

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

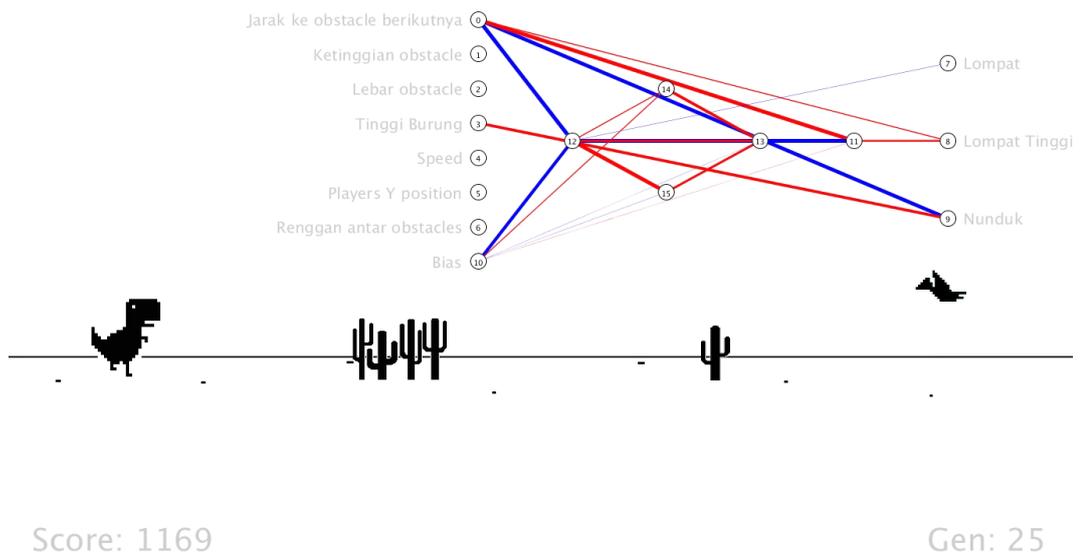
5.1 HASIL IMPLEMENTASI

Pada tahap ini penulis mengimplementasikan hasil rancangan yang telah dibuat pada tahap sebelumnya. Implementasi yang dimaksud adalah proses menterjemahkan rancangan menjadi sebuah *game* atau system yang dapat berjalan.

5.2 Tampilan *User Interface*

Berikut ini adalah tampilan sistem dengan tampilan gui dari *game Dino Google* yang dapat dilihat pada gambar 5.1.

1. *Dino*, sebagai karakter utama pada game (*T-Rex*).
2. *Obstacles*, merupakan rintangan yang harus di hindari *dino*, berupa kaktus dan butung (*pterodactyl*).
3. *Score*, untuk menampilkan *point* yang diperoleh saat menghindari rintangan,
4. *Gen*, untuk menampilkan generasi *Genetic Algorithm* yang sedang berjalan
5. *Neural Network Interface*, untuk menampilkan pola pembentukan jaringan saraf pada *neural network*.



Gambar 5.1 Tampilan Game *User Interface*

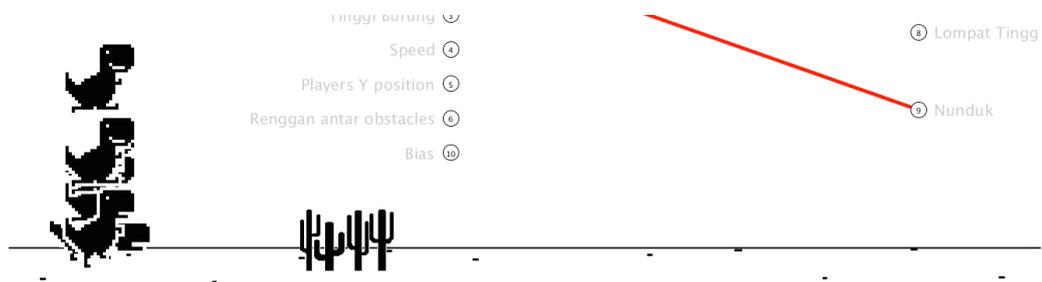
5.3 HASIL IMPLEMENTASI ALGORITMA

Pada tahap ini penulis mengimplementasikan hasil rancangan algoritma yang telah dibuat pada tahap sebelumnya. Implementasi yang dimaksud adalah proses algoritma yang terjadi pada *game*. Terdapat 2 jenis algoritma yang diterapkan pada *Game* yaitu *Genetic Algorithm* dan *Neural Network*.

5.3.1 *Genetic Algorithm*

Genetic Algorithm yang digunakan dalam game ini adalah *classic Genetic algorithm*, yang dimana setiap generasi berisi 500 populasi *player*, setiap *player* akan diseleksi hingga hanya 1 *player* terbaik yang akan tersisah pada setiap generasi. *Palyer* terbaik disetiap generasi akan di *crossover* untuk membentuk *offspring* (keturunan) yang baru (*children*). Titik *crossover string* biner dari awal kromosom orangtua pertama hingga batas titik *crossover* disalin ke keturunannya,

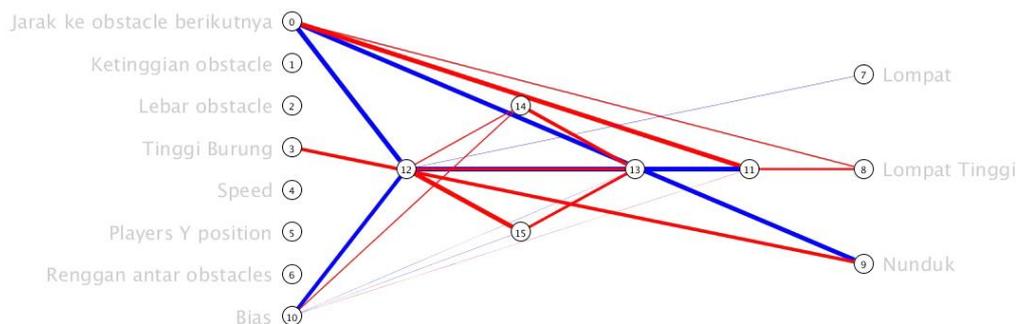
sisu biner keturunan disalin dari orangtua kedua, sehingga menghasilkan keturunan baru.



Gambar 5.2 Pergerakan *Player*

5.3.2 Neural Network

Artificial Neural Network (ANN) adalah model komputasi yang terinspirasi oleh otak manusia. Mereka terdiri dari sejumlah besar node yang terhubung, masing-masing melakukan operasi matematika sederhana. Output setiap node ditentukan oleh operasi ini, serta seperangkat parameter yang spesifik untuk node itu. Dengan menghubungkan node-node ini bersama-sama dan dengan hati-hati mengatur parameternya, fungsi yang sangat kompleks dapat dipelajari dan dihitung.



Gambar 5.3 Tampilan *Neural Network*

5.4 HASIL YANG DICAPAI

Pada bagian ini penulis menggambarkan hasil yang mungkin dicapai saat menjalankan program yang sebelumnya sudah dirancang, yang dapat dilihat pada beberapa generasi dibawah.

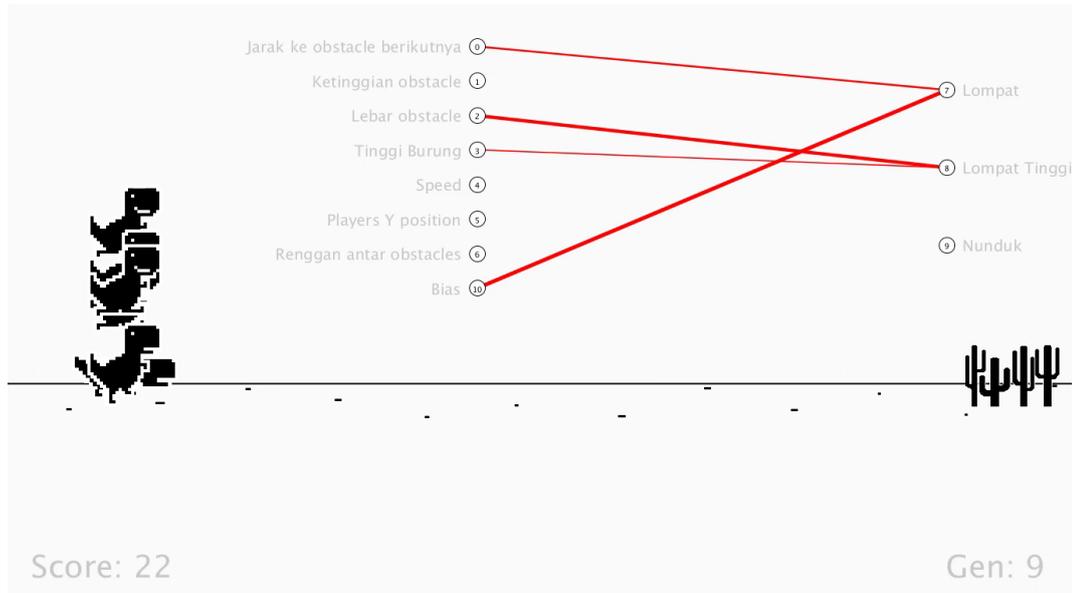
5.4.1 Gen 1



Gambar 5.4 Hasil Yang Dicapai Gen 1

Gen 1 merupakan generasi pertama dari proses evolusi, pada *gen* ini NN baru memiliki 1 *input* “bias” dan *output* “lompat tinggi”. Pada bagian ini terlihat seperti semua populasi *player* hanya menekan tombol lompat secara terus menerus.

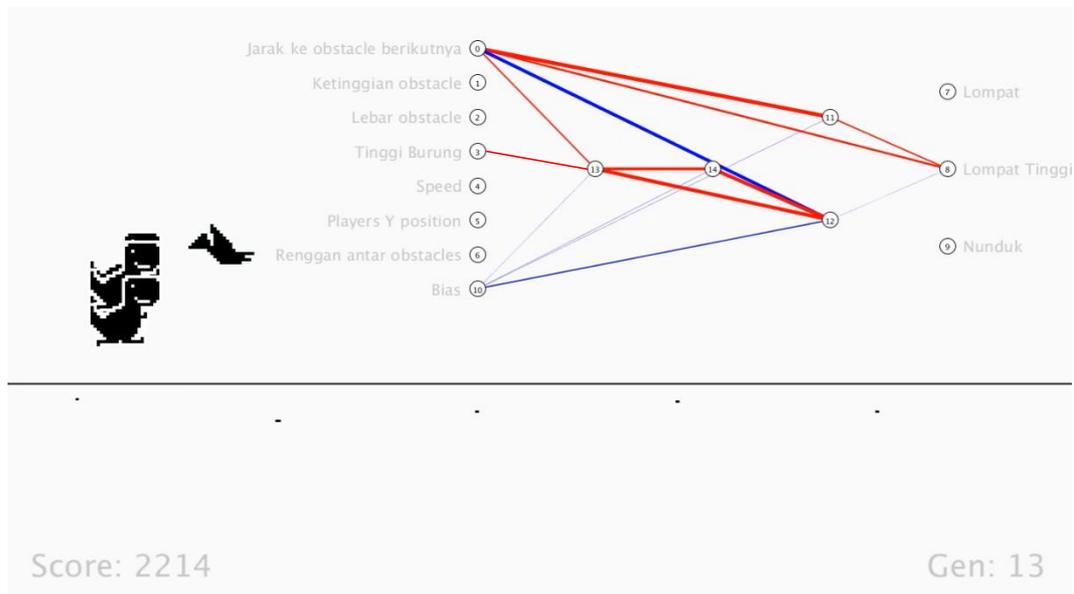
5.4.2 Gen 9



Gambar 5.5 Hasil Yang Dicapai Gen 9

Pada *gen 9* *AI* sudah mulai membentuk koneksi inputan lain untuk menghindari rintangan, dengan memberhitungkan “Jarak ke *obstacle* berikutnya”, “lebar *obstacle*”, “tinggi burung”, dan “bias”. Pada bagian ini *NN* belum memiliki *hidden layer*.

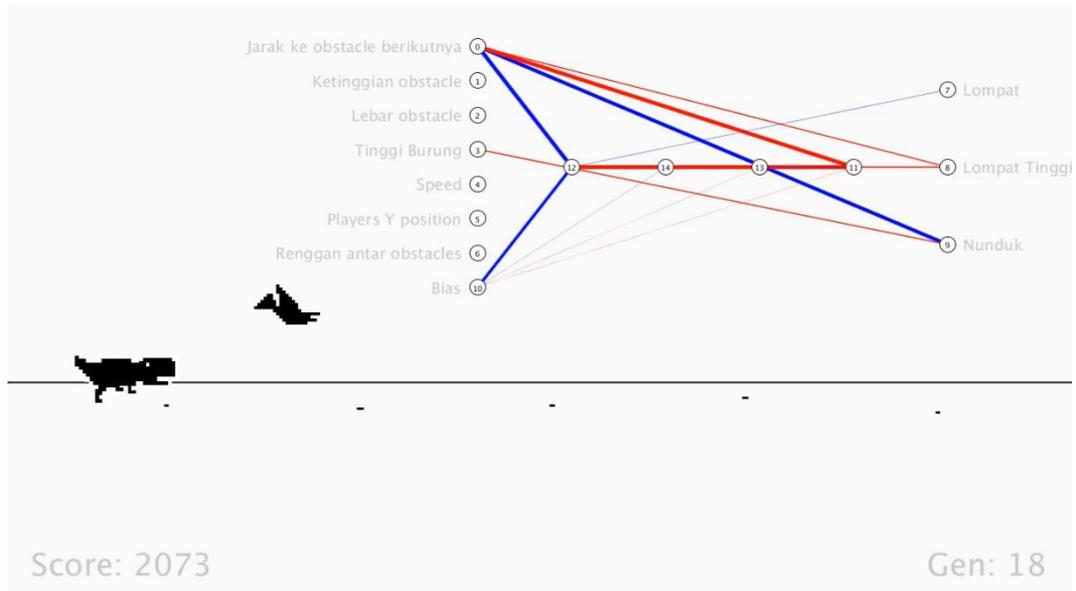
5.4.3 Gen 13



Gambar 5.6 Hasil Yang Dicapai Gen 13

Pada Gen 13 NN sudah mulai sudah memiliki *hidden layer*, dan mengurangi jumlah *input* yang sebelumnya 4 pada generasi ini hanya 3. Walaupun AI sudah memperhitungkan *input* “tinggi burung” tetapi AI belum memperhitungkan *output* yang tepat untuk menghindarinya dalam hal ini adalah “nunduk”. Karnanya *player* bukannya menghindarinya dengan cara menunduk tetapi malah mencoba untuk melompatinya dan menyebabkan semua player mati.

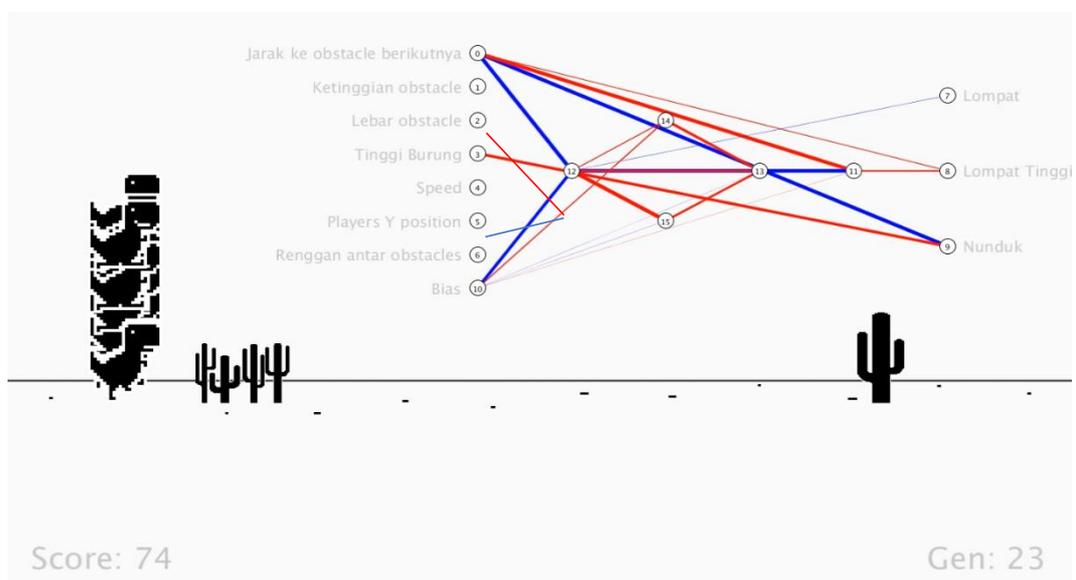
5.4.4 Gen 18



Gambar 5.7 Hasil Yang Dicapai Gen 18

Pada gen 18 AI barulah mempelajari cara untuk menghindari rintangan dengan cara menunduk.

5.4.5 Gen 23



Gambar 5.8 Hasil Yang Dicapai Gen 23

Pada gen 23 *Connection Neural Network* sudah mulai hampir terbentuk dengan sempurna, dan *player* bisa menghindari rintangan dengan sangat mudah. Akan tetapi prosesnya belum selesai sampai disini dikarenakan sifat evolusi pada *genetic algorithm* yang tidak terbatas dan bersifat *random* jadi apabila *program* dijalankan untuk yang kedua kalinya gambaran dari hasil yang dicapai pada halaman sebelumnya belum tentu akan sama.

5.5 PENGUJIAN SISTEM

Pada tahap ini penulis melakukan pengujian menggunakan metode pengujian *black box* dimana penulis melakukan pengecekan hasil keluaran dari setiap *input* dari *functional GUI* dan apabila hasil keluar tidak sesuai atau terjadi kesalahan maka penulis melakukan perbaikan agar hasil keluar dari aplikasi sesuai dengan hasil yang diharapkan. Pengujian memastikan bahawa keluaran yang diharapkan sesuai dengan hasil yang didapat dengan masukan masing-masing pengujian. Untuk lebih jelasnya dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian Sistem

Modul yang diuji	Prosedur pengujian	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Memulai <i>game</i>	Menekan <i>icon</i> aplikasi	Menampilkan gui dan secara otomatis menjalankan algoritma	Menampilkan gui dan algoritma berjalan dengan lancar	Berhasil
Menampilkan <i>player</i> terbaik pada setiap generasi	Menekan tombol "G"	Menampilkan <i>player</i> terbaik pada setiap generasi yang dimulai dari generasi pertama sampai	<i>Player</i> terbaik disetiap generasi ditampilkan	Berhasil

		generasi yang sedan berjalan		
Beralih ke generasi berikutnya saat menampilkan <i>player</i> terbaik pada setiap generasi	Menekat <i>arrow key</i> “ <i>right</i> ”	Saat menampilkan <i>player</i> terbaik pada setiap generasi yang dimulai dari generasi petrama dan menekan <i>arrow key</i> “ <i>right</i> ”, maka akan lanjut ke generasi selanjutnya tanpa harus menunggu <i>player</i> digenerasi tersebut mati	Saat menampilkan <i>player</i> terbaik pada setiap generasi dan menekan <i>arrow key</i> “ <i>right</i> ”, lanjut ke generasi selanjutnya	Berhasil

5.6 PENGUJIAN ALGORITMA *NEURAL NETWORK*

Pada tahap ini penulis juga menggunakan metode pengujian *black box* dalam melakukan pengujian dimana penulis melakukan pengecekan hasil keluaran dari setiap input dari functional GUI, hanya saja pengujian yang dilakukan berfokuskan terhadap *Neural Network* yang berlaku dalam *game*. Apabila hasil keluar tidak sesuai atau terjadi kesalahan maka penulis melakukan perbaikan agar hasil keluar dari aplikasi sesuai dengan hasil yang diharapkan. Pengujian memastikan bahawa keluaran yang diharapkan sesuai dengan hasil yang didapat dengan masukan masing-masing pengujian. Untuk lebih jelasnya dapat dilihat pada Tabel 5.2.

Tabel 5.2 Penhujian Algoritma

Modul yang diuji	Keluaran yang diharapkan	Kesimpulan
Jarak ke <i>obstacle</i> berikutnya	Mengukur jarak <i>player</i> dengan <i>obstacle</i> didepannya	Berhasil
Ketinggian <i>obstacle</i>	Mengukur tinggi <i>obstacle</i> rintangan yang akan dilewati <i>player</i>	Berhasil
Lebar <i>obstacle</i>	Mengukur lebar <i>obstacle</i> rintangan yang akan dilewati <i>player</i>	Berhasil
Tinggi Burung	Mengukur ketinggian burung yang harus dilewati <i>player</i>	Berhasil
Speed	Mengukur kecepatan <i>player</i> saat mendekati rintangan	Berhasil
Player Y position	Mengukur <i>player y position</i> (<i>player gravity</i>) sebelum melewati rintangan	Berhasil
Renggan Antar <i>obstacle</i>	Mengukur renggan <i>obstacle</i> rintangan yang akan dilewati <i>player</i>	Berhasil
Bias	Penambahan parameter awal untuk menyesuaikan <i>output</i> bersamaan dengan bobot <i>input</i> ke <i>neuron</i>	

5.7 KELEBIHAN DAN KEKURANGAN APLIKASI

Pada bagian ini, penulis akan menjelaskan tentang kelebihan dan kekurangan yang dimiliki *game dino google* ini yang sudah dilengkapi dengan *AI*. Dalam hasil dari perancangan aplikasi ini, penulis dapat menyampaikan kelebihan dan kekurangannya.

5.7.1 Kelebihan Aplikasi

Adapun kelebihan yang dimiliki *game* ini yaitu :

1. Aplikasi ini sangat mudah dioperasikan karna tampilan dirancang *user friendly*.

2. Rancangan aplikasi sangat sederhana sehingga tidak membuat bingung pengguna.
3. Aplikasi memiliki ukuran yang ringan dan bisa diakses tidak hanya di PC tapi juga bisa diakses pada perangkat *mobile (smartphone)*
4. Sistem memiliki fitur yang bisa menampilkan *player* terbaik pada generasi sebelumnya, sehingga *user* tidak harus selalu mengawasi pergerakan *player*.
5. Algoritma pada *game* bersifat *random*, sehingga apabila dimuat pada *device* dan waktu yang berbeda sangat kecil kemungkinannya hasil yang didapat sama. Dengan demikian diharapkan *game* ini dapat digunakan sebagai media pembelajaran.

5.7.2 kekurangan Aplikasi

Adapun kekurangan yang dimiliki *game* ini yaitu :

1. Hanya memiliki 1 *testing map*.
2. Tampilan GUI yang sangat standar.
3. Apabila pengguna tidak memahami konsep algoritma pada *game* ini, maka pengguna tersebut akan kesulitan memahami maksud dari adanya *game* ini.