan 2.3. Nilai yang ada dapat dilihat pada tabel 3.2

Tabel 3.3 Nilai Pergeseran Karakter

Pergeseran	Nilai pergeseran	Nilai desimal pergeseran	Nilai normalisasi pergeseran	Treshold
Kiri	0001	1	0.125	0 - 0.125
Atas	0010	2	0.25	0.126 - 0.25
Bawah	0100	4	0.5	0.26 - 0.5
kanan	1000	8	1	0.6 - 1

. Pada kasus ini nilai yang digunakan untuk learning rate adalah 0,5. Karena algoritma *neural network backpropataion* hanya dapat menerima inputan dari 0 – 1 maka nilai input dan output yang tidak sesuai dengan aturan *backpropagation* akan dilakukan normalisasi dengan menggunakan persamaan 2.3. Berikut adalah tabel normalisasi dari nilai input dapat dilihat pada tabel 3.7

Contoh normalisasi pada karakter :

$$V' = \frac{nilai\ karakter - nilai\ karakter\ terkecil}{nilai\ karakter\ terbesar - nilai\ karakter\ terkecil}$$

$$V' = \frac{1-0}{4-0}$$

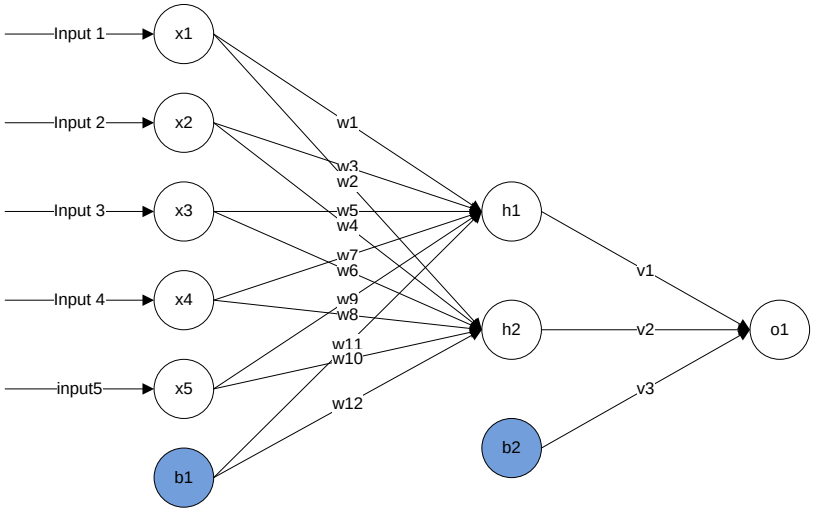
$$V' = \frac{1}{4}$$

$$V' = 0.25$$

Tabel 3.4 Normalisasi Nilai Pada Karakter Dipermainan Dazzle

No	Status karakter
0	0
1	0.25
2	0.5
3	0.75
4	1

Pada penelitian ini struktur *neural network backpropagation* yang digunakan dapat dilihat pada gambar 3.5.



Gambar 3.5 Struktur jaringan neural network backpropagation

Tahap pertama adalah pemberian bobot nilai secara acak pada setiap jaringan input yang terhubung ke layer hidden :

Tabel 3.5 Nilai Acak Bobot Variabel W

input	hidden 1	hidden2
1	0,38505625	0,106391
2	0,599540335	0,904816
3	0,887665583	0,964175
4	0,017552359	0,413133
5	0,599540335	0,904816
bias	0,920936866	0,196275

Lalu dilakukan pemberian bobot nilai secara acak terhadap jaringan hiden layer menuju output :

Tabel 3.6 Nilai Acak Bobot Variabel V

hidden	Output
1	0,846604
2	0,762539
bias	0,897255

3.1.8 Forward Propagation

Tahap selanjutnya adalah tahap *forward propagation* tahap ini akan menghitung nilai output dari hidden layer dan output dari bobot nilai acak yang sudah ada. Lalu akan dihitung nilai error dari output yang dihasilkan.

tahap pertama adalah menhitung nilai output dari hidden layer. Perhitungannya adalah sebagai berikut :

nethidden1 = w1*x1+w2*x2+w3*x3+w4*x4....+w6*x6+bias*1

nethidden1 = 1.183393044

selanjutnya output dihitung nilai aktivasi neth 1 menggunakan fungsi sigmoid dengan perhitungan seperti berikut :

$$h1 = \frac{1}{1 + exp^{-1.183393044}}$$

$$= 0.765557191$$

Lakukan proses yang sama untuk menghitung nilai neth 2 dan didapatkan hasil sebagai berikut :

$$h2 = \frac{1}{1 + exp^{-1.383854215}}$$

$$= 0.799609142$$

Tahap selanjutnya adalah menghitung nilai output dari output1, output 2, output3 dan output4 berikut adalah perhitungan yang dilakukan :

output 1 = v1*h1+v2*h2+v3*b2

netoutput1 = 1.697119295

selanjutnya output dihitung nilai aktivasi output 1 menggunakan fungsi sigmoid dengan perhitungan seperti berikut :

$$o1 = \frac{1}{1 + exp^{-1.697119295}}$$

$$= 0.845157974$$

Selanjutnya adalah menghitung nilai error dari output yang dihasilkan dengan perhitungan sebagai berikut :

$$\text{Erroruput1} = \frac{1}{2}(target - output)^2$$

$$= \frac{1}{2}(1 - 0.845157974)^2$$

$$= 0.023976053$$

Pada perhitungan diatas didapatkan hasil 0.023976053 dari *error output* yang dihasilkan. Karena error yang dihasilkan masih terlalu besar dari target *error* yang ditetapkan sebesar 0,001 dan nilai output yang dihasilkan jauh dari nilai target yang ditetapkan maka akan dilakukan proses pembelajaran dengan menggunakan metode *backpropagation* untuk merubah bobot jaringan yang ada, selanjutnya dilakukan proses perhitungan *forward propagation* untuk menghasilkan nilai output mendekati atau sama dengan nilai target yang ditetapkan dan untuk meminimalkan nilai *error* dari *output* yang dihasilkan.

3.1.9 Backpropagation

Tahap selanjutnya adalah mecari nilai delta 1 dari *output* yang dihasilkan. Nilai delta 1 ini akan digunakan pada perhitungan mencari nilai delta bobot *V*. Dengan menggunakan persamaan 2.14 didapatkan hasil sebagai berikut :

$\delta 1 = 0.020263552$

Tahap selanjutnya mencari nilai delta bobot dari *V*. Dengan menggunakan persamaan 2.15 didapatkan hasil sebagai berikut :

$\Delta v1 = 0.007756454$

$\Delta v2 = 0.008101461$

$\Delta v3 = 0.001962226$

Tahap selanjutnya menghitung delta net *hidden layer*, nilai delta net iniakan digunakan untuk mencari nilai delta *W*. Dengan menggunakan persamaan 2.17 Didapatkan hasil sebagai berikut :

δ net1 = 0.018278445

δ net2 = 0.006648563

δ net3 = 0.018373681

Tahap selanjutnya menghitung nilai delta *W*, nilai delta *hidden* 1 dan 2 ini akan digunakan untuk mencari delta bobot variable V input menuju hidden layer. Dengan menggunakan persamaan 2.18 Didapatkan hasil sebagai berikut :

δ hidden1 = 0.003280604

δ hidden2 = 0.001065328

Tahap selanjutnya menghitung nilai delta bobot *W*. Hasil nilai yang didapat akan digunakan untuk mencari nilai perubahan bobot W. Dengan menggunakan persamaan 2.19 dan 2.20 Didapatkan hasil sebagai berikut :

Δw1 = 0.001640302

Δw2 = 0.001640302

Δw3 = 0

Δw4 = 0

Δw5 = 0

Δw6 = 0.000528128

Δw7 = 0.000532664

Δw8 = 0.000532664

Δw9 =0

Δw10 = 0

Δw11 = 0

Δw12 = 0.000171502

tahap selanjutnya adalah menghitung delta bobot V. Dengan menggunakan persamaan 2.19 dan 2.20 Di dapatkan hasil sebagai berikut :

v1 = 0.909792039

v2 = 0.336205956

v3 = 0.908697661

tahap selanjutnya menghitung nilai perubahan bobot baru variable W. Dengan menggunakan persamaan 2.21. Didapatkan hasil sebagai berikut :

w1 = 0.920216919

w2 = 0.052209472

w3 = 0.585955382

w4 = 0.769066439

w5 = 0.844203009

w6 = 0.665953709

w7 = 0.323610577

w8 = 0.852313235

w9 = 0.193670461

w10 = 0.829157384

w11 = 0.050416578

w12 = 0.65652349

tahap selanjutnya menghitung nilai perubahan bobot baru pada variabel V. Dengan menggunakan persamaan 2.22. Didapatkan hasil sebagai berikut :

v1 = 0.909792039

v2 = 0.336205956

v3 = 0.908697661

Setelah dilakukan proses backpropagation, didaptkan hasil nilai dari pelatihan dengan nilai error 0.001 yang telah dilakukan. didapatkan hasil dari nilai output 1 = 0.64201 dan berhenti pada looping ke 17 berdasarkan treshold yang ditetapkan nilai output yang dihasilkan akan menggeserkan karakter ke arah kanan.

3.2 Analisis Kebutuhan Non fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga terdiri dari elemen atau komponen-komponen yang dibutuhkan oleh sistem yang akan dibangun sampai dengan sistem tersebut terimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran

yang diinginkan. Adapun kebutuhan yang digunakan dalam implementasi ini sebagai berikut :

1. Kebutuhan perangkat keras
2. Kebutuhan perangkat lunak

Analisis kebutuhan non fungsional bertujuan agar penerapan implementasi dapat digunakan sesuai dengan kebutuhan.

3.2.1 Analisis Kebutuhan Perangkat Keras

Untuk mengimplementasikan algoritma *neural network backpropagation*, diperlukan spesifikasi perangkat keras yang mendukung proses kerja dari suatu sistem. Adapun spesifikasi perangkat keras yang dibutuhkan dalam pembangunan aplikasi game dazzle seperti terdapat pada tabel 3.7 sebagai berikut :

Tabel 3.7 Spesifikasi Perengkat Keras

No	Perangkat Keras	Spesifikasi
1	CPU	Intel Dual core 1.80 GHz
2	VGA	Intel(R) HD Graphics
3	Harddisk	80 GB
4	RAM	2 GB

Sedangkan kebutuhan perangkat keras untuk menjalankan aplikasi yang dibangun, yang harus dipenuhi yaitu seperti pada tabel 3.8.

Tabel 3.8 Spesifikasi Perangkat Keras

Nama Perangkat	Spesifikasi
CPU	Intel (R) dual Core(TM) 2,1GHz
RAM	2 GB
Harddisk	Min. 80 GB
VGA Card	Intel (R) HD Graphics

3.2.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras dapat saling berinteraksi diantara keduanya. Perangkat lunak (*software*) merupakan hal terpenting dalam mendukung kineja sebuah sistem. Berikut ini, kebutuhan perangkat lunak untuk menerapkan algoritma *Neural Network Backpropagation* pada permainan dazzle, dapat dilihat pada tabel 3.9.

Tabel 3.9 Kebutuhan Perangkat Lunak

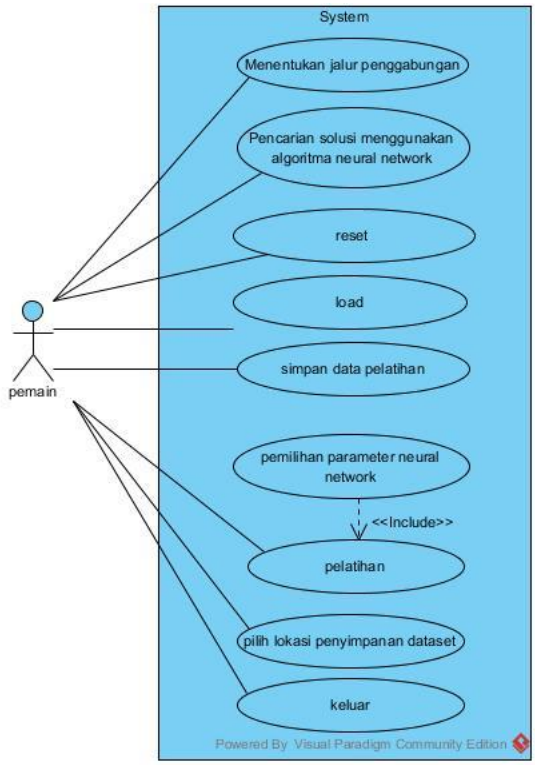
Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Windows 7
Bahasa Pemrograman	C#
IDE	Visual Studio 2012
.NetFramework	4.5

3.2.3 Analisis Kebutuhan Fungsional

Pemodelan yang digunakan pada *game* yang akan dibuat adalah analisis perancangan dan pengembangan perangkat lunak berbasis objek, dimana alat yang akan digunakan untuk pengembangan perangkat lunak yaitu pemodelan *Unified Modeling Language* (UML). Pemodelan dalam analisis tersebut antara lain pembuatan *use case diagram*, definisi aktor, definisi *use case*, skenario *use case*, *activity diagram, sequence diagram, dan class diagram.*

3.2.4 Use Case Diagram

Diagram *use case* menggambarkan hubungan antara aktor dan sistem yang dibangun. Perancangan proses-proses yang terdapat pada *game* yang akan dibangun akan digambarkan dengan diagram *use case*, dapat dilihat pada



Gambar 3.6 Use case diagram

3.2.5 Definisi Aktor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada diagram *use case*. Definisi aktor dapat dilihat pada tabel 3.10

Tabel 3.10 Definisi Aktor

No.	Aktor	Deskripsi
1.	Pemain	Pengguna yang memainkan *game.*

3.2.6 Definisi Use Case

Definisi *use case* berfungsi untuk menjelaskan fungsi *use case* yang terdapat pada diagram *use case.* Definisi *use case* dapat dilihat pada tabel 3.11

Tabel 3.11 Definisi Use Case

No.	Use Case	Keterangan
1.	Menentukan jalur penggabungan	Proses proses menentukan jalur pengaabungan karakter pada area permainan
2.	Pencarian solusi menggunakan algoritma *neural network backpropagation*	Proses pencarian solusi menggunakan algoritma neural netrwok backpropagation
3.	pelatihan	Proses pelatihan yang digunakan sebagai data training.
4.	Simpan data pelatihan	Proses untuk menyimpan data training.
5.	*Reset*	Proses penentuan posisi awal karakter yang berada pada area permaianan
6.	*Load*	Proses mengambil dataset yang akan digunakan untuk proses perhitungan neural network pada menu bermain
7.	Pilih lokasi peyimpanan dataset	Proses memilih lokasi penyimpanan dataset yang dipilih oleh pemain
8.	Parameter *neural network*	Proses pemilihan parameter yang dilakukan pemain untuk tahapan pencarian solusi menggunakan algoritma neural network
9.	keluar	Proses untuk keluar dari permainan

3.2.7 Skenario Use Case

Skenario pada setiap bagian pada *use case* menunjukkan proses apa saja yang terjadi pada setiap bagian dan respon apa yang dikeluarkan oleh sistem kepada pengguna memberikan perintah pada setiap bagian-bagian *use case* seperti terdapat pada tabel 3.12

Tabel 3.12 Skenario Use Case Menentukan Jalur Penggabungan

Identifikasi	
Nomor	U – 1
Nama	Menentukan jalur pengabungan
Tujuan	Mengabungkan karakter yang memiliki jenis dan status yang sama
Deskripsi	Mengabungkan karakter sesuai jalur yang tersedia.
Aktor	User
Skenario Utama	
Kondisi awal	Pemain berada di dalam permainan dan sudah memilih posisi awal karakter.
Aksi aktor	Reaksi Sistem
Memasukan nilai koordinat x,y dari posisi karakter	
	Sistem memindahkan karater
	Menampilkan penggeseran karakter pada posisi yang baru.
skenario alternatif	

	2a. karakter tetap diposisi sebelumnya
	3a. Menampilkan penggabungan karakter pada posisi yang baru.
Kondisi Akhir	Sistem menampilkan penggabungan karakter pada jalur yang dipilih user

Tabel 3.13 Skenario Use Case Pencarian Solusi Menggunakan Algoritma Neural Network Backpropagation

Identifikasi	
Nomor	U – 2
Nama	Solusi
Tujuan	Menjalankan otomatis pemilihan jalur penggabungan karakter
Deskripsi	Proses proses pemilihan jalur otomatis menggunakan algoritma neural network backpropagation
Aktor	pemain
Skenario Utama	
Kondisi awal	Pemain memilih menu pencarian solusi menggunakan algoritma neural network backpropagation
Aksi aktor	Reaksi Sistem
Memasukan data input pelatihan i1,i2,i3,i4,i5 dan output pergeseran	
	Membuat pohon pencarian

	Mengambil data target dan posisi karkater yang akan digeserkan
	Mencari mengambil jenis karakter pada pohon pencarian yang terpilih
	Menggeserkan karakter
	menggeserkan karakter pada posisi yang baru
Skenario alternatif	
	4a. Karakter tetap diposisi sebelumnya
	5a. menggabungkan karakter pada posisi yang baru
Kondisi Akhir	Karakter berhasil digeserkan dari posisi awal menuju posisi baru.

Tabel 3.14 Skenario Use Case Pelatihan

Identifikasi	
Nomor	U - 3
Nama	Pelatihan
Tujuan	Proses pelatihan dataset
Deskripsi	Untuk melatih dataset yang akan digunakan pada proses pencarian solusi menggunakan algoritma neural network
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain sudah memuat dataset permainan dan memasukan parameter neural network
Aksi aktor	Reaksi Sistem
memasukan nilai input1, input2 , input 3, input 4, input5 dan target output	
	mengambil data input koordinat awal pada area permainan
	normalisasi
	backpropagation
	membandingkan nilai error output dan target
	mendapatkan nilai output
Skenario alternatif	

	5a. Mendapatkan nilai output sementara
Kondisi Akhir	Sistem menyimpan data training

Tabel 3.15 Skenario Use Case Simpan Data Pelatihan

Identifikasi	
Nomor	U – 4
Nama	Simpan data pelatihan
Tujuan	Melakukan proses penyimpanan data trainin
Deskripsi	Proses untuk mendapatkan data training dari hasil permainan pengguna.
Aktor	User
Skenario Utama	
Kondisi awal	Permainan telah seleseai (menggabungkan 4a dan 4a, 4b dan 4b)
Aksi aktor	Reaksi Sistem
Memasukan nilai inpit1, input 2, input3, input 4, input5, dan target output	
	Menyimpan dataset permainan
	Melakukan reset aray

Kondisi Akhir	Sistem menyimpan dataset permainan

Tabel 3.16 Skenario Use Case Reset

Identifikasi	
Nomor	U – 5
Nama	Reset
Tujuan	Menempatkan posisi awal karakter
Deskripsi	Proses untuk menentukan posisi.
Aktor	User
Skenario Utama	
Kondisi awal	Pemain berada di dalam permainan
Aksi aktor	Reaksi Sistem
Memasukan nilia koordinat x,y baru	
	Sistem memasukan nilai posisi baru pada area permaian
	Sistem menampilkan karakter pada area permaianan
Kondisi Akhir	Pemain dapat memulai permainan untuk menggeserkan karakter

Tabel 3.17 Skenario Use Case Load

Identifikasi	
Nomor	U - 6
Nama	Load
Tujuan	Mengambil dataset pelatihan
Deskripsi	Proses melakukan load dataset yang

	akan digunakan.
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada dalam menu pelatihan
Aksi aktor	Reaksi Sistem
Pemain memasukan data yang akan di load	
	sistem melakukan reset array
	sistem memasukan data load
Kondisi akhir	sistem mengambil dataset yang dipilih oleh pemain

Tabel 3.18 Skenario Use Case Memilih Lokasi Penyimpanan

Identifikasi	
Nomor	U - 7
Nama	Memilih lokasi penyimpanan dataset
Tujuan	Untuk menempatkan dataset pada lokasi penyimpanan yang dipilih oleh pemain
Deskripsi	Proses melakukan pemilihan tempat penyimpanan dataset.
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada dalam menu pelatihan
Aksi aktor	Reaksi Sistem
Pemain memilih lokasi penyimpanan	

	sistem menyimpan lokasi penyimpanan.
Kondisi akhir	sistem menyimpan lokasi penyimpanan dataset yang dipilih oleh pemain

Tabel 3.19 Skenario Use Case Pemilihan Parameter Neural Network

Identifikasi	
Nomor	U - 8
Nama	Pemilihan parameter neural network
Tujuan	Untuk parameter masukan pada data neural network yang akan diproses
Deskripsi	Proses memasukan parameter neural network
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain sudah memuat dataset yang akan dilatih.
Aksi aktor	Reaksi Sistem
Memilih parameter learning rate , dan error	
	Memasukan paramter yang diinputkan pemain
	Sistem menyimpan data parameter neural network
Skenario alternatif	

	3a. Sistem membatalkan penyimpanan paramter
Kondisi akhir	Sistem melakukan proses *backpropagation* berdasarkan parameter yang telah dipilih.

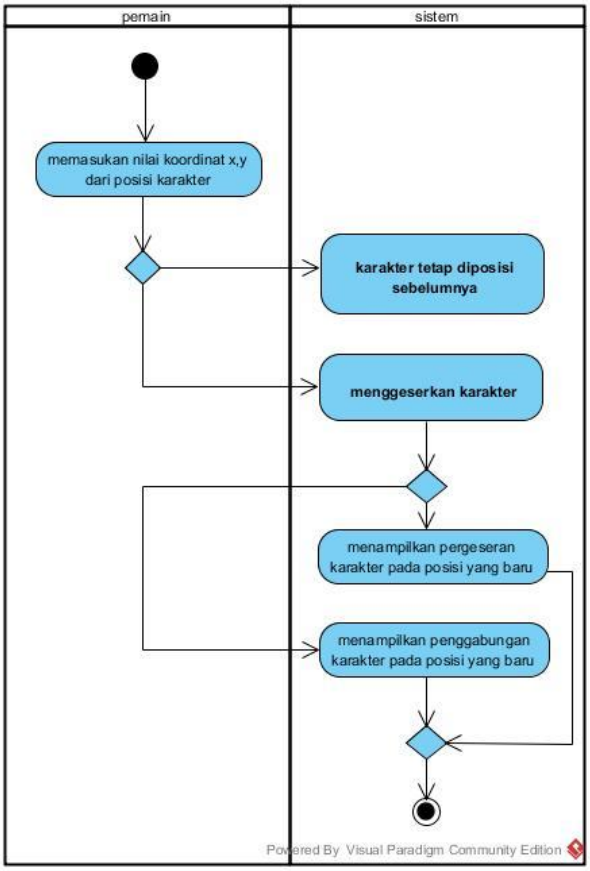
Tabel 3.20 Skenario Use Case Keluar

Identifikasi	
Nomor	U - 9
Nama	Keluar
Tujuan	Keluar dari permainan
Deskripsi	Proses keluar dari permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada dalam menu utama
Aksi aktor	Reaksi Sistem
Memilih keluar dari permainan	
	Permainan akan terhenti dan keluar dari sistem
Kondisi akhir	Pemain keluar dari permainan

3.2.8 Activity Diagram

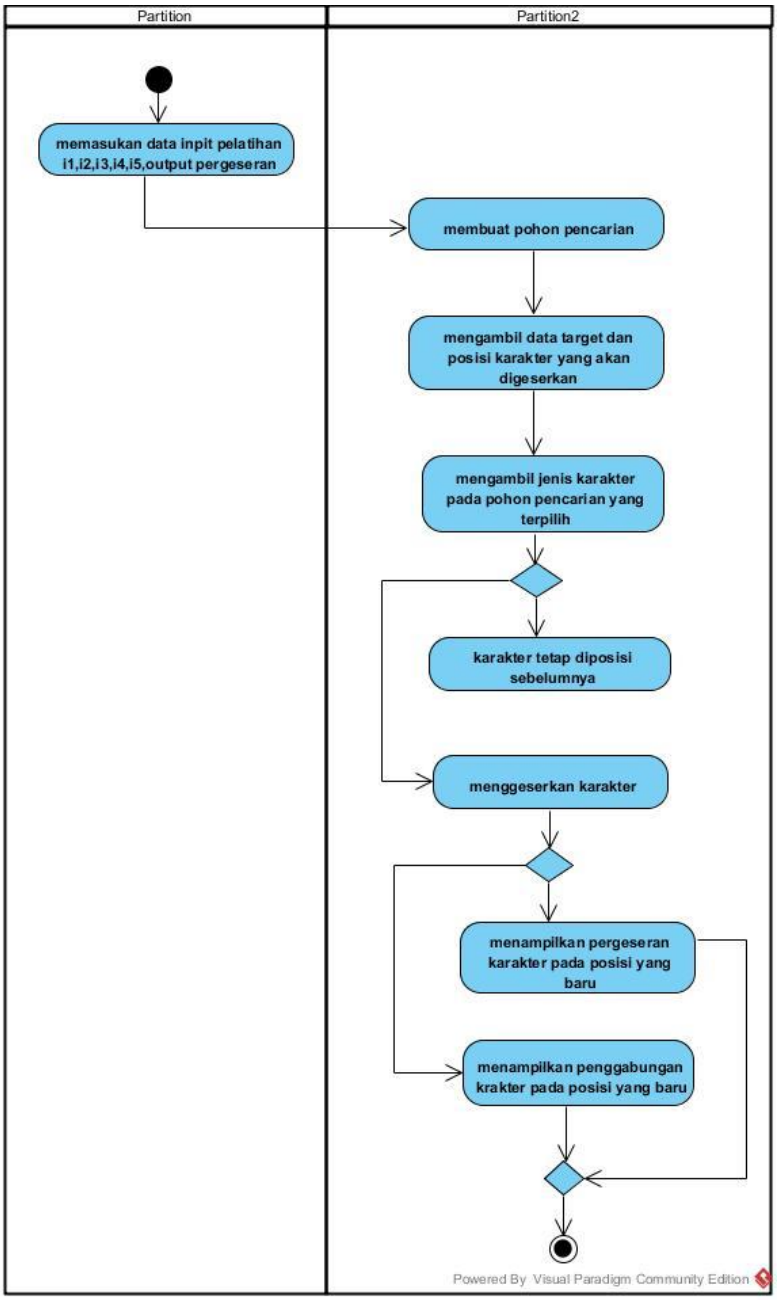
Activity diagram meneggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang. Berikut ini beberapa activity diagram yang terdapat pada permainan ini yaitu :

1. *Activity Diagram* Menentukan Jalur Penggabungan



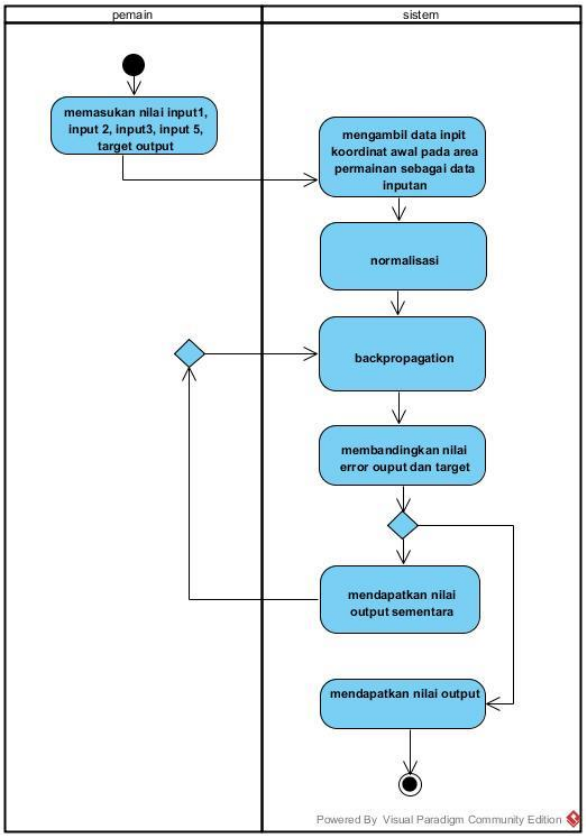
Gambar 3.7 *Activity Diagram* Menentukan jalur penggabungan

2. *Activity Diagram* Pencarian Solusi Menggunakan Algoritma Neural Network Backpropagation



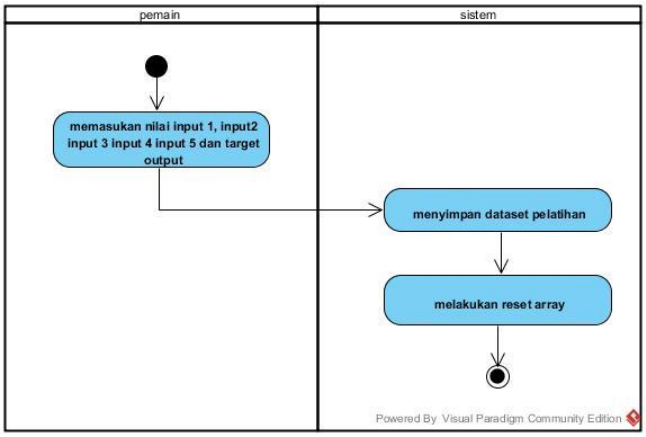
Gambar 3.8 *Activity Diagram* Pencarian Solusi Menggunakan Algoritma *Neural Network Backpropagation*

3. *Activity Diagram* Pelatihan



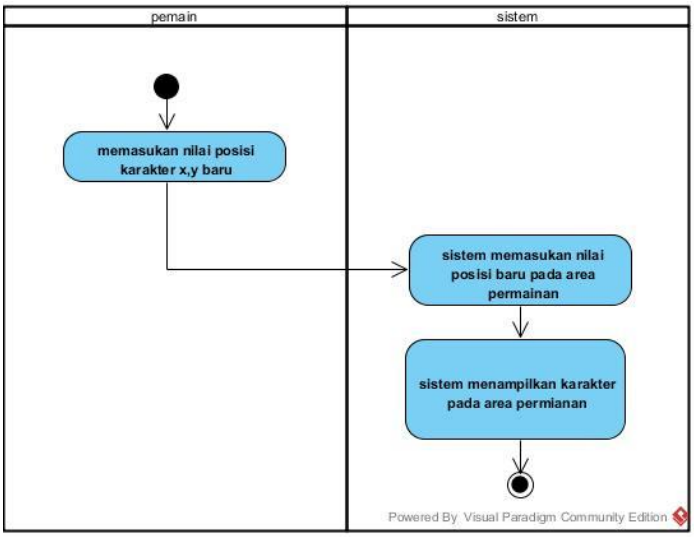
Gambar 3.9 *Activity Diagram* Pelatihan

4. *Activity Diagram* Simpan Data Pelatihan



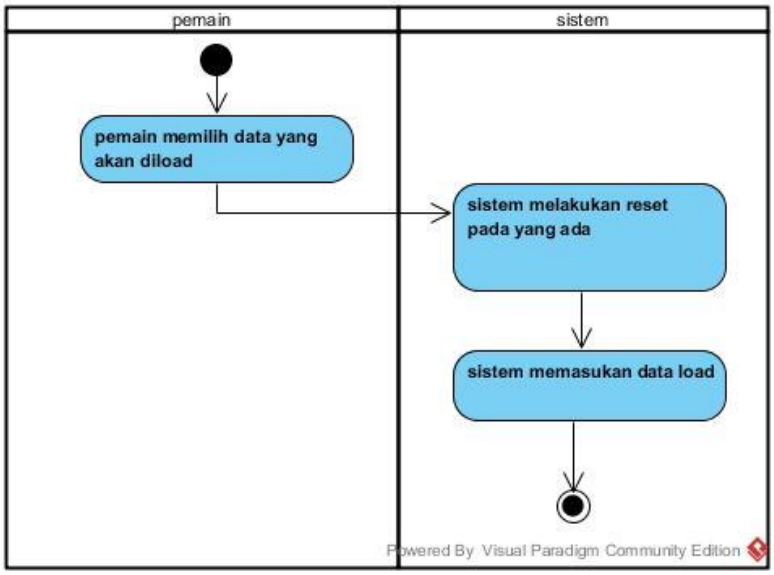
Gambar 3.10 *Activity Diagram* Simpan Data Pelatihan

5. *Activity Diagram Reset*



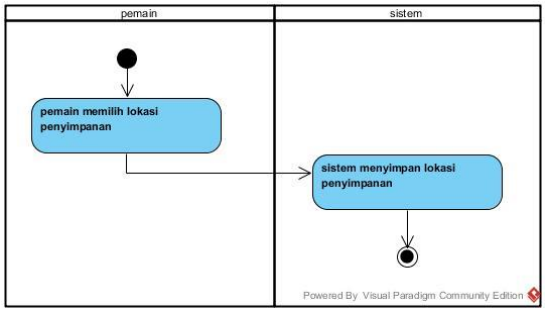
Gambar 3.11 ***Activity Diagram Reset***

7. Activity Diagram Load



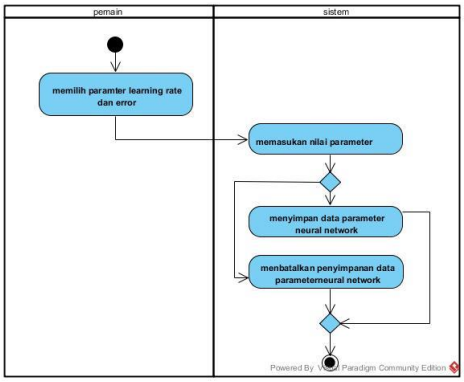
Gambar 3.12 ***Activity Diagram Load***

8. *Activity Diagram* Memilih Lokasi Penyimpanan



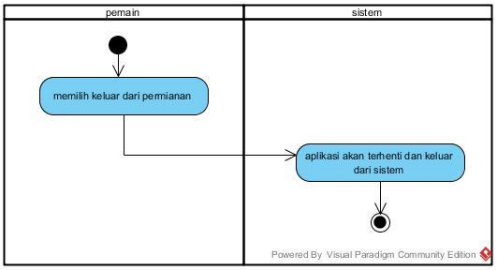
Gambar 3.13 *Activity Diagram* Memilih Lokasi Penyimpanan

9. *Activity Diagram* Pemilihan Parameter *Neural Network*



Gambar 3.14 *Activity Diagram* Pemilihan Parameter *Neutal Network*

10. *Activity Diagram* Keluar

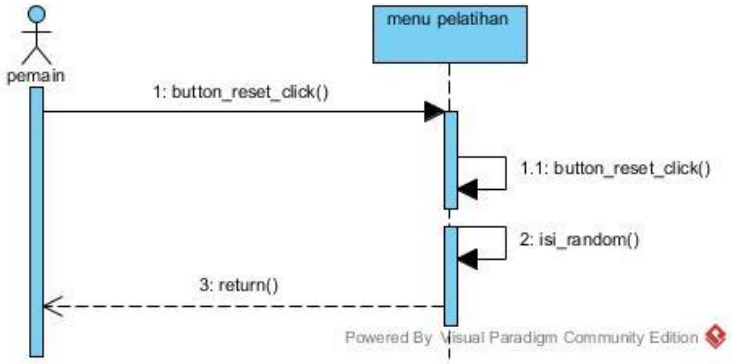


Gambar 3.15 Activity Diagram Keluar

3.2.9 Sequence Diagram

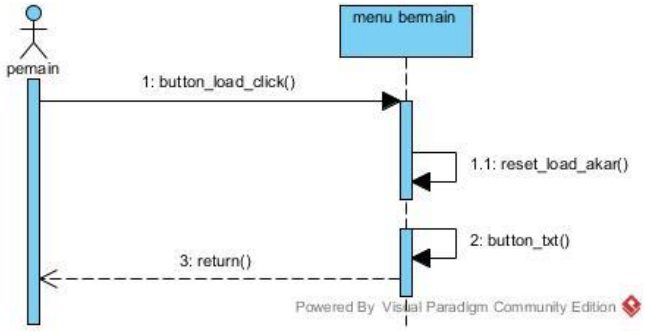
Sequence diagram menggambarkan perilaku sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *Use case*. Komponen utama *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress* vertikal. Berikut diagram *sequence* dari permainan ini :

1. Sequence *Diagram Reset*



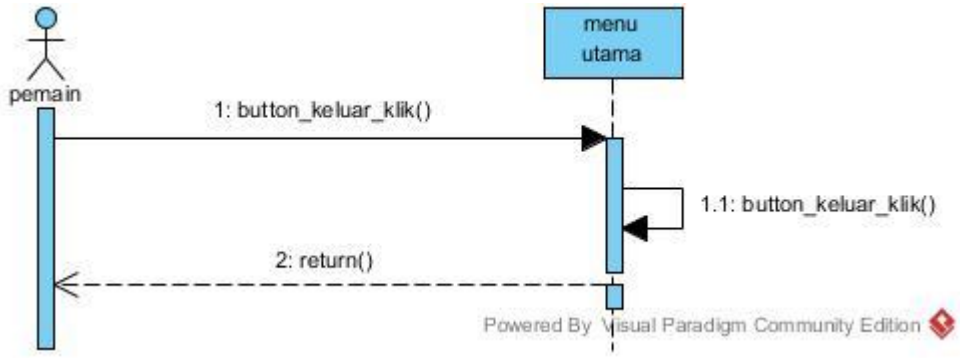
Gambar 3.16 *Sequence Diagram Reset*

2. Sequence *diagram load*



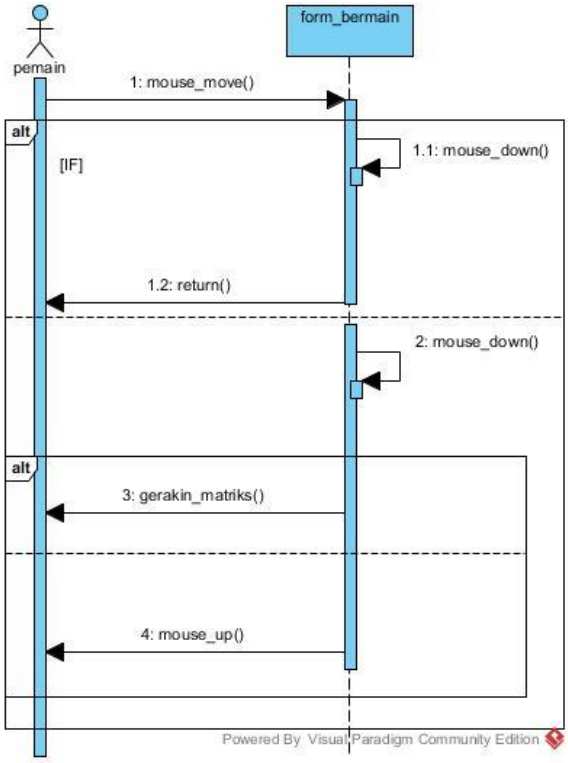
Gambar 3.17 *Sequence Diagram Load*

3. *Sequence Diagram* Keluar



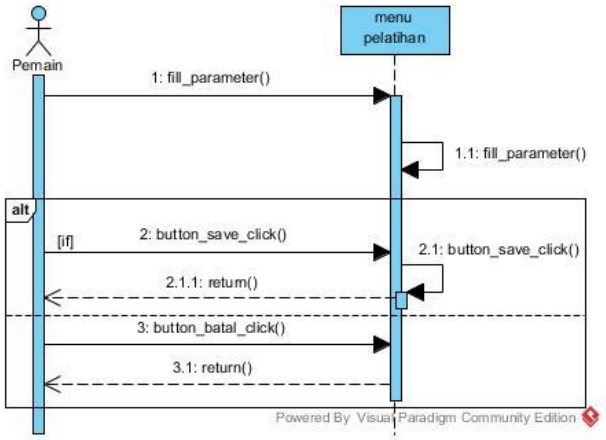
Gambar 3.18 *Sequence Diagram* Keluar

4. *Sequence Diagram* Menentukan Jalur Penggabungan



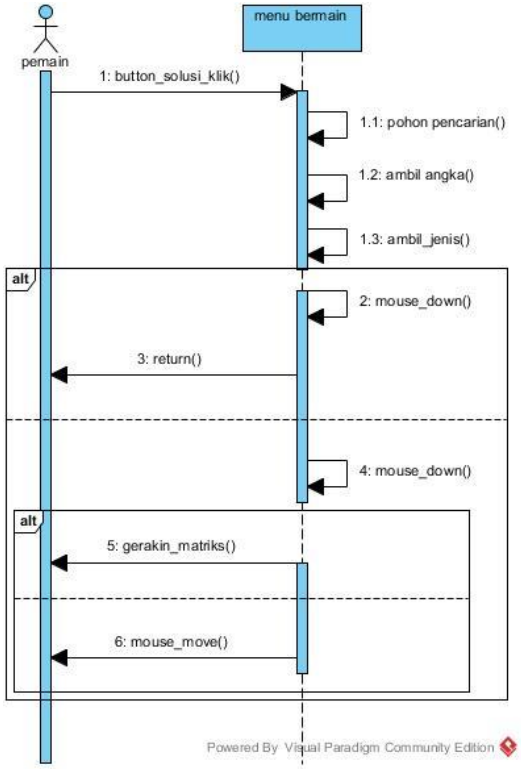
Gambar 3.19 *Sequence Diagram* Menentukan Jalur Penggabungan

5. *Sequence Diagram* Pemilihan Parameter *Neural Network*



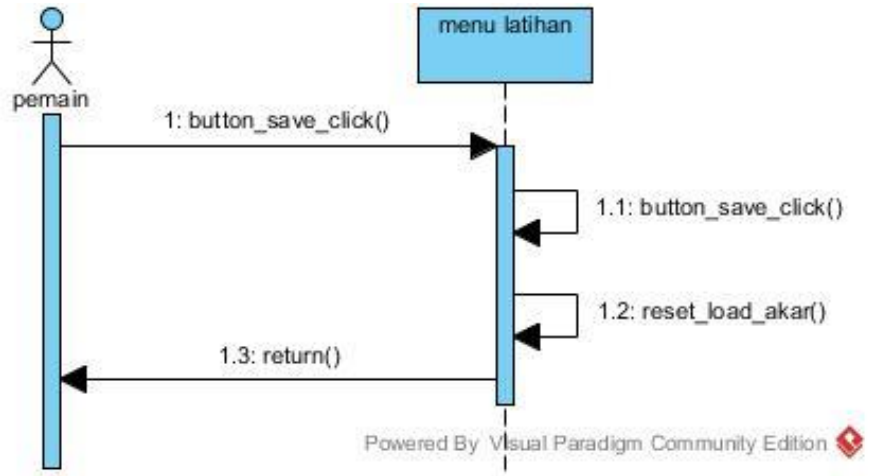
Gambar 3.20 *Sequence Diagram* Pemilihan Parameter *Neural Network*

6. *Sequence Diagram* Solusi Menggunakan Algoritna *Neural Network Backpropagation*



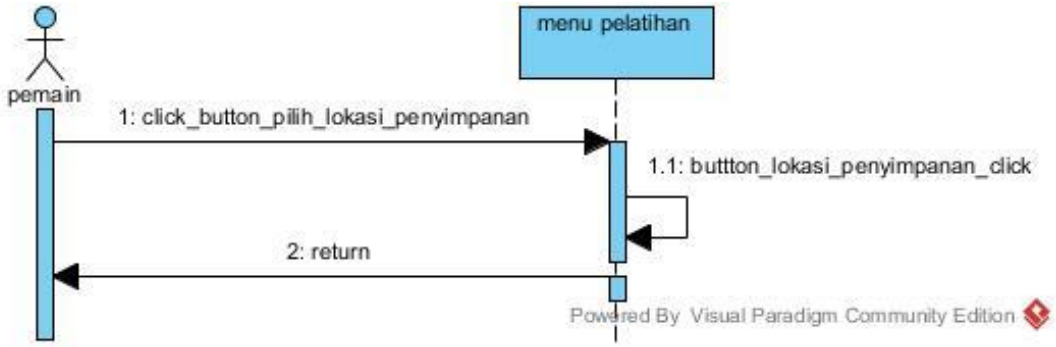
Gambar 3.21 *Sequence Diagram* Pemilihan Parameter *Neural Network*

7. *Sequence Diagram* Simpan Data Pelatihan



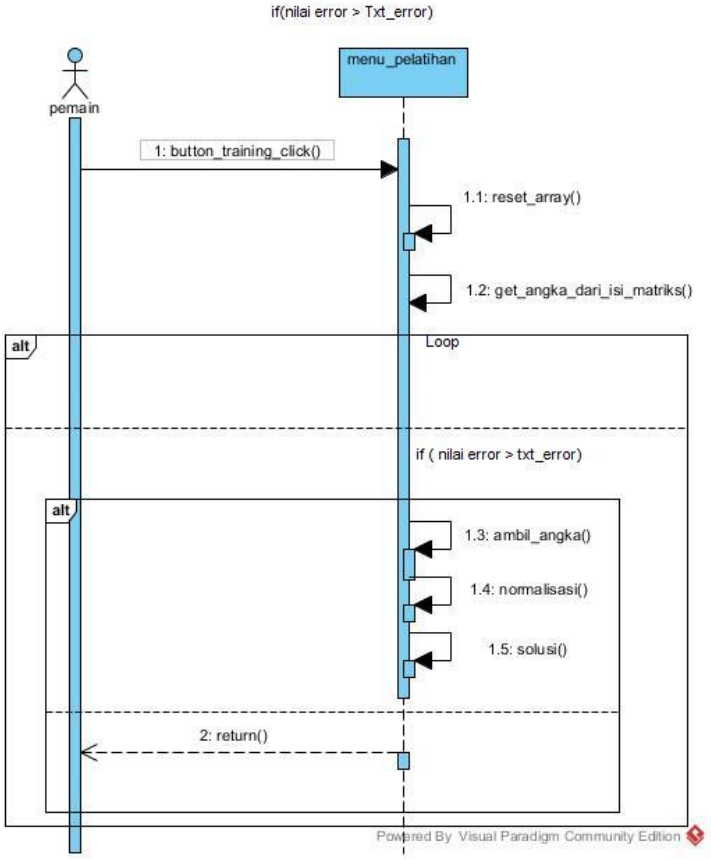
Gambar 3.22 *Sequence Diagram* Simpan Data Pelatihan

8. *Sequence Diagram* Pilih Lokasi Penyimpanan



Gambar 3.23 *Sequence Diagram* Pilih Lokasi Penyimpanan

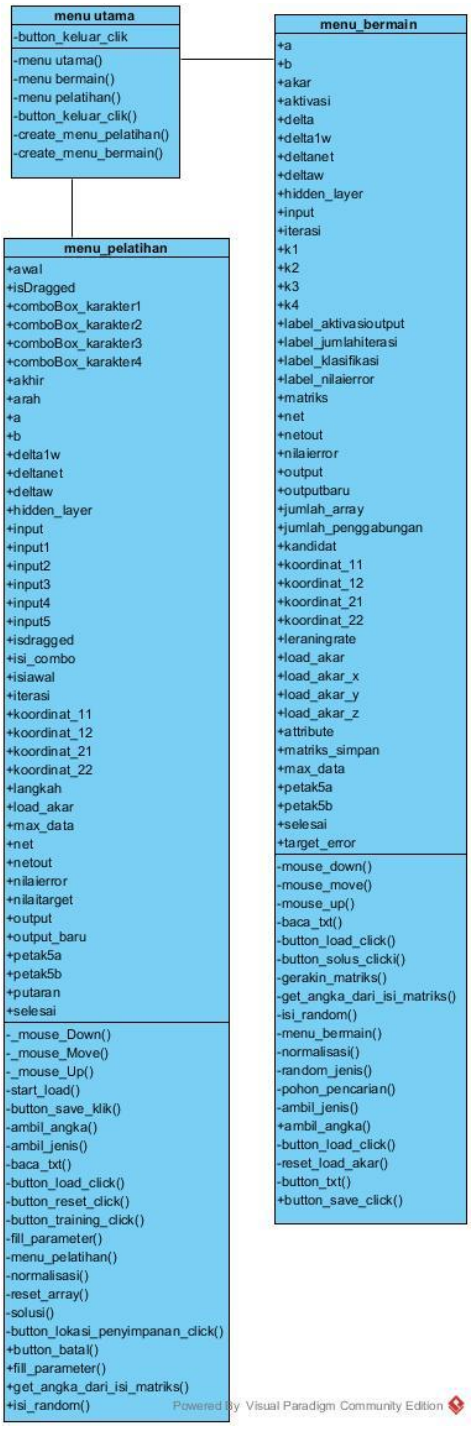
9. *Sequence Diagram* Pelatihan



Gambar 3.24 *Sequence Diagram* Pelatihan

3.2.10 *Class* Diagram

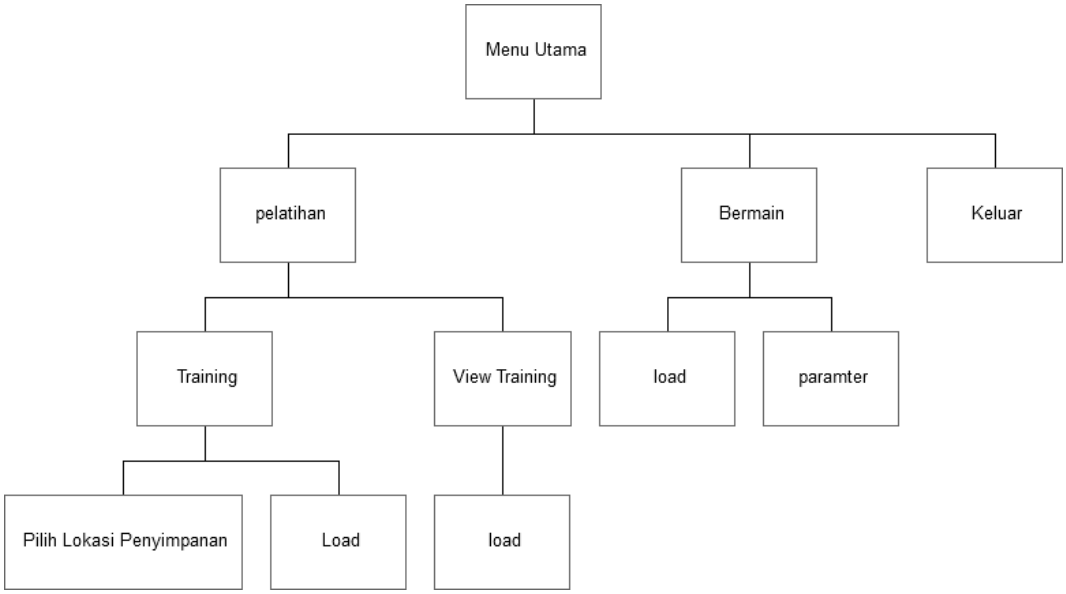
Class diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi *atribut* dan *method* yang ada pada masing-masing *class* seperti terlihat pada gambar :



Gambar 3.25 Class Diagram

3.2.11 Perancangan Sturuktur Menu

Perancangan struktur menu aplikasi solver game dazzle ini membantu dalam merancang bagian-bagian dari sistem yang sebenerya dan untuk mengetahui bagian mana yang terlebih dahulu yang akan diakses setelah program tersebut selesai

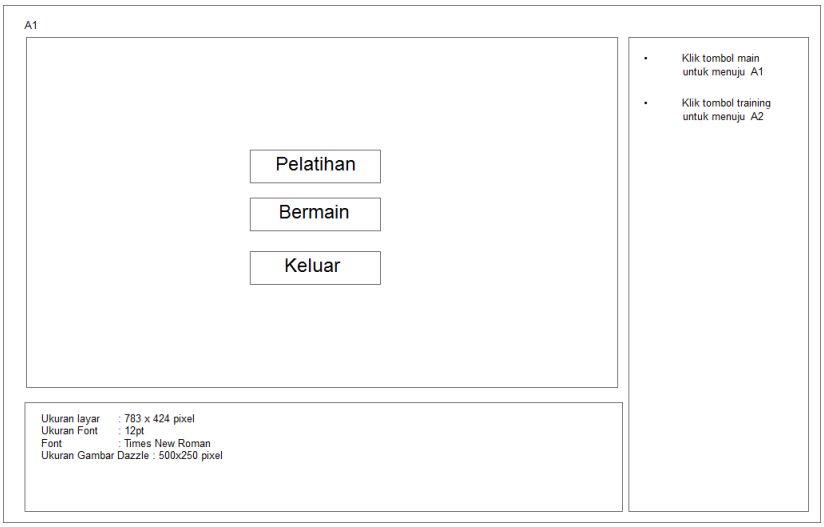


Gambar 3.26 Struktur Menu Aplikasi Solver Game Dazzle

3.2.12 Perancangan Antar muka

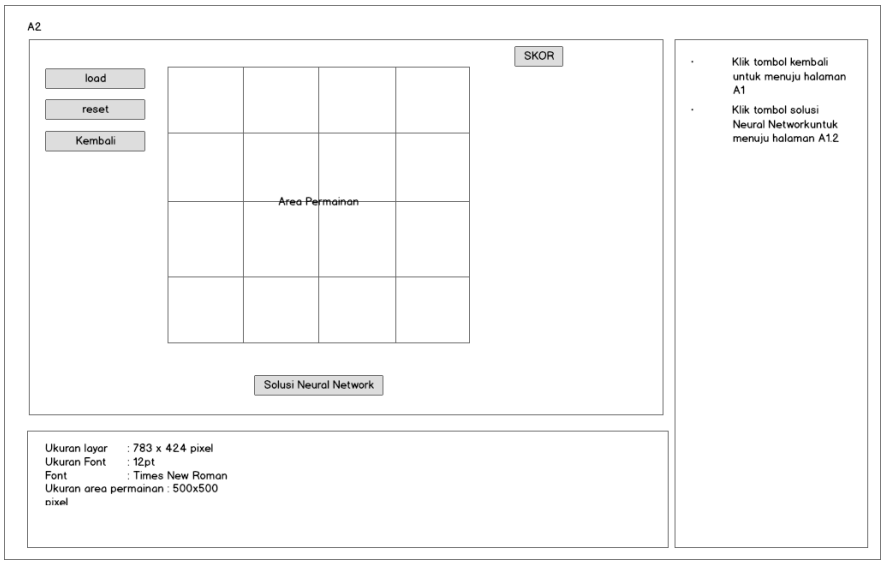
Perancangan antarmuka dilakukan untuk mempermudah dalam menginplementasikan sistem yang akan dibangun. Berikut adalah perancangan antarmuka aplikasi *solver game* dazzle.

1. Perancangan antarmuka menu utama.



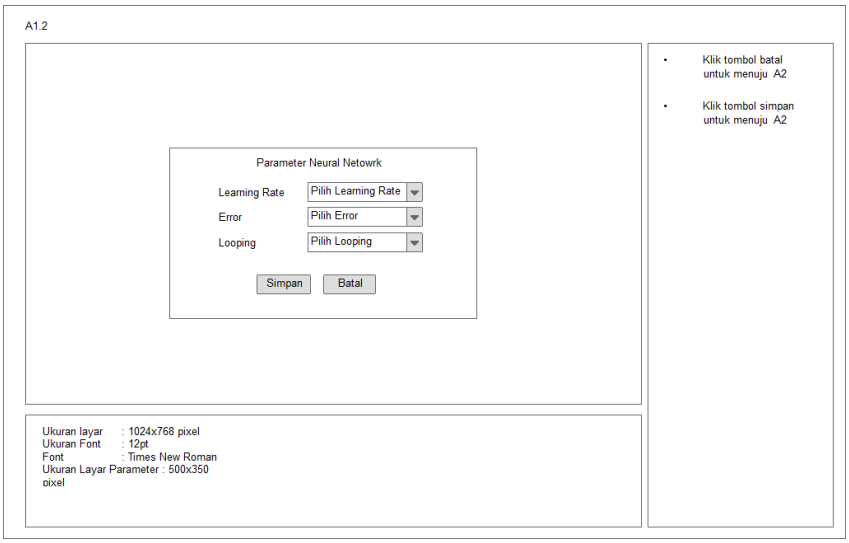
Gambar 3.27 Perancangan Menu Utama

2. Perancangan Menu Bermain



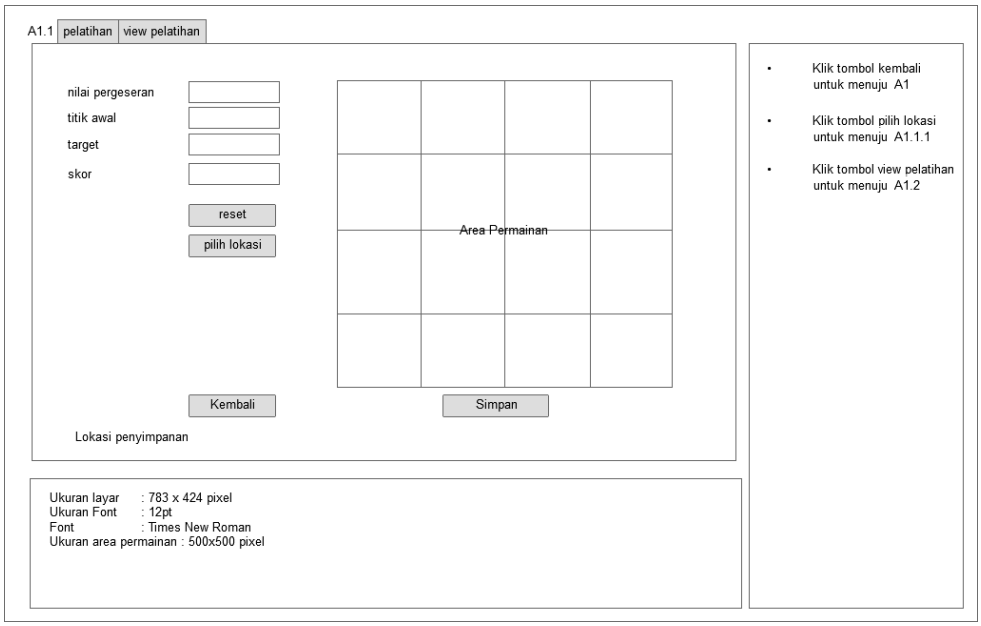
Gambar 3.28 Perancangan Menu Bermain

3. Perancangan Menu Parameter Neural Network



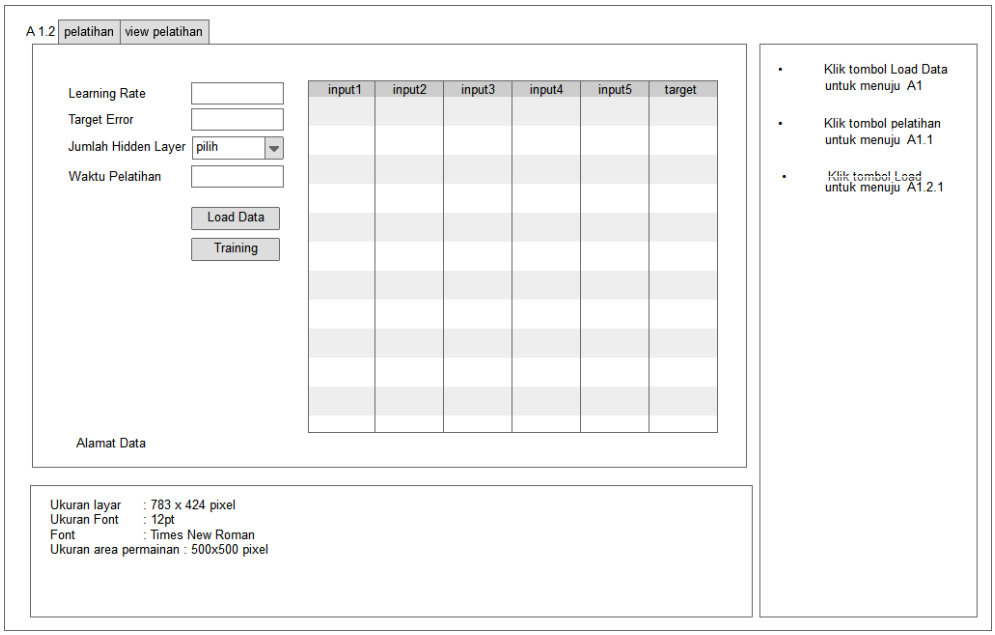
Gambar 3.29 Perancangan Menu Parameter Neural Network

4. Perancangan Menu Pelatihan



Gambar 3.30 Perancangan Menu Pelatihan

5. Perancangan Menu View Training

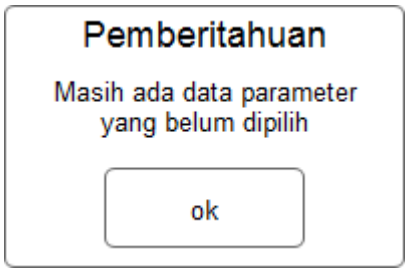


Gambar 3.31 Perancangan Menu View Traning

3.2.13 Perancangan Pesan

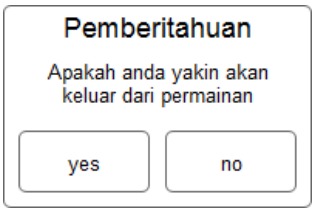
Perancangan pesan adalah rancangan antarmuka dari pesan yang akan ditampilkan apabila terjadi kesalahan dan pemberitahuan dalam proses yang akan dilakukan. Berikut adalah perancangan pesan yang dibuat.

1. Perancangan Pesan P1



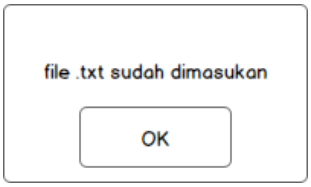
Gambar 3.32 Perancangan Pesan Pemberitahuan Parameter Neural Netwok Belum Dipilih

2. Perancangan Pesan P2



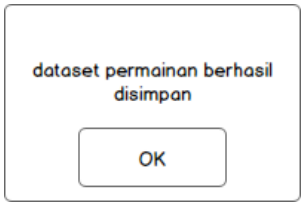
Gambar 3.33 Perancangan Pesan Pemberitahuan Jika Menekan Tombol Keluar

3. Perancangan Pesan P3



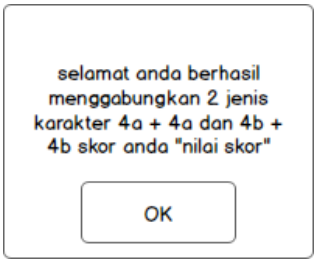
Gambar 3.34 Perancangan Pesan Pemberitahuan Data Berhasil Diload

4. Perancangan Pesan P4



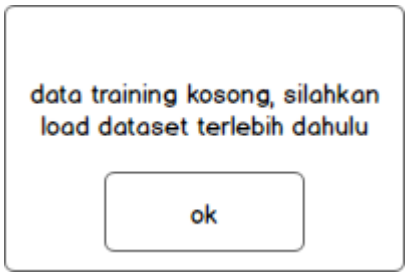
Gambar 3.35 Perancangan Pesan Pemberitahuan Dataset Permainan Berhasil Disimpan

5. Perancangan Pesan P5



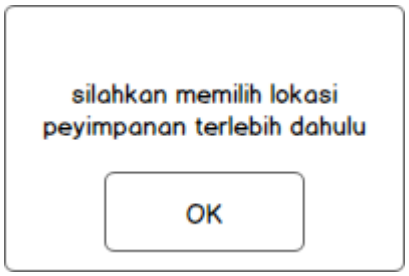
Gambar 3.36 Perancangan Pesan Pemberitahuan Memenangkan Permaninan

6. Perancangan Pesan P6



Gambar 3.37 Perancangan Pesan Pemberitahuan Dataset Training Kosong

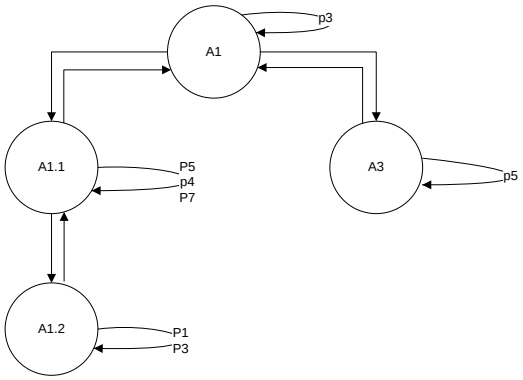
7. Perancangan Pesan P7



Gambar 3.38 Perancangan Pesan Pemberitahuan Belum Memilih Lokasi Penyimpanan dataset

3.2.14 Jaringan Semantik

Jaringan semantuk menggambarkan perancangan terhadap aliran menu-menu yang ada pada sistem. Berikut adalah jaringan semantik aplikasi *solver game* dazzle.



Gambar 3.39 Jaringan Semantik