PENERAPAN METODE *PERCEPTRON* UNTUK PEMILIHAN SERANGAN OLEH *NPC* DALAM *TURN-BASED RPG* GAME PEMBELAJARAN MALWARE

SKRIPSI

Oleh:
FEBRYAN CAHYA PERMANA
NIM. 10650067

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016

PENERAPAN METODE *PERCEPTRON* UNTUK PEMILIHAN SERANGAN OLEH *NPC* DALAM *TURN-BASED RPG* GAME PEMBELAJARAN MALWARE

SKRIPSI

Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
FEBRYAN CAHYA PERMANA
NIM.10650067

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2016

HALAMAN PERSETUJUAN

PENERAPAN METODE *PERCEPTRON* UNTUK PEMILIHAN SERANGAN OLEH *NPC* DALAM *TURN-BASED RPG* GAME PEMBELAJARAN MALWARE

SKRIPSI

Oleh :

Nama : Febryan Cahya Permana
NIM : 10650067
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji:

Dosen Pembimbing 1	**Dosen Pembimbing 2**
Dr.M.Faisal,MT.	**Fresy Nugroho, M.T**
NIP. 19740510 200501 1 007	**NIP. 19710722 201101 1 001**

Tanggal, Juni 2016

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

PENERAPAN METODE *PERCEPTRON* UNTUK PEMILIHAN SERANGAN OLEH *NPC* DALAM *TURN-BASED RPG* GAME PEMBELAJARAN MALWARE

SKRIPSI

Oleh :

Febryan Cahya Permana
NIM.10650067

Telah Dipertahankan Didepan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal, Juni 2016

Susunan Dewan Penguji :		**Tanda Tangan**
1. Penguji Utama	**: Hani Nurhayati, MT**	**( )**
	NIP. 19780625 200801 2 006	
2. Ketua Penguji	**: Fachrul Kurniawan, M.MT**	**( )**
	NIP. 19771020 200901 1 001	
3. Sekretaris	**: Dr.Muhammad Faisal,MT.**	**( )**
	NIP. 19740510 200501 1 007	
4. Anggota Penguji	**: Fresy Nugroho, M.T**	**( )**
	NIP. 19710722 201101 1 001	

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PERSEMBAHAN

Alhamdulillah tiada henti kuucapkan rasa syukur ini kepada Allah SWT yang mengabulkan doaku dalam menyelesaikan tugas penelitianku ini. Setelah lama berjuang akhirnya sampailah aku pada titik dimana aku dapat melihat dan menikmati dari hasil kerja kerasku dalam selama menempuh masa kuliah.

Aku persembahkan hasil karyaku ini kepada orang-orang penting dalam hidupku, atas dukungan dan doa merekalah aku dapat menemukan semangat dalam diri agar tidak menyerah serta yakin dan percaya pada diri sendiri.

Untuk ayahku Ardi Panggayuh dan mamaku Endahyati Umiyarsih terima kasih telah percaya pada anakmu ini, terima kasih atas dukungan, doa serta kasih sayang kalian serta segala hal yang telah kalian persembahkan untukku. Aku tahu atas doa kalianlah perjuanganku semakin mudah dan berhasil hingga saat ini.

Teruntuk adikku Hendy Utomo Putra terima kasih kau telah mendukung kakakmu ini, terima kasih kau selalu ada untuk menyemangatiku serta selalu hadir menghiburku.

HALAMAN MOTTO

“Impian dan cita-cita adalah arah hidup, tanpanya kita hanyalah badan kosong yang menjalani hidup tanpa arah dan terbawa arus. Mimpilah setinggi langit dan gapailah.”

HALAMAN PERNYATAAN

ORISINALITAS PENELITIAN

Saya yang bertanda tangan dibawah ini :

Nama	: Febryan Cahya Permana
NIM	: 10650067
Fakultas / Jurusan	: Sains dan Teknologi / Teknik Informatika
Judul Penelitian	: Penerapan Metode Perceptron Untuk Pemilihan Serangan Oleh NPC Dalam Turn-Based RPG Game Pembelajaran Malware.

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain. Kecuali dengan mencantumkan sumber cuplikan pada Daftar Pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Juni 2016

Yang Membuat Pernyataan,

Febryan Cahya Permana

NIM.10650067

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT yang telah melimpahkan rahmat serta karuniaNya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul "Penerapan Metode *Perceptron* Untuk Pemilihan Serangan Oleh *Npc* Dalam *Turn-Based Rpg* Game Pembelajaran Malware" dengan baik. Shalawat serta salam semoga tercurah kepada Nabi Agung Muhammad SAW yang telah membimbing umatnya dari gelapnya kekufuran menuju cahaya Islam yang terang benderang. Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untukmenyelesaikan skripsi ini.

Maka dari itu dengan segenap kerendahan hati patutlah penulis ucapkan terima kasih kepada:

1. Prof. DR. H. Mudjia Rahardjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.

2. Dr. drh. Bayyinatul Muchtaromah, M.Si, selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

3. Dr. Cahyo Crysdian, Selaku Ketua Jurusan Teknik Informatika UIN Maulana Malik Ibrahim Malang.

4. Dr.M.Faisal,MT. selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan, memberi kemudahan dan melancarkan proses penyelesaian skripsi ini.

5. Fresy Nugroho, M.T dosen pembimbing II sekaligus dosen wali yang telah memberikan masukan, bimbingan dalam menyelesaikan skripsi ini.

6. Segenap sivitas akademika Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu yang diberikan sebagai bekal bagi penulis.

7. Bapak dan Ibu tercinta yang memberikan doa dan restunya kepada penulis dalam menuntut ilmu.

8. Maulidiwati sri wahyuningsih, atas bantuan, masukan, dukungan serta motivasi kepada penulis.

9. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran. Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna. Semoga apa yang menjadi kekurangan bias disempurnakan oleh peneliti selanjutnya. Apa yang menjadi harapan penulis, semoga karya ini bermanfaat bagi kita semua dan menambah pengetahuan kita semua. Amin

Malang, Juni 2016

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Permana, Febryan Cahya. 2016. PENERAPAN METODE *PERCEPTRON* UNTUK PEMILIHAN SERANGAN OLEH *NPC* DALAM *TURN-BASED RPG* GAME PEMBELAJARAN MALWARE. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Dr.M.Faisal,M.T. (II) Fresy Nugroho, M.T

Kata Kunci : *Game 3D, Classification, Perceptron, 2-stage*

Dewasa ini perkembangan teknologi terbilang sangatlah cepat. Salah satu teknologi yang berkembang dengan sangat cepat adalah smartphone. Smartphone memiliki fitur yang sangat beragam dan fitur tersebut sangat membantu masyarakat dalam menjalani kehidupan sehari-harinya. Namun dibalik manfaat dari smartphone yang sangat membantu terdapat ancaman yang membahayakan isi atau data dari smartphone dan bahkan smartphone itu sendiri. Bahaya itu adalah serangan malware. Tingginya tingkat penggunaan smartphone menarik perhatian orang-orang tidak bertanggung jawab untuk menyebarkan malware yang dapat mencuri dan bahkan merusak data pribadi dan informasi penting dalam smartphone.

Oleh sebab itu pengetahuan mengenai malware menjadi sangat penting untuk dipelajari. Game pembelajaran malware dibuat untuk menarik minat masyarakat agar mereka tertarik untuk belajar. Dengan memanfaatkan video game masyarakat akan merasa senang dan juga memperoleh pengetahuan akan malware dari memainkan game ini. Dalam game ini pemain akan belajar mengenai malware dengan cara bertarung melawan para malware. Dengan bertarung melawan malware pemain akan memahami serangan yang dilakukan oleh malware dan cara mengatasi atau mencegah malware tersebut. Metode perceptron diterapkan dalam game ini karena malware itu sendiri memiliki berbagai macam serangan. Dengan menerapkan metode perceptron pada NPC malware maka NPC malware akan dapat melancarkan serangan sesuai dengan kondisi yang ada.

ABSTRACT

Permana, Febryan Cahya. 2016. IMPLEMENTATION OF PERCEPTRON METHOD FOR ATTACK SELECTION BY NPC IN TURN-BASED RPG MALWARE LEARNING GAME. Thesis. Department of Informatics Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Preceptor : (I) Dr.M.Faisal,M.T. (II) Fresy Nugroho, M.T

Keywords : *3D games, Classification, Perceptron, 2-stage*

Nowadays development of technology can be said growing so fast. One of the fast develop technology is smartphone. Smartphone has many features and those are very helpful for people to do their activities. But behind all of those features there are threats that threatening files inside smartphone and even the smartphone itself. That threat is attack of malware. High amounts of smartphone usage rate make some irresponsible people spread their malware attack, that attack can steal or even worse destroy data or important information in smartphone.

Because of that reason, knowledge about malware become very important to learn. Malware learning game created to attract people to learn about malware. Use video game as learning media, people will feel happy and also get knowledge from playing this game. In this game player will learn about malware by way fight against malware. With fight against malware player will understand attack of malware and how to resolve and prevent that malware. Perceptron method implemented in this game because the malware itself has various attack. With implementing this method into malware NPC, the NPC can makes decision which attack that it will choose depend on situation.

ملخص

Febryan Cahya Permana. 2016. تنفيذ المستقبلات طريقة اختيار هجوم من قبل مجلس الشعب في بدورها القائم **RPG** البرمجيات الخبيثة تعلم لعبة. أطروحة. قسم المعلوماتية كلية العلوم والتكنولوجيا في جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانج.

مؤدب: (I) Dr.M.Faisal,M.T. (II) Fresy Nugroho, M.T

كلمات البحث: الألعاب 3D، وتصنيف، المستقبلات، 2-مرحلة

في الوقت الحاضر تطوير تكنولوجيا يمكن أن يقال المتنامية بسرعة. واحدة من تطوير سريع التكنولوجيا هو الهاتف الذكي. الهاتف الذكي لديها العديد من الميزات وتلك هي مفيدة جدا للناس للقيام بأنشطتها. ولكن وراء كل تلك الميزات هناك التهديدات التي تهدد الملفات داخل الهاتف الذكي والهاتف الذكي حتى نفسه. هذا التهديد هو هجوم من البرمجيات الخبيثة. على كميات عالية من معدل استخدام الهواتف الذكية تجعل بعض الناس غير مسؤول تنتشر هجومهم البرمجيات الخبيثة، هذا الهجوم يمكن سرقة أو أسوأ من ذلك تدمير البيانات أو المعلومات الهامة في الهاتف الذكي.

بسبب هذا السبب، والمعرفة حول البرامج الضارة تصبح مهمة جدا للتعلم. البرمجيات الخبيثة تعلم لعبة خلقت لجذب الناس لمعرفة المزيد عن البرامج الضارة. استخدام لعبة فيديو كما تعلم وسائل الإعلام، والناس يشعرون بالسعادة وأيضا الحصول على المعرفة من هذه اللعبة. في هذه المباراة لاعب سوف تتعلم حول البرامج الضارة عن طريق مكافحة البرمجيات الخبيثة. مع الحرب ضد لاعب البرمجيات الخبيثة سوف تفهم هجوم من البرامج الضارة وكيفية حل ومنع أن البرمجيات الخبيثة. طريقة المستقبلات تنفيذها في هذه اللعبة لأن البرمجيات الخبيثة نفسه لديه هجوم المختلفة. مع تنفيذ هذه الطريقة في البرمجيات الخبيثة مجلس الشعب، ويمكن للمجلس الوطنى يجعل القرار التي تهاجم أنه سيختار تعتمد على الوضع.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini perkembangan teknologi sangatlah pesat. Salah satu teknologi yang perkembangan sangat pesat adalah *smartphone*. *Smartphone* telah menjadi suatu hal yang tidak lagi asing bagi setiap orang. Hampir setiap aspek dalam kehidupan manusia dapat dikerjakan melalui *smartphone*. Mulai dari komunikasi, pendidikan, bisnis, kesehatan, dan berbagai aspek kehidupan lainnya. *Smartphone* juga telah merambah berbagai kalangan masyarakat. Mulai dari masyarakat kelas atas hingga masyarakat kelas menengah kebawah. Hasil survei terbaru dari *GFK*, sebuah *market research institute* terbesar di Jerman menunjukkan di kawasan Asia Tenggara penjualan *smartphone* 12 bulan terakhir di Singapura, Malaysia, Thailand, Indonesia, Filipina, Vietnam, dan Kamboja mencapai 120 juta unit dengan valuasi US$ 16,4 miliar(indotelko, 2014). Tingginya tingkat konsumsi masyarakat terhadap *smartphone* membuat mereka tidak dapat lepas dari *smartphone* mereka dan selalu bergantung pada *smartphone* mereka. Tidak sedikit pula masyarakat yang memanfaatkan *smartphone* mereka untuk menyimpan dokumen atau data yang bersifat sangat penting dan rahasia misalnya nomor rekening, data-data pribadi, nomor ktp, dan lain-lain. Hal ini tentu dapat menarik minat orang-orang yang tidak bertanggung jawab untuk mencuri atau merusak data-data tersebut.

Salah satu cara yang digunakan untuk mencuri data dari *smartphone* adalah melalui *malware*. Berdasarkan laporan yang dirilis oleh *Alcatel-Lucent,* sebuah perusahaan perlengkapan telekomunikasi global di Perancis, bahwa ancaman *security* pada perangkat seluler dan residensial serta serangan pada jaringan komunikasi meningkat di tahun 2014. Lebih jauh *Alcatel-Lucent* mengungkapkan bahwa diperkiran 16 juta perangkat *mobile* di seluruh dunia telah terinfeksi oleh *malware* yang digunakan oleh penjahat *cyber* untuk spionasi korporat dan personal, pencurian informasi, *denial of service attacks* pada bisnis dan pemerintahan, penipuan perbankan, serta periklanan(indotelko, 2015). Menurut survei *Juniper Networks Mobile,* sebuah perusahaan yang mengembangkan dan menjual produk *networking,* pada periode 12 bulan sampai Maret 2013, serangan *malware* ke perangkat *mobile* naik 614 persen, dengan 92 persennya menyasar *Android*. Survei ini dirasa sangat masuk akal karena 75 persen pasar smartphone di dunia dikuasai oleh *android*(republika, 2013).

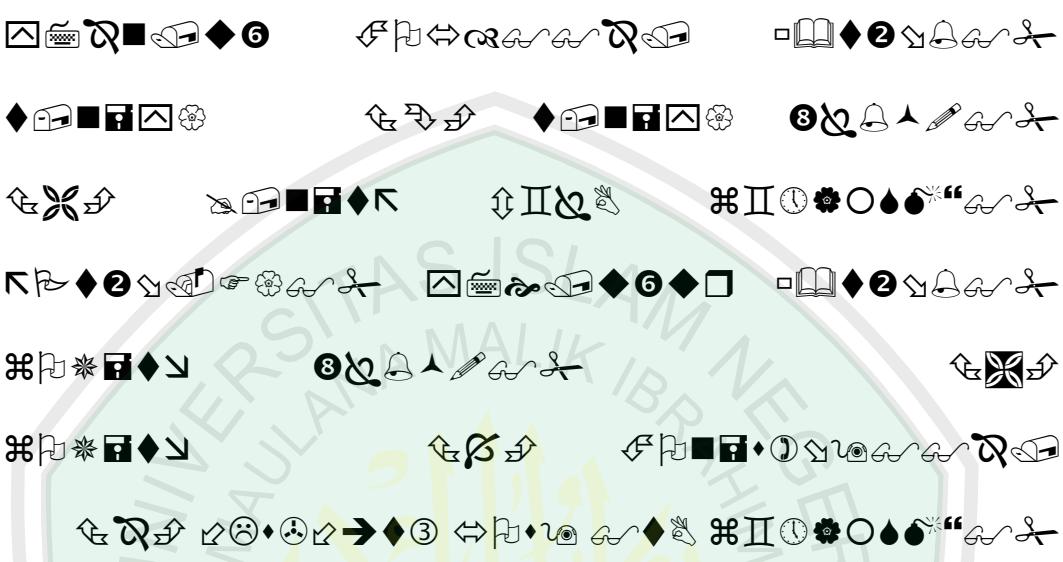
Di Indonesia sendiri menurut laporan *GFK, android* merupakan sistem operasi yang mendominasi dengan persentase mencapai 59,91 persen di pasar *smartphone* Indonesia(techinasia, 2015). Hal ini tentunya membuat masyarakat Indonesia rawan akan serangan *malware* terutama pengguna *smartphone android*. Perkembangan *malware* tersebut juga didukung oleh fakta bahwa rendahnya kesadaran masyarakat akan bahaya dari *malware,* serta kurangnya pengetahuan akan jenis-jenis dan bagaimana cara pencegahan dan penanganan dari *malware* tersebut. Dengan adanya fenomena ini penulis mencoba mempelajari bagaimana caranya untuk meningkatkan kesadaran masyarakat Indonesia akan bahaya dari *malware* serta memberikan pengetahuan akan *malware* tersebut.

Smartphone selain sebagai media komunikasi dapat pula dimanfaatkan sebagai sarana hiburan. Salah satu hiburan yang cukup digemari dan sering kali dilakukan di *smartphone* adalah bermain *video game*. Menurut survei yang dilakukan oleh *Trend Micro,* sebuah perusahaan *global security software* yang dilakukan untuk mengetahui perilaku pengguna *mobile game* di Indonesia, mengungkapkan bahwa 78 persen responden di Indonesia memainkan *mobile game* setidaknya sekali seminggu, dengan 48 persen di antaranya memainkan *mobile game* setiap hari(aktualita, 2015). Namun bermain *video game* sering kali dianggap tidak bermanfaat dan hanya untuk hiburan semata. Di lain sisi *video game* tidak hanya berfungsi sebagai hiburan semata, namun dapat dimanfaatkan sebagai media pembelajaran. Berdasarkan studi yang dilakukan oleh Vikranth Bejjanki dan rekannya dari *National Academy of Sciences,* mendapat kesimpulan bahwa bermain *video game* dapat meningkatkan kemampuan persepsi, perhatian, serta pemahaman. Ini disebabkan karena *video game* meningkatkan pembelajaran mengenai struktur dan aturan dari suatu lingkungan(science20, 2014).

Yang dimaksud di sini adalah ketika seseorang bermain *video game*, sering kali orang tersebut terbawa suasana dari *game* yang dia mainkan dikarenakan penggambaran dalam *video game* tersebut. Ketika sudah terlarut dalam suasana *video game* tersebut maka akan muncul rasa penasaran serta rasa ingin tahu, dan disitulah terjadi proses pembelajaran.

Berdasarkan penelitian tersebut penulis ingin memanfaatkan *video game* ini sebagai media dalam pembelajaran mengenai *malware* untuk *smartphone* berbasis *android*. *Game* ini dibuat agar dapat menarik minat serta memudahkan pengguna dalam mempelajari dan mengenali jenis-jenis *malware* yang ada. Hal ini sesuai

dengan apa yang telah diajarkan dalam Islam. Di dalam Al-quran dituliskan bahwa kita diharuskan untuk belajar, seperti yang tertulis dalam surat Al-alaq ayat 1-5. Berikut firman Allah SWT:



1. bacalah dengan (menyebut) nama Tuhanmu yang Menciptakan,

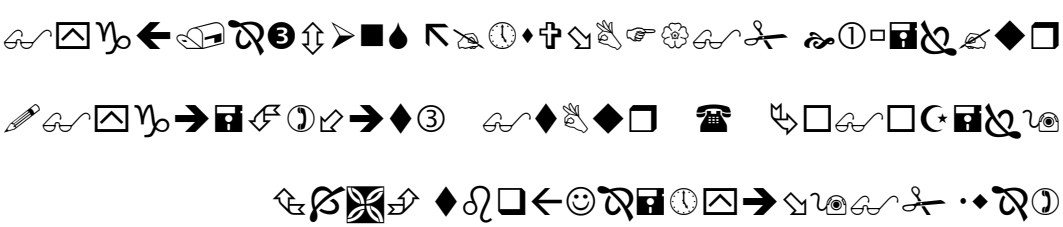
2. Dia telah menciptakan manusia dari segumpal darah.

3. Bacalah, dan Tuhanmulah yang Maha pemurah,

4. yang mengajar (manusia) dengan perantaran kalam,

5. Dia mengajar kepada manusia apa yang tidak diketahuinya.

Perintah untuk belajar juga terdapat dalam surat Al-ankabut ayat 43. Berikut firman Allah SWT:



43. dan perumpamaan-perumpamaan ini Kami buat untuk manusia; dan tiada yang memahaminya kecuali orang-orang yang berilmu.

Game ini akan dibuat dengan *genre turn-based RPG. Player* akan bermain sebagai *superhero* yang bertarung melawan *malware* jahat. Masing-masing *malware* akan memiliki tampilan dan serangan yang berbeda-beda sesuai dengan jenis *malwarenya*. Malware sendiri memiliki berbagai macam serangan, mulai dari meng-*hidden file*, mengubah *file* menjadi *shortcut*, sampai mencuri informasi berharga. Serangan-serangan ini akan dijadikan materi pembelajaran dalam bentuk serangan-serangan yang dilancarkan *malware* kepada *player* dalam *game*. Hal ini bertujuan agar *player* atau pengguna dapat mempelajari dan mengenal *malware* tersebut dari serangan yang dilakukan. Serangan yang dilancarkan oleh *malware* kepada *player* akan berbeda-beda sesuai dengan keadaan yang ada. *Malware* akan memilih salah satu dari berbagai serangan yang dimiliki sesuai dengan keadan. Namun untuk dapat memilih serangan yang sesuai di saat yg tepat *malware* harus mengetahui waktu atau pola yang benar dalam melancarkan suatu serangan. Berdasarkan permasalahan tersebut maka metode *perceptron* diterapkan ke dalam *game* untuk membantu *malware* dalam pengenalan pola, sehingga bisa dihasilkan serangan yang sesuai dengan keadaan yang ada.

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka perumusan masalah yang dapat ditarik adalah, bagaimana menerapkan algoritma perceptron untuk membantu NPC dalam memilih serangan sesuai dengan keadaan?

1.3 Batasan Masalah

- Pembelajaran yang diberikan merupakan pembelajaran mengenai *malware*.
- *Game* ini berbasis *Android*.
- *Game* ini bergenre *turn-based RPG*.
- Penggunaan metode *perceptron* diterapkan pada pemilihan serangan yang akan dilancarkan oleh *NPC*.
- Jenis *malware* yang dipelajari terbatas yaitu virus komputer, trojan, worm, adware.

1.4 Tujuan Penelitian

Membuat media pembelajaran mengenai *malware* yang menarik dan mudah untuk dipahami. Serta menerapkan algoritma *perceptron* pada pemilihan serangan oleh *NPC*.

1.5 Manfaat Penelitian

Manfaat penilitian ini adalah memberikan pengetahuan mengenai *malware* serta membantu pengguna untuk memahami *malware*.

BAB II

KAJIAN PUSTAKA

2.1 Video Game

Video game adalah sarana hiburan yang biasa dimainkan melalui perangkat *mobile* maupun komputer. Dalam penerapannya *video game* tidak hanya dapat digunakan sebagai sarana hiburan saja, namun dapat juga digunakan sebagai media pembelajaran. *Video game* ini dapat menjadi sarana pembelajaran mengenai *malware* yang menarik serta mudah dipahami.

2.2 Malware

Malware yang merupakan sebuah perangkat lunak berbahaya *(malicious software)* saat ini semakin mudah menyebar dan menginfeksi komputer. Tanpa kita sadari sistem dan aplikasi komputer kita telah dirusak bahkan informasi pribadi milik kita pun bisa diketahui dan disalahgunakan oleh orang lain hanya karena aktifitas browsing yang kita lakukan. Pada dasarnya malware adalah program berbahaya dan tidak diinginkan yang dapat merusak sistem komputer, menghambat akses internet dan yang paling berbahaya yaitu mencuri informasi seperti password dan nomor kartu kredit kita.

Jenis-jenis malware ini diantaranya adalah *trojan, virus, worm, spyware, adware, rootkit* dan sebagainya.

- *Virus Komputer* adalah jenis malware yang menyerang file eksekusi (.exe) yang akan menyerang dan menggandakan diri ketika file exe yang terinfeksi di jalankan. Virus komputer menyebar dengan cara menyisipkan program dirinya pada program atau dokumen yang ada dalam komputer.

- *Worm* adalah sebuah program komputer yang dapat menggandakan dirinya secara sendiri dalam sistem komputer. Sebuah worm dapat menggandakan dirinya dengan memanfaatkan jaringan (LAN/WAN/Internet) tanpa perlu campur tangan dari user itu sendiri.Worm memanfaatkan celah keamanaan yang memang terbuka atau lebih dikenal dengan sebutan vulnerability.

- *Spyware* adalah program yang bertindak sebagai mata-mata untuk mengetahui kebiasaan pengguna komputer dan mengirimkan informasi tersebut ke pihak lain. Spyware biasanya digunakan oleh pihak pemasang iklan.

- *Adware* adalah iklan yang dimasukan secara tersembunyi oleh pembuat program, biasanya pada program yang bersifat freeware untuk tujuan promosi atau iklan.

- *Trojan* atau *trojan horse* adalah program yang diam-diam masuk ke komputer kita, kemudian memfasilitasi program lain seperti virus, sypware, adware. keylogger dan malware lainnya untuk masuk, merusak system, memungkinkan orang lain meremote komputer dan mencuri informasi seperti password atau nomor kartu kredit kita.

- *Keylogger* adalah sebuah program yang dapat memantau penekanan tombol pada keyboard, sehingga orang lain dapat mengetahui password dan informasi apapun yang kita ketik.

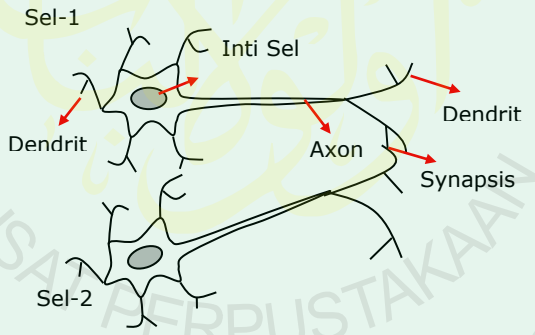
- *Rootkit* adalah program yang menyusup kedalam system komputer, bersembunyi dengan menyamar sebagai bagian dari system (misalnya menempel pada patch, keygen, dan crack), kemudian mengambil alih, memantau kerja sistem yang disusupinya. Rootkit dapat mencuri data yang lalu-lalang di jaringan, melakukan keylogging, mencuri cookies akun bank dan lain-lain.

- *Phishing* adalah suatu bentuk penipuan untuk memperoleh informasi pribadi seperti userID, password, ATM, kartu kredit dan sebagainya melalui e-mail atau website palsu yang tampak asli(catatanteknisi, 2011).

2.3 Jaringan Syaraf Tiruan

Jaringan saraf tiruan (JST) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan saraf manusia. JST dapat mengubah strukturnya untuk memecahkan masalah berdasarkan input yang mengalir melalui jaringan tersebut. JST adalah sebuah alat pemodelan data yang dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data.

Otak manusia berisi berjuta-juta sel syaraf yang bertugas untuk memproses informasi. Tiap-tiap sel bekerja seperti suatu prosesor sederhana. Masing-masing sel tersebut saling berinteraksi sehingga mendukung kemampuan kerja otak manusia.



Gambar 2. 1 Susunan syaraf manusia.

Gambar 2.1 menunjukkan susunan syaraf pada manusia. Setiap sel syaraf (neuron) akan memiliki satu inti sel, inti sel ini nanti yang akan bertukan untuk melakukan pemrosesan informasi. Informasi yang datang akan diterima oleh

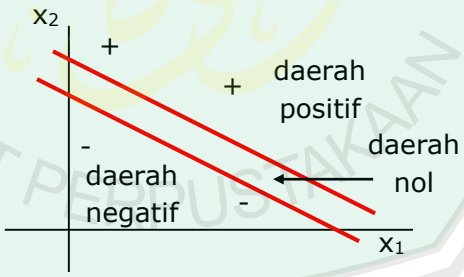
dendrit. Selain menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini akan menjadi masukan bagi neuron lain yang mana antar dendrit kedua sel tersebut dipertemukan dengan synapsis. Informasi yang dikirimkan antar neuron ini berupa rangsangan yang dilewatkan melalui dendrit. Informasi yang datang dan diterima oleh dendrit akan dijumlahkan dan dikirim melalui axon ke dendrit akhir yang bersentuhan dengan dendrit dari neuron yang lain. Informasi ini akan diterima oleh neuron lain jika memenuhi batasan tertentu, yang sering dikenal dengan nama nilai ambang (*threshold*). Pada kasus ini, neuron tersebut dikatakan teraktivasi. Hubungan antar neuron terjadi secara adaptif, artinya struktur hubungan tersebut terjadi secara dinamis. Otak manusia selalu memiliki kemampuan untuk belajar dengan melakukan adaptasi (Kusumadewi, 2003).

Jaringan syaraf tiruan juga memiliki cara kerja yang sama dengan syaraf manusia. Jaringan syaraf tiruan terdiri dari beberapa neuron, dan saling terhubung antara neuron yang satu dengan neuron yang lain. Input akan dikirim ke neuron dengan bobot tertentu. Input ini akan diproses oleh suatu fungsi yang mana fungsi ini akan menjumlahkan semua nilai bobot yang ada. Hasil penjumlahan akan dibandingkan dengan suatu *threshold* menggunakan fungsi aktivasi. Apabila input tersebut melebihi suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot outputnya ke semua neuron yang berhubungan dengannya.

2.4 Perceptron

Perceptron adalah salah satu bentuk jaringan syaraf tiruan yang sederhana. *Perceptron* biasanya digunakan untuk mengklasifikasikan suatu tipe pola tertentu yang sering dikenal dengan pemisahan secara linear. Jaringan ini hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.

Pada dasarnya, *perceptron* pada jaringan syaraf dengan satu lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang (*threshold*). Algoritma yang digunakan oleh aturan perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Nilai *threshold* (θ) pada fungsi aktivasi adalah non negatif. Fungsi aktivasi ini dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif.



Gambar 2. 2 Pembatasan linear dengan perceptron.

Garis pemisah antara daerah positif dan daerah nol memiliki pertidaksamaan:

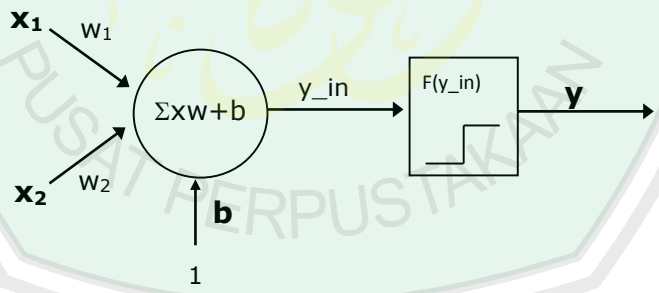
$$w_1x_1 + w_2x_2 + b > \theta$$

Sedangkan garis pemisah antara daerah negatif dengan daerah nol memiliki pertidaksamaan:

$$w_1x_1 + w_2x_2 + b < -\theta$$

Arsitektur jaringan perceptron mirip dengan arsitektur jaringan Hebb. Jaringan ini terdiri dari beberapa unit masukan ditambah sebuah bias, dan memiliki sebuah unit keluaran. Hanya saja fungsi aktivasi bukan merupakan fungsi biner (1, 0) atau bipolar (1, -1), tetapi memiliki kemungkinan nilai -1, 0 atau 1.

$$y = \begin{cases} 1, & \text{jika } y\_in > \theta \\ 0, & \text{jika } -\theta \le y\_in \le \theta \\ -1, & \text{jika } y\_in < \theta \end{cases}$$



Gambar 2. 3 Arsitektur jaringan.

Misalkan kita gunakan pasangan vektor input s dan vektor output sebagai pasangan yang akan dilatih.

Algoritma:

0. inisialisasi semua bobot dan bias: (Untuk sederhananya set semua bobot dan bobot bias sama dengan nol).

   Set *learning rate*: α ($0 < \alpha \leq 1$).

   (untuk sederhananya set sama dengan 1).

1. Selama kondisi berhenti bernilai *false*, lakukan langkah-langkah sebagai berikut:

   (i). Untuk setiap pasangan pembelajaran s-t, kerjakan:

   a. Set input dengan nilai sama dengan vektor input:

      $x_i = s_i$;

   b. Hitung respon untuk unit output:

      $$y\_in = b + \sum_i x_i w_i$$

      $$y = \begin{cases} 1, & \text{jika } y\_in > \theta \\ 0, & \text{jika } -\theta \leq y\_in \leq \theta \\ -1, & \text{jika } y\_in < \theta \end{cases}$$

   c. Perbaiki bobot dan bias jika terjadi *error*:

      Jika $y \neq t$ maka:

      $w_i(\text{baru}) = w_i(\text{lama}) + \alpha * t * x_i$

      $b(\text{baru}) = b(\text{lama}) + \alpha * t$

jika tidak, maka:

$w_i(baru) = w_i(lama)$

$b(baru) = b(lama)$

(ii). Tes kondisi berhenti: jika tidak terjadi perubahan bobot pada (i) maka kondisi berhenti TRUE, namun jika masih terjadi perubahan maka kondisi berhenti FALSE.

Algoritma tersebut bisa digunakan baik untuk input biner maupun bipolar, dengan θ tertentu, dan bias yang dapat diatur. Pada algoritma tersebut bobot-bobot yang diperbaiki hanyalah bobot-bobot yang berhubungan dengan input yang aktif ($x_i \neq 0$) dan bobot-bobot yang tidak menghasilkan nilai y yang benar (Kusumadewi, 2003).

2.5 Android

Android adalah sistem operasi berbasis *Linux* yang digunakan untuk telepon *mobile* seperti *smartphone* dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh bermacam piranti bergerak. *Android* kini telah menjelma menjadi sistem operasi *mobile* terpopuler di dunia(Yosef Murya, 2014).

Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *API* yang disediakan menawarkan akses ke *hardware*, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga(Stephanus Hermawan, 2011).

2.6 Role Playing Game (RPG)

Game dengan *genre* RPG adalah sebuah game di mana player memainkan suatu tokoh yang berada dalam game dalam dunia yang fiksi. Umumnya player akan bermain menjalankan sebuah *quest* atau misi untuk mengalahkan musuh atau monster. Fitur utama dari game dengan genre RPG adalah perkembangan dari tokoh yang dimainkan oleh player menjadi lebih kuat dengan cara mengumpulkan *exp point* dari musuh-musuh yang dikalahkan. Setiap kali *exp point* yang dimiliki sang tokoh telah mencapai nilai maksimal maka sang tokoh akan mengalami *level up*. Setiap kali *level up*, *attribute-attribute* yang dimiliki tokoh akan meningkat.

2.7 Turn-Based Strategy (TBS)

Turn-based strategy adalah *sub genre* dari game bergenre strategy. Dalam game ini dua player akan saling bertarung dan saling menunjukkan strategi untuk mempertahankan diri sendiri dan mengalahkan lawan. Player menjalankan aksinya secara bergantian dengan player lain. Ketika seorang player mendapatkan giliran untuk menjalankan aksinya maka player yang lain harus menunggu hingga player yang mendapatkan giliran menyelesaikan aksinya.

Dalam game bergenre TBS ini, tersedia unit dengan attribute yang berbeda-beda yang player dapat gunakan untuk mengalahkan player yang lain. Game ini menuntut seorang player untuk membuat strategy terbaik dan efektif dalam mengontrol unit yang ada.

2.8 Turn-Based Strategy Role Playing Game (TBSRPG)

Turn-based RPG atau Turn-Based Strategy Role Playing Game (TBSRPG) menggabungkan fitur dari TBS dengan fitur dari RPG game. *Gameplay* dari genre ini merupakan gabungan dari TBS dan RPG. Saat dalam pertarungan genre ini memiliki gameplay yang sama dengan TBS, yaitu bergantian dalam melaksanakan aksi. Saat diluar pertarungan genre ini memiliki gameplay yang sama dengan RPG.

2.9 Software Pendukung

Dibutuhkan beberapa aplikasi pendukung sebelum mengembangkan game berbasis android ini, antara lain:

- **Unity3D,** adalah sebuah game engine yang bersifat *multi-platform.* Unity dapat digunakan untuk mengembangkan game yang berbasis *mobile* seperti *android* dan *iPhone* serta yang berbasis *PC* ataupun *game console* seperti *PS3* dan *X-BOX*. Bahasa yang digunakan dalam unity adalah *JavaScript*, *C#*, dan *Boo.*

- **Android SDK (Software Development Kit),** adalah tools API yang diperlukan untuk pengembangan aplikasi pada platform android, atau dapat dikatakan sebagai perangkat lunak yang digunakan dalam pengembangan aplikasi pada android.

- **Photoshop,** adalah perangkat lunak editor citra buatan Adobe Systems. Selain untuk editor citra photoshop dapat digunakan untuk membuat desain GUI, desain world, serta desain karakter dari sebuah game.

- **3D Blender** adalah aplikasi grafik komputer yang memungkinkan kita untuk memproduksi suatu gambar atau animasi berkualitas tinggi dengan menggunakan geometri tiga dimensi. Tidak hanya untuk membuat suatu model atau animasi 3 dimensi, aplikasi 3D

Blender pun sudah cukup mumpuni untuk digital sculpting, mengedit video, 2D & 3D tracking, postproduction bahkan untuk membuat game. Dan aplikasi ini juga bisa di jalankan di berbagai macam *platform* sistem operasi, seperti Microsoft Windows, Mac OS, Linux, dan lain-lain. Yang membuat 3D Blender berbeda dari perangkat lunak 3D lainnya adalah aplikasi 3D Blender merupakan proyek *open source* dan diberikan secara gratis. Proyek *open source* seperti 3D Blender mengandalkan bantuan dari penggunanya untuk ikut mengembangkan atau membiayai pengembangan software ini. Karakteristik lain dari proyek *open source* adalah sifatnya yang terbuka. Di mana *source code* asli dari 3D Blender bisa diperoleh oleh siapa saja. Diharapkan mereka yang memperoleh *source code*-nya dapat membantu pengembangan dengan menambahkan fitur atau perbaikan tertentu pada 3D Blender(bpptik.kominfo, 2014).

2.10 Penelitian Terkait

1. Penelitian yang dilakukan oleh Maulidiawati Sri Wahyuningsih "Penerapan Metode Perceptron Untuk Leveling Fitur Kuis Pada Aplikasi Pembelajaran Bahasa Korea" mengutarakan bahwa Algoritma Perceptron dapat digunakan untuk mengenali pola kenaikan level pada fitur kuis dalam aplikasi pembelajaran bahasa korea. Dengan algoritma tersebut membuat pengguna dapat melompati level, dari level mudah langsung ke level sulit tanpa harus melewati level sedang sesuai dengan rule yang diterapkan.
2. Zulfian Azmi dalam penelitiannya "Aplikasi Jaringan Syaraf Tiruan Untuk Pengenalan Pola Pembukaan Permainan Catur" menggunakan model perceptron yang digunakan untuk mengenali pola pembukaan permainan catur Ruy Lopez atau bukan. Dari penelitian tersebut didapat kesimpulan bahwa Aplikasi Jaringan Syaraf Tiruan yang telah dibuat dapat digunakan untuk pengenalan pola pembukaan Ruy Lopez dengan menggunakan metode Perceptron dengan tampilan output, pembukaan Ruy Lopez atau bukan Ruy Lopez.
3. Ardi Pujiyanta dalam penelitiannya "Pengenalan Citra Objek Sederhana Dengan Jaringan Saraf Tiruan Metode Perceptron" menyebutkan bahwa algoritma perceptron dapat digunakan untuk mengenali pola pencitraan objek sederhana.

4. Penelitian yang dilakukan oleh Mike Susmikanti, Entin Hartini dan Antonius Sitompul "Identifikasi Pengaruh Umur, Suhu Dan Radiasi

Terhadap Strukturmikro Ferritic Steel Berbasis Kecerdasan Buatan" menyebutkan, pada proses pengenalan pola strukturmikro bahan ferritic dengan efek radiasi, suhu, umur dan yang belum mengalami perubahan, dalam pembelajaran, pelatihan dan simulasi, diperoleh hasil yang sesuai dan yang diharapkan. Dengan demikian dapat disimpulkan bahwa metode perceptron, sesuai untuk pengenalan pola klasifikasi strukturmikro bahan dalam system jaringan syaraf.

5. Ryan Mas Aryo Brilliant dalam penelitiannya "Analisis Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Perceptron Dalam Memprediksi Penyakit Jantung Koroner" mengambil kesimpulan bahwa metode perceptron lebih baik dan lebih cepat dibandingkan menggunakan metode backpropagation, karena tingkat akurasi dari metode perceptron lebih tinggi bisa mencapai 100% dibandingkan menggunakan metode backpropagation.
6. Yessy Asri dalam penelitiannya "Penerapan Aturan Perceptron Pada Jaringan Saraf Tiruan Dalam Pengenalan Pola Penyakit Mata" mengambil kesimpulan bahwa aturan perceptron mampu mengenali semua pola penyakit mata dengan tingkat kecocokan mencapai 100%.
7. Penelitian yang dilakukan oleh Mike Susmikanti, dan Arya Adhiyaksa mengenai "Identifikasi Huruf Menggunakan Metode Pembelajaran Perceptron Dalam Jaringan Neural" menyebutkan bahwa perceptron adalah metode yang tepat dalam persoalan mengidentifikasi huruf.

BAB III

DESAIN DAN PERANCANGAN APLIKASI

3.1 Deskripsi Sistem

Aplikasi yang dibuat dalam penilitian ini adalah sebuah game pembelajaran dengan materi mengenai malware. Game ini dibuat untuk *platform android*. Dalam game ini player akan bertarung melawan berbagai macam malware. Malware-malware dalam game ini memiliki tampilan serta berbagai serangan(*moveset*) yang berbeda-beda. Pengguna dapat belajar mengenai malware dari tampilan dan moveset yang dimiliki oleh malware tersebut. Dengan mengetahui serangan yang dilakukan oleh malware dalam game, maka pengguna dapat mengetahui malware apa yang menyerang ketika suatu saat smartphone atau komputer yang dimiliki oleh pengguna diserang di kehidupa nyata. Selain itu juga *player* yang dimainkan oleh pengguna juga memiliki moveset, yang mana salah satu atau lebih serangan dari moveset ini adalah kelemahan dari malware-malware tersebut, tergantung malware apa yang dihadapi oleh player. Dalam hal ini pengguna dapat mempelajari cara menangani malware yang tepat ketika menghadapinya di kehidupan nyata.

3.2 Penggunaan Aplikasi

Game pembelajaran malware ini membantu pengguna untuk mengenal malware serta cara penanganannya melalui pertarungan dalam game. Pertarungan dalam game ini menggunakan *turn-based system*, yang mana antara pengguna dan

NPC akan menyerang secara bergantian sampai salah satu diantara mereka kalah. Untuk memainkan game ini pengguna cukup menekan tombol play pada *main menu*. Setelah itu pengguna akan dihadapkan pada menu *stage select.* Pengguna tinggal menekan *stage* yang ingin dimainkan.

Setelah memilih *stage* pengguna akan bermain sebagai *player* didalam *stage* yang telah dipilih. Dalam *stage* tersebut terdapat malware yang harus dikalahkan. Ketika *player* melakukan kontak dengan malware tersebut maka *player* akan memasuki *battle scene*. Dalam *battle scene* player akan bertarung melawan malware yang telah melakukan kontak. Pengguna dapat memilih aksi yang akan diambil ketika bearada di dalam *battle scene* atau saat bertarung melawan malware, apakah akan melawan, *use item*, atau *run*. Saat pengguna memilih untuk melawan, maka pengguna akan dihadapkan pada pilihan *moveset* yang tersedia. Jika memilih *use item* maka pengguna dapat memilih item yang ingin digunakan. Item-item ini memiliki efek yang berbeda-beda seperti menambah *health point*, menambah *power point*, atau menambah *defense point*. Jika pengguna memilih *run*, maka *player* akan meninggalkan *battle scene*. Selain itu setiap kali *player* berhasil menghancurkan *malware* maka *player* akan memperoleh *level up* atau kenaikan tingkat *player*. Setiap kali *player* mengalami *level up* maka *power point* serta *defense point* dari *player* akan meningkat.

3.3 Materi Pada Aplikasi

Materi yang diberikan dalam game ini berupa *malware* yang harus dihancurkan oleh *player***.** Malware-malware tersebut memiliki moveset yang berbeda-beda sesuai dengan jenis malwarenya. Moveset dari malware-malware tersebut disesuaikan dengan serangan dari malware yang sebenarnya. Sehingga

pengguna dapat mempelajari serangan yang dilakukan oleh malware yang sebenarnya, berdasarkan serangan dari malware dalam game. Selain itu juga dengan mengetahui serangan yang efektif untuk melawan malware dalam game pengguna dapat mempelajari cara menangani malware dalam kehidupan nyata.

3.4 Game Mechanic

Masing-masing *actor* dalam game ini baik *NPC* maupun *player* keduanya memiliki *attributes* yang sama yaitu *power, defense,* serta *health point. Power* adalah ukuran besarnya nilai kekuatan dari *NPC* atau *player*. *Defense* adalah ukuran besarnya nilai pertahanan dari *NPC* atau *player. Health* merupakan ukuran kesehatan *NPC* atau *player.*

Dalam *battle scene* pertarungan dianggap selesai bila salah satu dari *player* atau *NPC* memiliki *health point (HP)* kurang dari sama dengan nol. Setiap kali *player* atau *NPC* menyerang maka lawan dari *player* atau *NPC* akan menerima *damage* atau kerusakan. *HP* akan dikurangi dengan *damage* tiap kali *player* atau *NPC* terkena serangan. Besarnya *damage* yang diterima dipengaruhi oleh *power, defense,* dan besarnya nilai kekuatan dari serangan yang dilancarkan.

Damage memiliki perhitungan sebagai berikut:

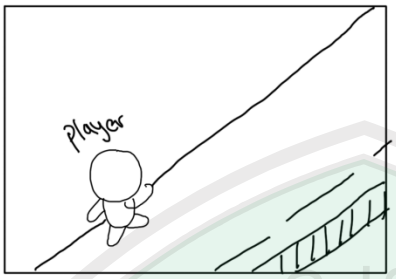
Damage NPC = (NPC power+nilai kekuatan serangan NPC)-defense player

Damage player = (player power+nilai kekuatan serangan player)-defense NPC

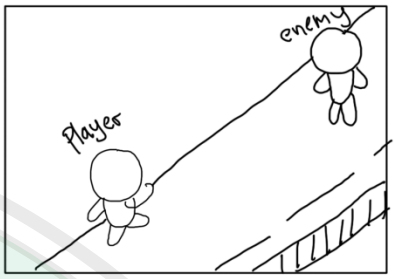
Khusus untuk *player* ketika serangan yang dipilih merupakan kelemahan dari *NPC,* maka damage yang dihasilkan *player* akan dikalikan dua.

UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM
PUSAT PERPUSTAKAAN

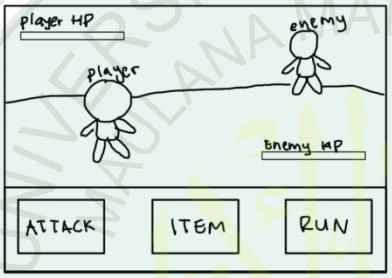
3.5 Storyboard



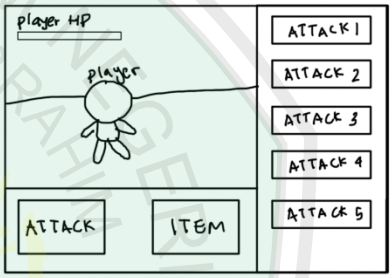
1. Player akan berjalan sepanjang platform dari start sampai ke finish



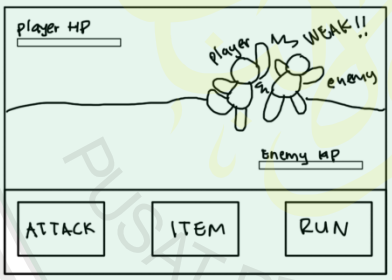
2. Selama perjalanan player dapat bertemu dengan musuh yang menghadang. Player dapat melawannya dengan kontak dengan musuh atau menghindar dengan melompatinya.



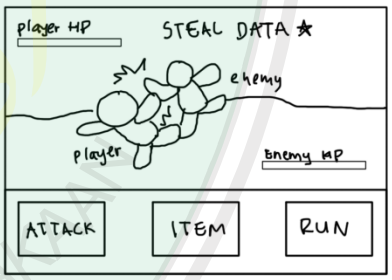
3. Jika player kontak dengan musuh, player dan musuh akan masuk kedalam battle scene. Player dapat memilih aksi yang akan dilakukan.



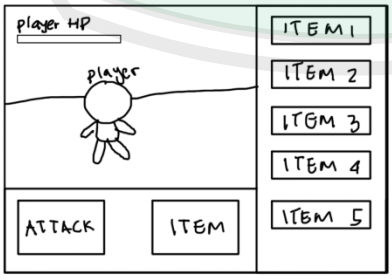
4. Jika player memilih untuk menyerang maka akan muncul moveset yang bisa dipilih oleh player.



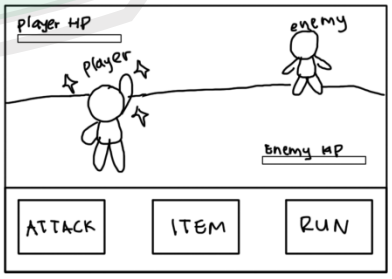
5. Setelah memilih moveset, player akan melancarkan serangan kepada musuh sesuai dengan moveset yang dipilih. Bila moveset yang dipilih adalah kelemahan musuh maka akan muncul tulisan “weak”.



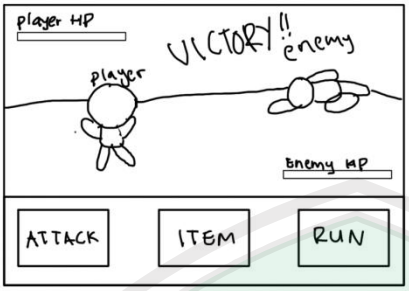
6. Setelah player selesai melancarkan aksinya, berikutnya giliran musuh untuk menyerang bergantian terus menerus dengan player. Tiap kali musuh menyerang akan muncul nama serangan yang dilancarkan.



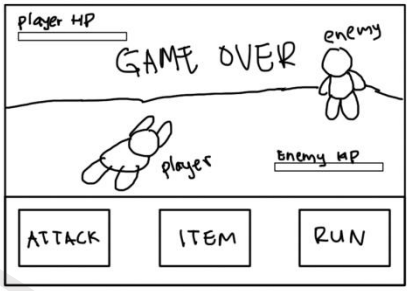
7. Selain menyerang player juga dapat menggunakan item. Sama seperti menyerang setelah menekan tombol item akan muncul pilhan item yang bisa digunakan.



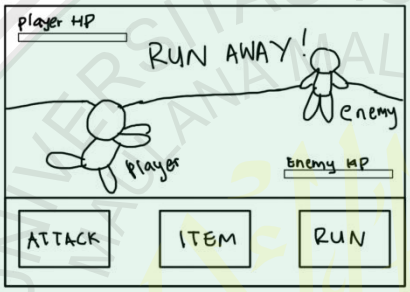
8. Setelah memilih item yang digunakan, player akan menjalankan animasi item use dan berikutnya masuk ke giliran musuh.



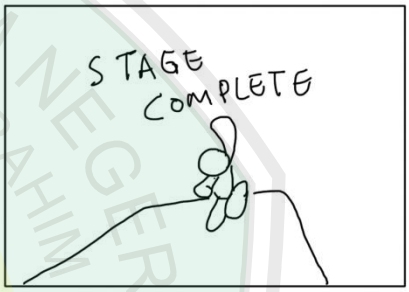
9. Bila HP dari musuh kurang dari sama dengan 0, maka player memenangkan pertarungan dan kembali ke platform untuk melanjutkan perjalanan.



10. Bila HP dari player kurang dari sama dengan 0, maka player kalah dalam pertarungan dan kembali ke stage select menu.



11. Selain menyerang dan menggunakan item player juga dapat melarikan diri dari pertarungan dengan menekan tombol run. Player keluar dari battle scene dan kembali ke platform.



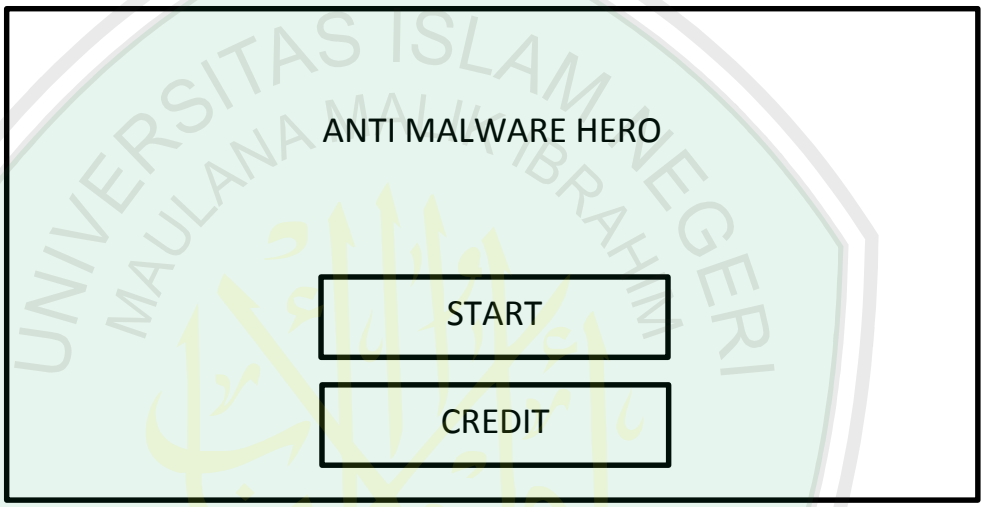
12. Setelah player mencapai finish maka player akan kembali ke stage select menu.

Gambar 3. 1 Storyboard Game

3.6 Rancangan Tampilan

Tampilan *Main Menu*

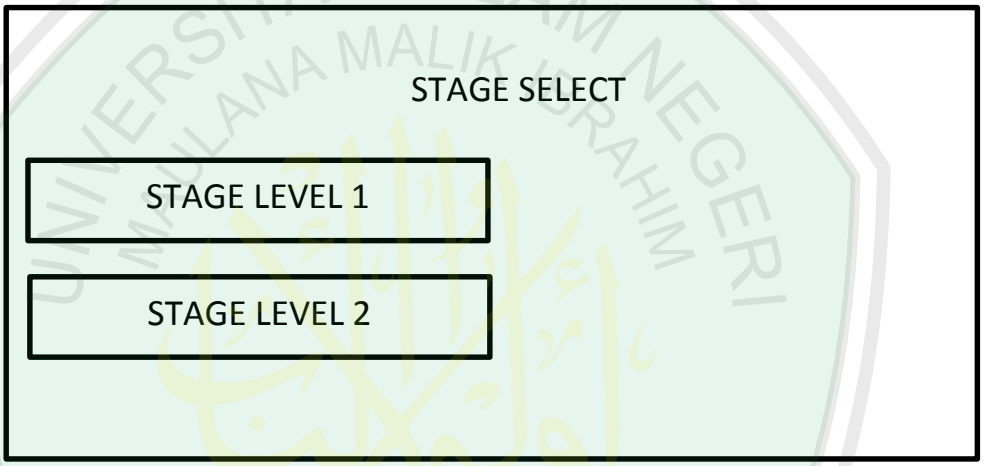
Tampilan *Main Menu* merupakan tampilan awal yang pertama kali muncul ketika pengguna menjalankan aplikasi ini. Dalam *Main Menu* pengguna dapat memilih untuk memainkan *game* atau melihat *Credit Menu.*



Gambar 3. 2 Tampilan Main Menu

Tampilan *Stage Select*

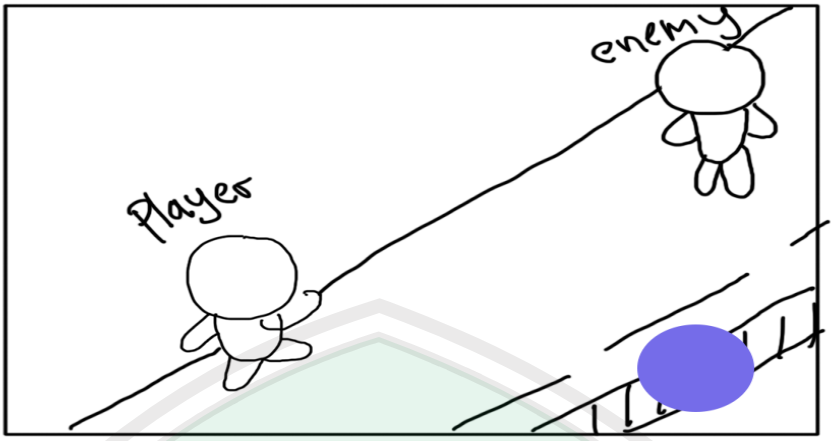
Dalam menu *Stage Select* pengguna dapat memilih *level stage* tempat pengguna bermain nantinya. Dalam tiap-tiap *level stage* terdapat musuh atau *malware* yang berbeda-beda. Terdapat dua *level stage* yang dapat dimainkan pengguna.



Gambar 3. 3 Tampilan Stage Select

Tampilan *Stage* Permainan

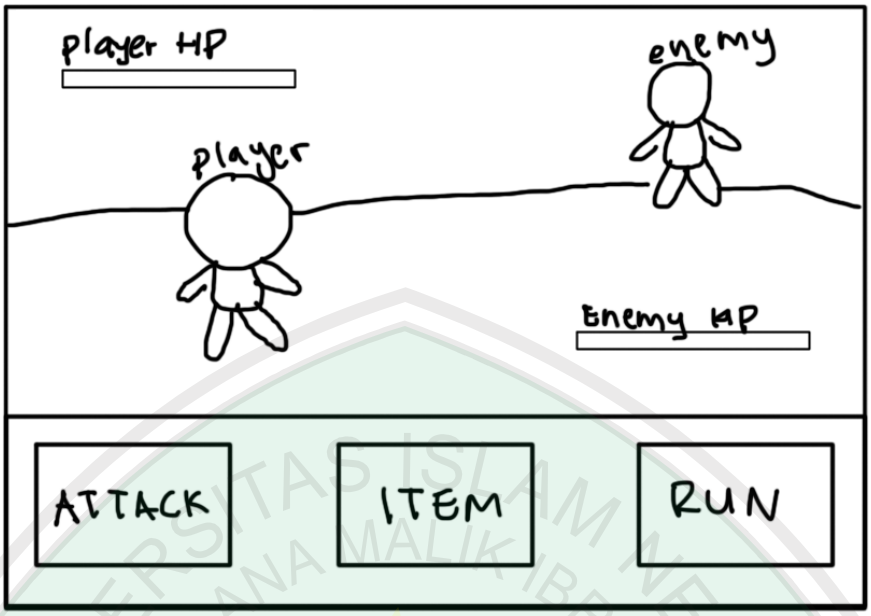
Stage merupakan tempat pengguna bermain. Dalam *stage* terdapat *platform* tempat *player* berpijak serta *NPC Malware* yang harus dilawan. Pengguna akan berjalan sepanjang *platform* untuk melawan *NPC Malware* hingga di ujung platform. Selain itu juga terdapat tombol *jump* yang bisa digunakan untuk lompat ke *platform* yang lebih tinggi atau untuk menghindari pertarungan.



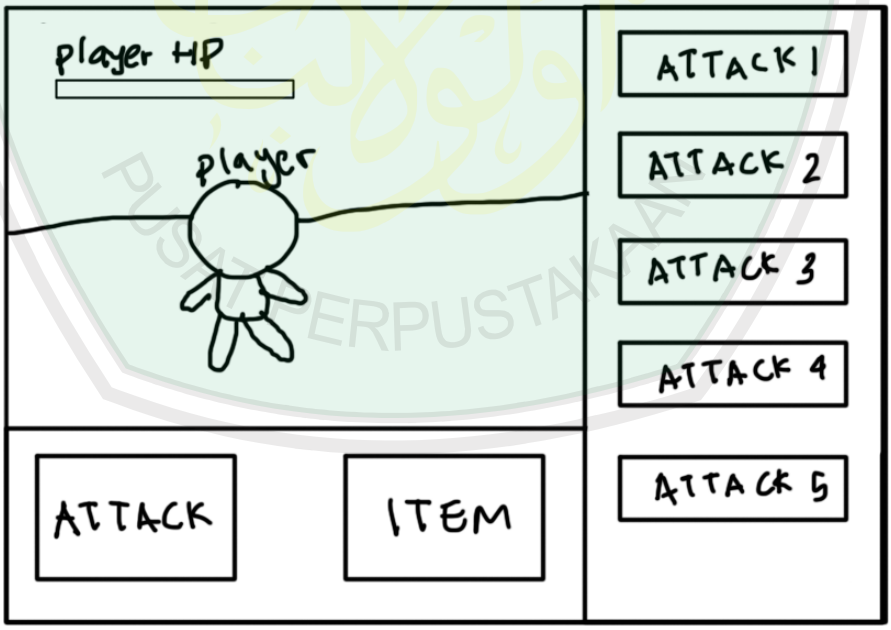
Gambar 3. 4 Tampilan Stage

Tampilan *Battle Scene*

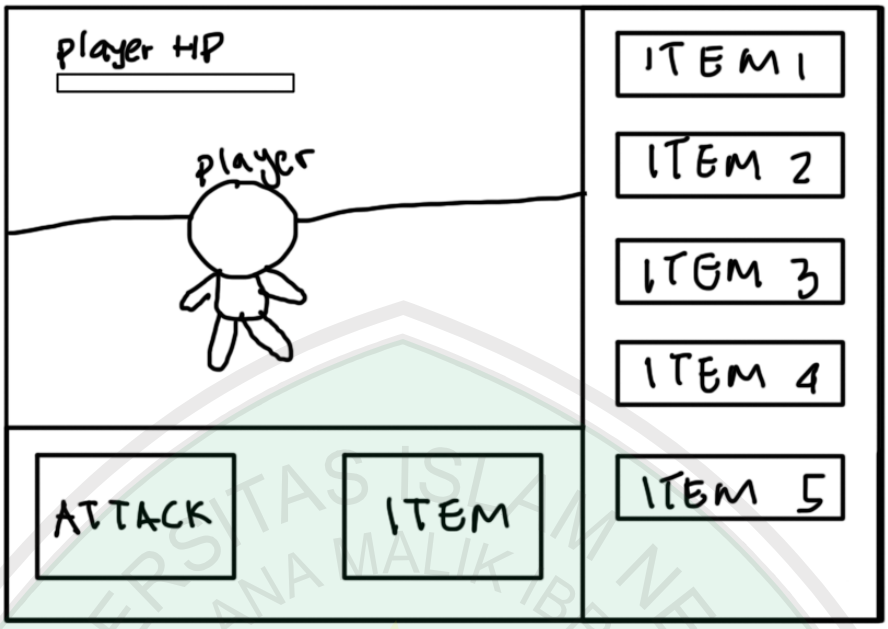
Ketika *player* dan *NPC* saling melakukan kontak maka mereka akan memasuki *Battle Scene*. Dalam *Battle Scene* terdapat tiga pilihan yaitu *attack, item*, serta *run*. Ketika pengguna menekan tombol *attack*, maka akan muncul pilihan *moveset*. Bila pengguna menekan tombol *item*, maka akan muncul pilihan *item* yang bisa digunakan. Bila menekan tombol *run*, maka akan kembali ke *stage*. Tidak melupakan sisi pembelajaran didalam game tiap kali *NPC* menyerang maka akan muncul nama dari serangan tersebut di dalam battle scene. Sehingga pengguna dapat mempelajari serangan dari tiap-tiap *malware*. Selain itu juga tiap kali pengguna memilih serangan yang merupakan kelemahan dari *malware* yang dilawan, maka akan muncul tulisan "*weak*". Sehingga pengguna dapat mempelajari cara menangani *malware* tersebut. Selain itu ditampilkan juga *bar HP* dari *player* dan *NPC*.



Gambar 3. 5 Tampilan Battle Scene



Gambar 3. 6 Tampilan pilih serangan



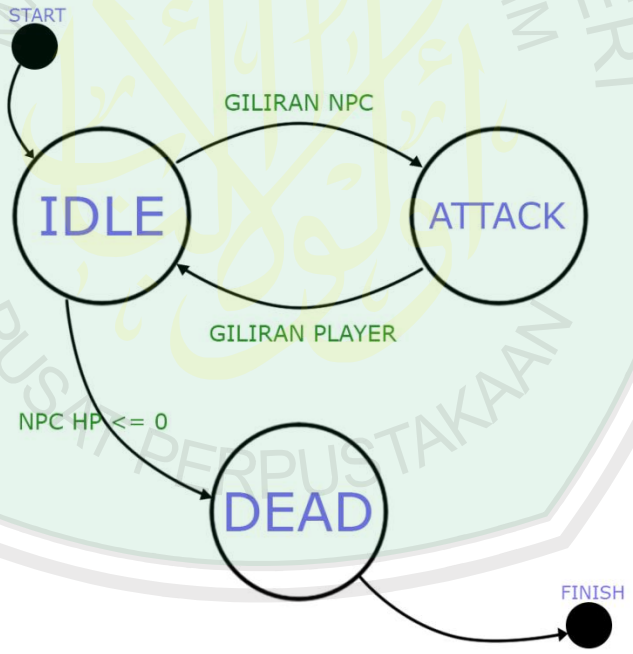
Gambar 3. 7 Tampilan pilih item

3.7 Rancangan NPC

Dalam game ini terdapat dua jenis *NPC*, yaitu *normal NPC* dan *boss NPC*. *Normal NPC* terdiri dari *Virus* dan *Adware*, sedangkan *boss NPC* terdiri dari *Trojan* dan *Worm*. *Normal NPC* adalah *NPC* yang menghadang *player* selama perjalanan dan dapat ditemukan di sepanjang *platform*. *Boss NPC* adalah *NPC* yang menjadi target utama dan harus dikalahkan oleh *Player*. *Boss NPC* berada dekat dari ujung *platform*. *Player* tidak dapat melompati atau menggunakan *Run* saat melawan *boss NPC*.

Adware

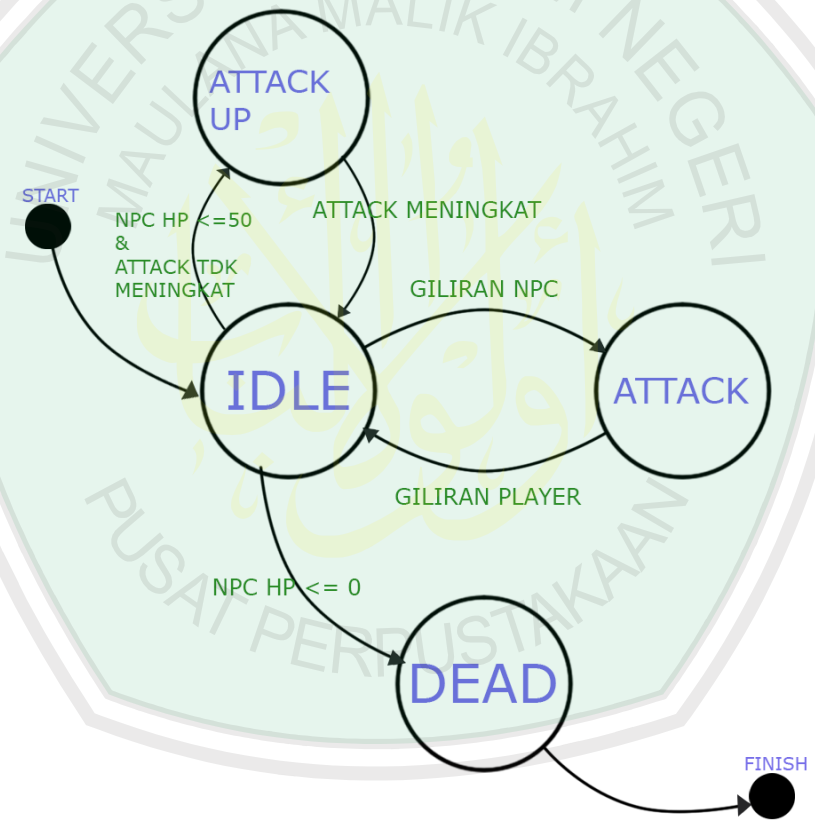
Pada gambar 3.8 menunjukkan skenario dari *NPC Adware.* Karena game ini *bergenre turn-based RPG,* maka antara *player* dan *NPC* akan menyerang secara bergantian. Dari gambar di bawah ketika memasuki giliran *player,* maka *NPC* menjadi posisi *idle* atau tidak melakukan apa-apa sampai giliran *player* selesai. Setelah giliran *player* selesai berikutnya adalah memasuki giliran *NPC*. Saat memasuki giliran *NPC, NPC* akan menyerang player sesuai dengan serangan yang dipilih oleh *NPC* bila *HP* dari *NPC* tidak kurang dari sama dengan nol. Bila *HP* dari *NPC* kurang dari sama dengan nol maka *NPC* akan mati atau hancur.



Gambar 3. 8 FSM NPC ADWARE

Virus

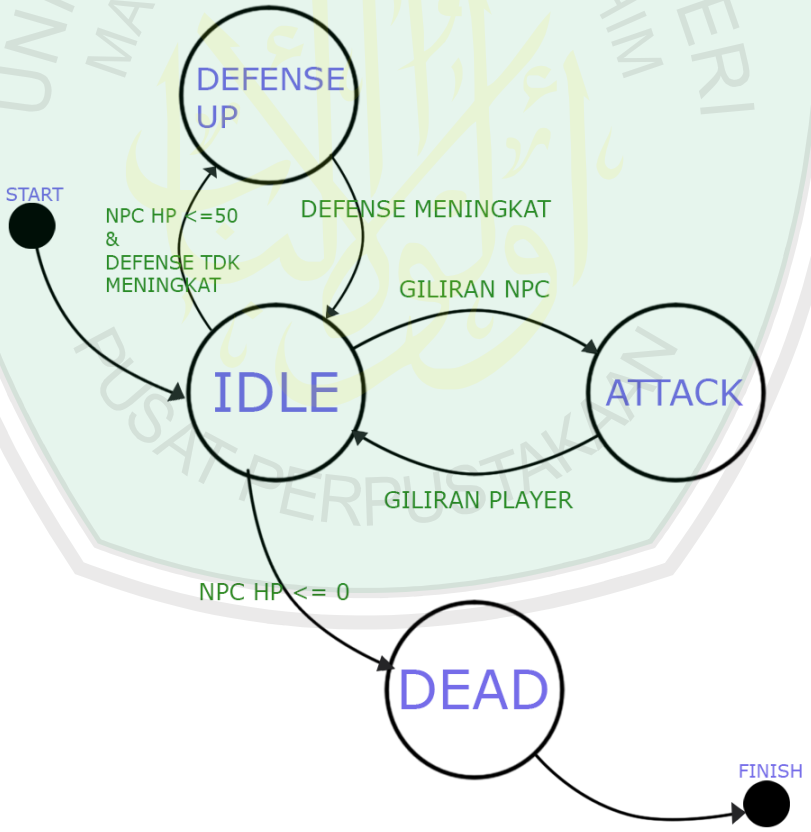
Pada gambar 3.9 menunjukkan skenario dari *NPC Virus*. Virus komputer dapat menggandakan dirinya ketika seseorang mengaktifkan program yang telah tertanam virus didalamnya. Hal ini dalam skenario digambarkan sebagai *attack up*. Karena ketika virus sudah menyebar ke dalam suatu sistem atau komputer maka serangan dari virus ini akan semakin berbahaya atau bisa dikatakan *attack* dari virus meningkat.



Gambar 3. 9 FSM NPC VIRUS

Trojan

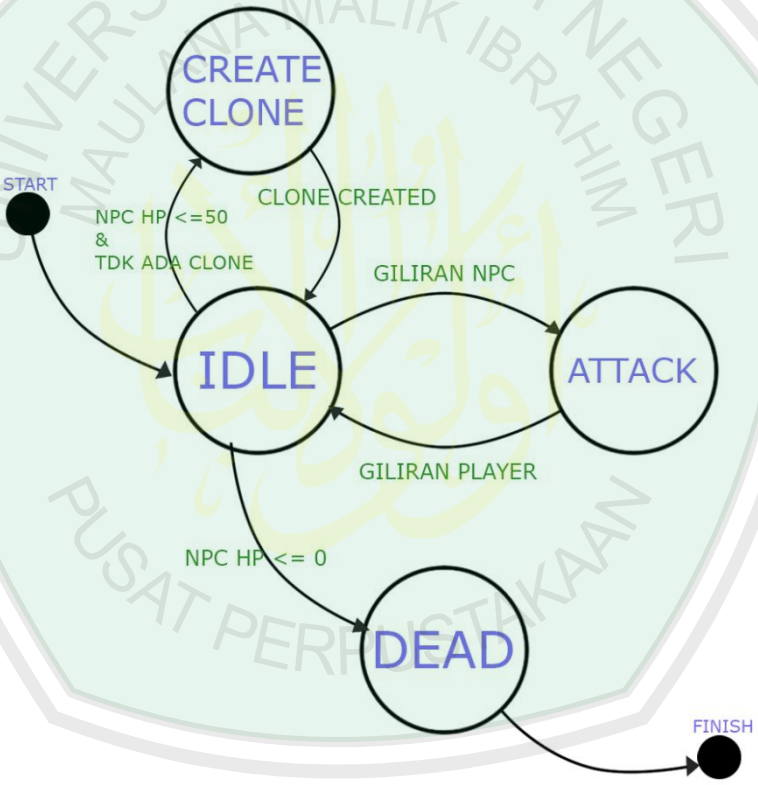
Pada gambar 3.10 menunjukkan skenario dari *NPC Trojan.* Trojan adalah program yang memiliki tujuan untuk mendapatkan informasi-informasi penting yang terdapat pada komputer yang terinfeksi. Dalam aksinya Trojan cenderung bersifat tersembunyi atau stealth, dengan cara menyamar menjadi program yang baik-baik. Kemampuan ini dalam skenario digambarkan sebagai *defense up*. Kemampuan Trojan untuk menyamar adalah upaya untuk melindungi dirinya agar tidak diketahui oleh user, atau bisa dikatakan meningkatkan defense dari ancaman yang bisa membahayakan si Trojan.



Gambar 3. 10 FSM NPC TROJAN

Worm

Pada gambar 3.11 menunjukkan skenario dari *NPC Worm*. Sama seperti virus, worm dapat menggandakan dirinya, namun worm tidak membutuhkan pihak ketiga untuk menggandakan dirinya. Worm dapat menggandakan dirinya sesuai dengan keinginannya sendiri. Dalam skenario hal ini digambarkan dengan *create clone*. Clone yang diciptakan akan membantu worm dalam melancarkan aksi jahatnya.

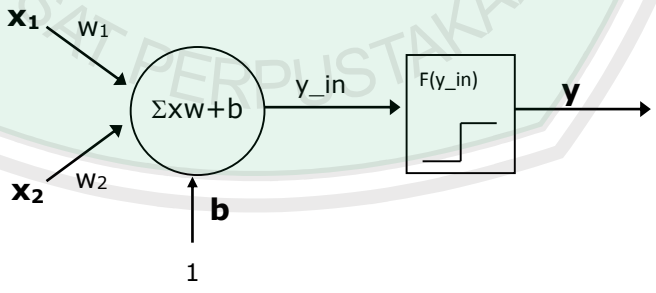


Gambar 3. 11 FSM NPC WORM

Moveset dari keempat *NPC* tersebut terdiri dari tiga jenis serangan yaitu *weak attack*, *heavy attack*, dan *special attack*. Serangan-serangan tersebut akan dipilih oleh *NPC* ketika akan menyerang player. Serangan apa yang akan dipilih oleh *NPC* itu tergantung dari parameter-parameter yang telah ditentukan. Parameter tersebut adalah *player health* dan *NPC health*. Masing-masing serangan yang akan dipilih oleh NPC memiliki aturan parameter yang berbeda-beda. Setiap jenis-jenis serangan memiliki nilai kekuatan yang berbeda-beda. *Heavy attack* memiliki nilai kekuatan yang lebih besar daripada *weak attack*. *Special attack* memiliki nilai kekuatan yang lebih besar dari pada *heavy* dan *weak attack*.

3.8 Penerapan Metode Perceptron

Pada kasus pemilihan serangan oleh *NPC* menggunakan perceptron single layer, karena memberikan solusi yang sifatnya sederhana yaitu untuk mengklasifikasikan jenis serangan. Perceptron yang diterapkan pada NPC terdiri dari dua input dan sebuah output.



Gambar 3. 12 Desain Arsitektur Jaringan Perceptron

Output yang dihasilkan merupakan salah satu dari tiga kategori serangan milik NPC, yaitu *weak attack, heavy attack*, atau *special attack*. Menentukan serangan yang akan dipilih oleh NPC di tinjau dari 2 faktor atau input, yaitu *player health* dan *NPC health*.

Nilai *health* maksimal yang bisa dimiliki oleh *NPC* dan *player* adalah 100 point. Sedangkan nilai minimalnya adalah 0 point, yaitu ketika *player* atau *NPC* mati atau kalah. Dari data-data input tersebut dibuat dataset yang nantinya pola dari dataset tersebut akan dipelajari oleh perceptron.

Angka 1 – 10 melambangkan range dari NPC health dan player health yang terbagi kedalam 10 group

0 - 10 = 1

11 - 20 = 2

21 - 30 = 3

31 - 40 = 4

41 - 50 = 5

51 - 60 = 6

61 - 70 = 7

71 - 80 = 8

81 - 90 = 9

91 - 100 = 10

Berikut adalah data target yang diberikan:

Player Health	NPC Health	Target
10	10	*heavy attack*
9	10	*heavy attack*
8	10	*heavy attack*
7	10	*heavy attack*
6	10	*heavy attack*
5	10	*heavy attack*
4	10	*heavy attack*
3	10	*heavy attack*
2	10	*heavy attack*
1	10	*heavy attack*
10	9	*heavy attack*
9	9	*heavy attack*
8	9	*heavy attack*
7	9	*heavy attack*
6	9	*heavy attack*
5	9	*heavy attack*
4	9	*heavy attack*
3	9	*heavy attack*
2	9	*heavy attack*
1	9	*heavy attack*
10	8	*special attack*
9	8	*heavy attack*
8	8	*heavy attack*
7	8	*heavy attack*
6	8	*heavy attack*
5	8	*heavy attack*
4	8	*heavy attack*

3	8	*heavy attack*
2	8	*heavy attack*
1	8	*heavy attack*
10	7	*special attack*
9	7	*special attack*
8	7	*heavy attack*
7	7	*heavy attack*
6	7	*heavy attack*
5	7	*heavy attack*
4	7	*heavy attack*
3	7	*heavy attack*
2	7	*heavy attack*
1	7	*heavy attack*
10	6	*special attack*
9	6	*special attack*
8	6	*special attack*
7	6	*heavy attack*
6	6	*heavy attack*
5	6	*heavy attack*
4	6	*heavy attack*
3	6	*heavy attack*
2	6	*heavy attack*
1	6	*heavy attack*
10	5	*special attack*
9	5	*special attack*
8	5	*special attack*
7	5	*special attack*
6	5	*heavy attack*
5	5	*heavy attack*

4	5	*heavy attack*
3	5	*heavy attack*
2	5	*heavy attack*
1	5	*heavy attack*
10	4	*special attack*
9	4	*special attack*
8	4	*special attack*
7	4	*special attack*
6	4	*special attack*
5	4	*heavy attack*
4	4	*heavy attack*
3	4	*heavy attack*
2	4	*heavy attack*
1	4	*heavy attack*
10	3	*weak attack*
9	3	*weak attack*
8	3	*weak attack*
7	3	*weak attack*
6	3	*weak attack*
5	3	*weak attack*
4	3	*heavy attack*
3	3	*heavy attack*
2	3	*heavy attack*
1	3	*heavy attack*
10	2	*weak attack*
9	2	*weak attack*
8	2	*weak attack*
7	2	*weak attack*
6	2	*weak attack*

5	2	*weak attack*
4	2	*weak attack*
3	2	*heavy attack*
2	2	*heavy attack*
1	2	*heavy attack*
10	1	*weak attack*
9	1	*weak attack*
8	1	*weak attack*
7	1	*weak attack*
6	1	*weak attack*
5	1	*weak attack*
4	1	*weak attack*
3	1	*weak attack*
2	1	*heavy attack*
1	1	*heavy attack*

Tabel 3. 1 Tabel Dataset

3.9 Kebutuhan Sistem

Berikut sistem yang dibutuhkan untuk mendukung pengoperasian aplikasi ini di antaranya:

a. Perangkat keras (*Hardware*)
   - PC/Laptop dengan spesifikasi *processor* Intel(R) Core(TM) i3-2370M CPU @ 2.40 GHz 2.40GHz, RAM 4 GB digunakan untuk pembuatan aplikasi
   - *Smartphone* dengan spesifikasi *Android OS* versi 4.1.2 digunakan untuk uji coba aplikasi.

b. *Game Engine (Software)*

   Unity3D**,** game engine yang dapat digunakan untuk mengembangkan game yang berbasis *mobile.*

c. Android SDK tools API yang diperlukan untuk pengembangan aplikasi pada platform android.

d. Photoshop perangkat lunak editor citra yang dapat digunakan untuk membuat desain GUI, desain world, serta desain karakter dari sebuah game.

e. Blender digunakan untuk membuat 3d model karakter dari game. Versi blender yang digunakan adalah versi 2.73.

BAB IV

HASIL DAN PEMBAHASAN

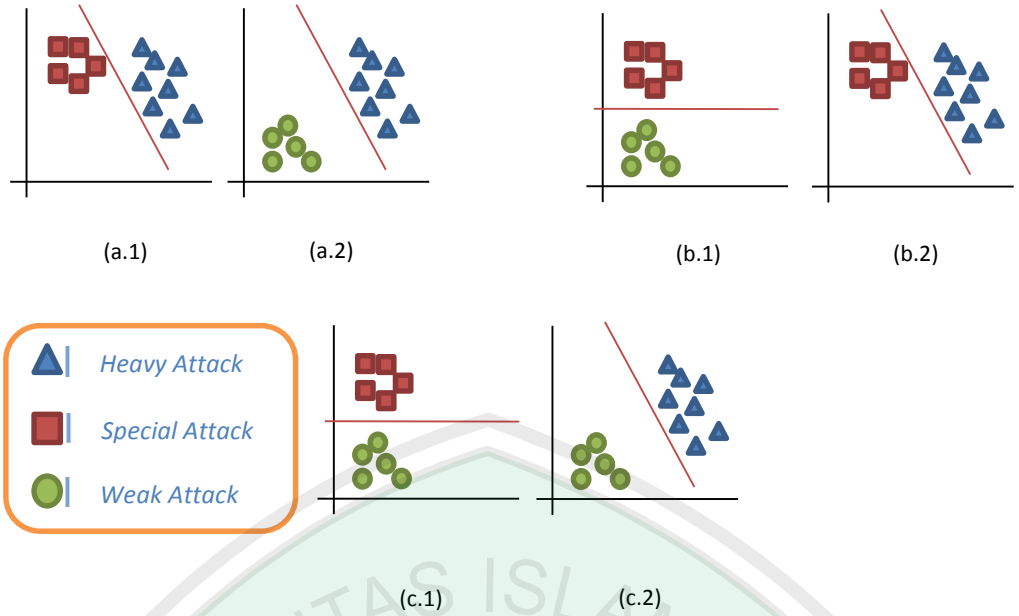
Dalam bab ini mengulas mengenai hasil dan pembahasan tentang implementasi *algoritma perceptron* ke dalam game pembelajaran *malware*.

4.1 Implementasi Algoritma Perceptron pada Pemilihan Serangan Oleh NPC

Algoritma perceptron diimplementasikan kepada *NPC* agar *NPC* dapat memilih serangan yang akan dilancarkan berdasarkan kondisi yang ada. Pada dasarnya *perceptron* digunakan untuk mengklasifikasikan tipe *class* serangan. *Perceptron* mampu mengklasifikasikan tipe *class* menjadi dua *class*, yaitu daerah positif dan daerah negatif. Masing-masing daerah dipisahkan oleh sebuah garis secara linear. Pengklasifikasian didasarkan pada input yang diberikan.

Dalam game pembelajaran *malware*, *algoritma perceptron* diimplementasikan untuk mengklasifikasikan tiga jenis serangan yang akan dipilih oleh *NPC*. Karena pada dasarnya *algoritma perceptron* digunakan untuk mengklasifikasikan dua tipe *class*, maka diperlukan perlakuan khusus agar metode perceptron dapat diimplementasikan ke dalam game ini.

Untuk dapat mengklasifikasikan tiga jenis serangan menggunakan perceptron, maka dilakukan penerapan perceptron dengan dua tahap atau perceptron dua stage. Masing-masing *class* akan diklasifikasikan dengan *class-class* yang lain membentuk beberapa group.



Gambar 4. 1 (a.1)Heavy1, (a.2)Heavy2, (b.1)Special1, (b.2)Special2, (c.1)Weak1, (c.2)Weak2.

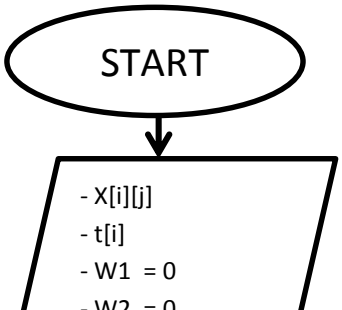
Heavy1 adalah group antara *heavy attack* dengan *special attack*. Dalam gambar di atas *heavy attack* bernilai +1 sedangkan *special attack* bernilai -1. *Heavy2* adalah group antara *heavy attack* dengan *weak attack*. Dalam gambar di atas *heavy attack* bernilai +1 sedangkan *weak attack* bernilai -1. *Special1* adalah group antara *special attack* dengan *weak attack*. Dalam gambar di atas *special attack* bernilai +1 sedangkan *weak attack* bernilai -1. *Special2* adalah group antara *special attack* dengan *heavy attack*. Dalam gambar di atas *special attack* bernilai +1 sedangkan *heavy attack* bernilai -1. *Weak1* adalah group antara *weak attack* dengan *special attack*. Dalam gambar di atas *weak attack* bernilai +1 sedangkan *special attack* bernilai -1.

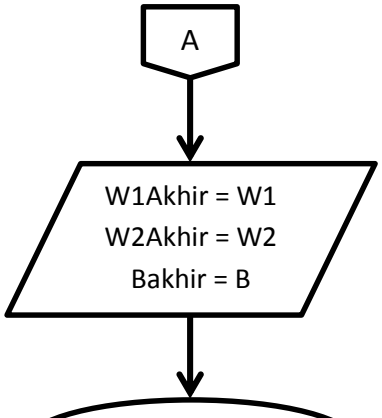
Weak2 adalah group antara *weak attack* dengan *heavy attack*. Dalam gambar 4.2 *weak attack* bernilai +1 sedangkan *heavy attack* bernilai -1. Grup *Heavy1* dan

Heavy2 mewakili klasifikasi *class Heavy Attack,* karena pada *Heavy1* dan *Heavy2* nilai *Heavy Attack* bernilai +1. Grup *Special1* dan *Special2* mewakili klasifikasi *class Special Attack,* karena pada *Special1* dan *Special2* nilai *Special Attack* bernilai +1. Grup *Weak1* dan *Weak2* mewakili klasifikasi *class Weak Attack,* karena pada *Weak1* dan *Weak2* nilai *Weak Attack* bernilai +1.

Dari pengklasifikasian tersebut dapat dipilih *class* yang sesuai dengan input yang dimasukkan. Input yang dimasukkan kedalam masing-masing group klasifikasi akan menghasilkan nilai +1 atau -1. Setelah masing-masing grup mendapatkan nilainya, langkah berikutnya adalah menghitung nilai +1 yang diperoleh oleh masing-masing *class.* Karena masing-masing grup mewakili sebuah *class* maka kita bisa menghitung nilai +1 yang diperoleh suatu *class* dengan menghitung nilai yang diperoleh masing-masing grup. *Class* yang memiliki nilai +1 terbanyak adalah *class* yang terpilih.

Berikut adalah *flowchart* dari implementasi *algoritma perceptron.*




A
W1Akhir = W1
W2Akhir = W2
Bakhir = B

Gambar 4. 2 Flowchart training process dalam perceptron.

Gambar 4.3 menunjukkan proses dari *training* data dalam *algoritma perceptron* yang diimplementasikan ke dalam *game* pembelajaran *malware*. Untuk memulai proses training tersebut dibutuhkan data yang akan ditraining, yaitu input dan target. X[i][j] adalah input yang terdiri dari *player health* dan *NPC health*. Selanjutnya t[i] adalah target yang berupa *heavy attack, low attack,* dan *special attack* yang diinisialisasikan dengan nilai -1 atau +1 tergantung grup klasifikasi seperti pada gambar 4.2. Setelah menentukan data trainingnya langkah berikutnya adalah adalah menentukan bobot awal dan bias awal. Bobot awal dan bias awal dapat diisi berapa pun, namun untuk memudahkan pengerjaan bobot awal dan bias awal di set sama dengan nol. Bobot awal adalah W1 dan W2, serta bias awal adalah B. Selanjutnya adalah mengeset nilai learning rate (α) sebesar 0.8, dan nilai threshold (θ) sebesar 0.5. Proses training akan dilakukan terus menerus atau selama kondisi sama dengan false, sampai tidak terjadi kesalahan atau kondisi sama dengan true.

Setelah semua data input yang dibutuhkan telah diset berikutnya adalah menghitung nilai hasil aktivasi dengan rumus berikut:

$$y\_in = b + \sum_i x_i w_i$$

Setelah mendapatkan nilai dari aktivasi langkah berikutnya adalah membandingkan nilai tersebut dengan fungsi aktivasi sebagai berikut:

$$y = \begin{cases} 1, & jika\ y\_in > \theta \\ 0, & jika -\theta \le y\_in \le \theta \\ -1, & jika\ y\_in < -\theta \end{cases}$$

Langkah selanjutnya adalah mencocokkan nilai dari fungsi aktivasi yaitu y dengan target yang diinginkan yaitu t[i]. Bila nilai y dengan nilai t[i] mempunyai nilai yang sama, maka tidak perlu dilakukan perubahan nilai bobot dan bias. Namun bila nilai y dengan nilai t[i] tidak mempunyai nilai yang sama maka perlu dilakukan perubahan nilai bobot dan bias. Berikut adalah penghitungan nilai bobot dan bias baru:

$$W_{new1} = W_1 + \alpha * t[i] * X[i][j]$$

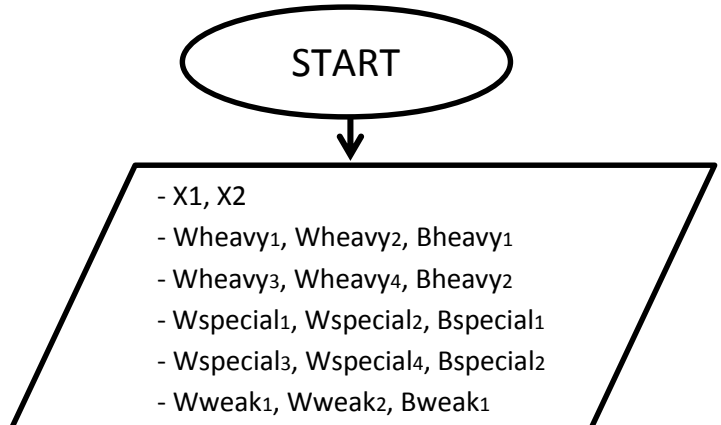
$$W_{new2} = W_2 + \alpha * t[i] * X[i][j+1]$$

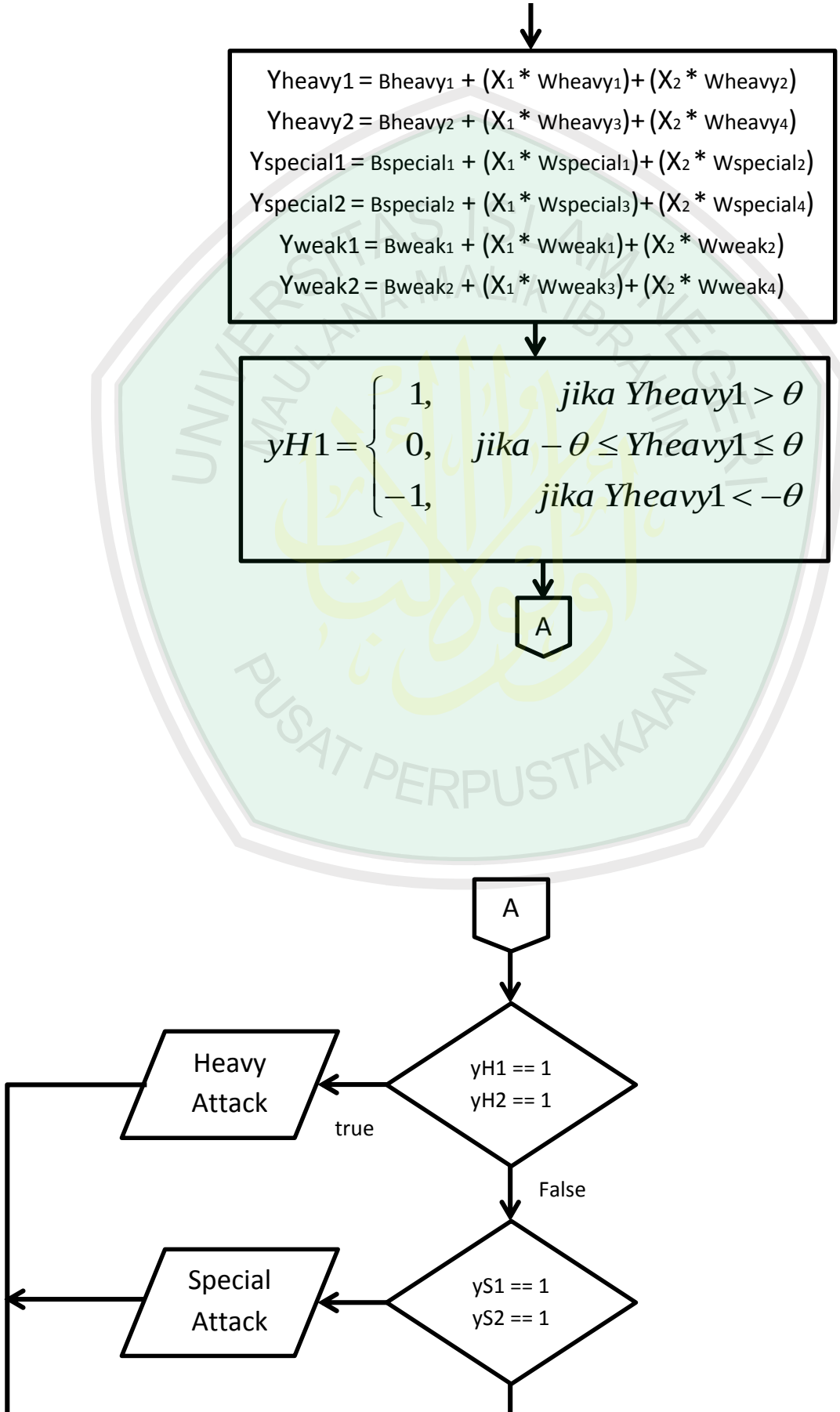
$$B_{new} = B + \alpha * t[i]$$

Setelah penghitungan selesai langkah selanjutnya adalah melanjutkan proses training ke data training yg berikutnya bila ada, menggunakan bobot baru maupun lama yang telah proses. Proses perhitungan ini dilakukan untuk semua data training yang ada. Jika semua data training telah ditraining yang mana jumlah data training dilambangkan dengan X.length, maka itu dihitung sebagai satu epoch. Setelah menyelesaikan satu epoch, langkah selanjutnya adalah mengecek kondisi, apakah terjadi kesalahan selama proses training, yaitu nilai y tidak sama dengan nilai t[i]. Bila dalam satu epoch training masih terjadi kesalahan maka kondisi benilai false dan perlu dilakukan proses training lagi dari awal menggunakan nilai bobot dan

bias yang terakhir didapat. Jika dalam suatu epoch tidak terjadi kesalahan, maka kondisi akan menjadi true dan proses training berhenti. Tujuan dari proses training ini adalah mencari bobot dan bias yang mana bobot dan bias tersebut mampu digunakan untuk mengklasifikasikan class serangan yang akan dipilih oleh *NPC*. Proses training ini akan dijalankan kepada semua grup klasifikasi, sehingga akan dihasilkan 12 bobot baru dan 6 bias baru.

Setelah mendapatkan semua bobot dan bias, langkah berikutnya adalah pengklasifikasian. Berikut adalah *flowchart* dari prose pengklasifikasian.




Yheavy1 = Bheavy1 + (X1 * Wheavy1)+ (X2 * Wheavy2)
Yheavy2 = Bheavy2 + (X1 * Wheavy3)+ (X2 * Wheavy4)
Yspecial1 = Bspecial1 + (X1 * Wspecial1)+ (X2 * Wspecial2)
Yspecial2 = Bspecial2 + (X1 * Wspecial3)+ (X2 * Wspecial4)
Yweak1 = Bweak1 + (X1 * Wweak1)+ (X2 * Wweak2)
Yweak2 = Bweak2 + (X1 * Wweak3)+ (X2 * Wweak4)
$yH1 = \begin{cases} 1, & jika\ Yheavy1 > \theta \\ 0, & jika\ -\theta \le Yheavy1 \le \theta \\ -1, & jika\ Yheavy1 < -\theta \end{cases}$
A
A
yH1 == 1
yH2 == 1
true
Heavy Attack
False
yS1 == 1
yS2 == 1
Special Attack

true

False

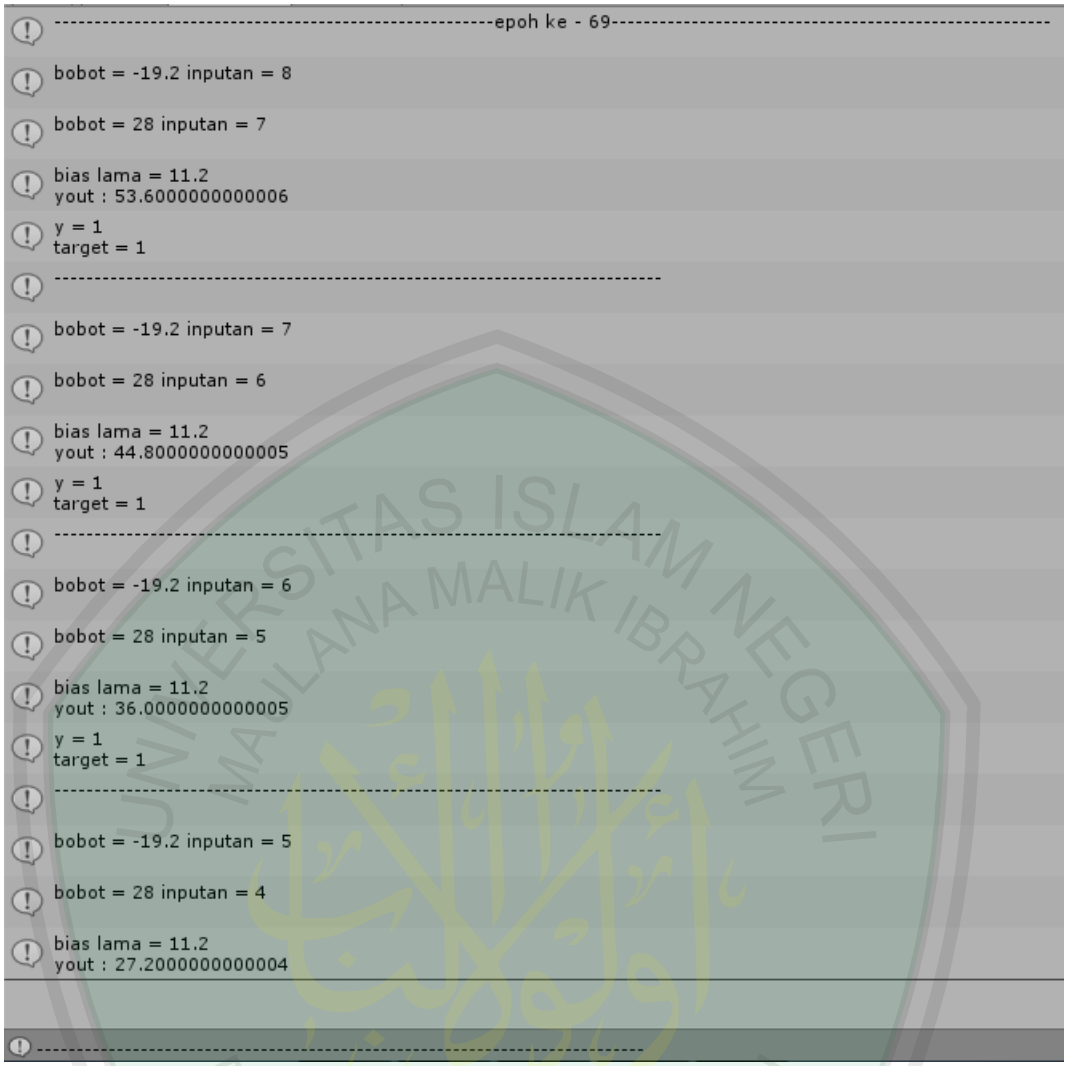
Gambar 4. 3 Flowchart klasifikasi process dalam perceptron.

Untuk dapat melakukan proses klasifikasi, pertama-tama dibutuhkan input yang akan diklasifikasikan. X1 adalah *player HP* sedangkan X2 adalah *NPC HP*. Setelah mendapatkan X1 dan X2 langkah berikutnya adalah menentukan nilai bobot dan nilai bias nya. Dalam pengimplementasian ke dalam game pembelajaran malware ini dibutuhkan enam set bobot dan bias.

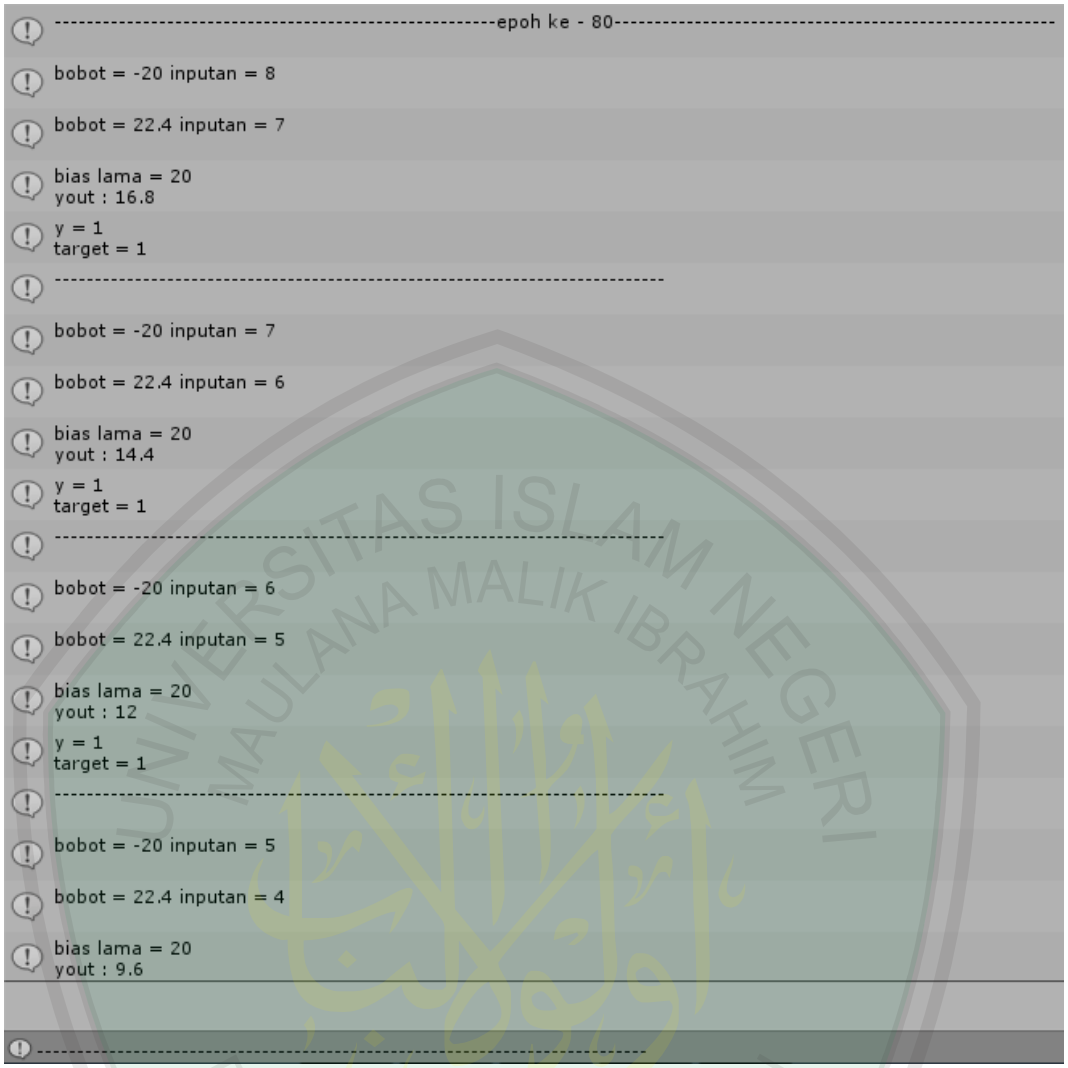
Nilai Wheavy1, Wheavy2, dan Bheavy1 diperoleh dari hasil training antara *heavy attack* dengan *special attack.* Nilai Wheavy3, Wheavy4, dan Bheavy2 diperoleh dari hasil training antara *heavy attack* dengan *weak attack.* Nilai Wspecial1, Wspecial2, dan Bspecial1 diperoleh dari hasil training antara *special attack* dengan *weak attack.* Nilai Wspecial3, Wspecial4, dan Bspecial2 diperoleh dari hasil training antara *special attack* dengan *heavy attack.* Nilai Wweak1, Wweak2, dan Bweak1 diperoleh dari hasil training antara *weak attack* dengan *special attack.* Nilai

Wweak3, Wweak4, dan Bweak2 diperoleh dari hasil training antara *weak attack* dengan *heavy attack.* Setelah semua input data ditentukan, berikutnya adalah menghitung nilai hasil aktivasi menggunakan rumus yang sama seperti pada proses *training.* Langkah selanjutnya adalah membandingkan nilai tersebut dengan fungsi aktivasi, sehingga diperoleh enam output, yH1, yH2, yS1, yS2, yW1, yW2. Keenam output tersebut akan memiliki nilai -1 atau +1. yH1 dan yH2 adalah nilai dari *class heavy attack*. yS1 dan yS2 adalah nilai dari *class special attack*. yW1 dan yW2 adalah nilai dari *class weak attack*. Class yang memiliki nilai +1 yang paling banyak adalah class yang terpilih.

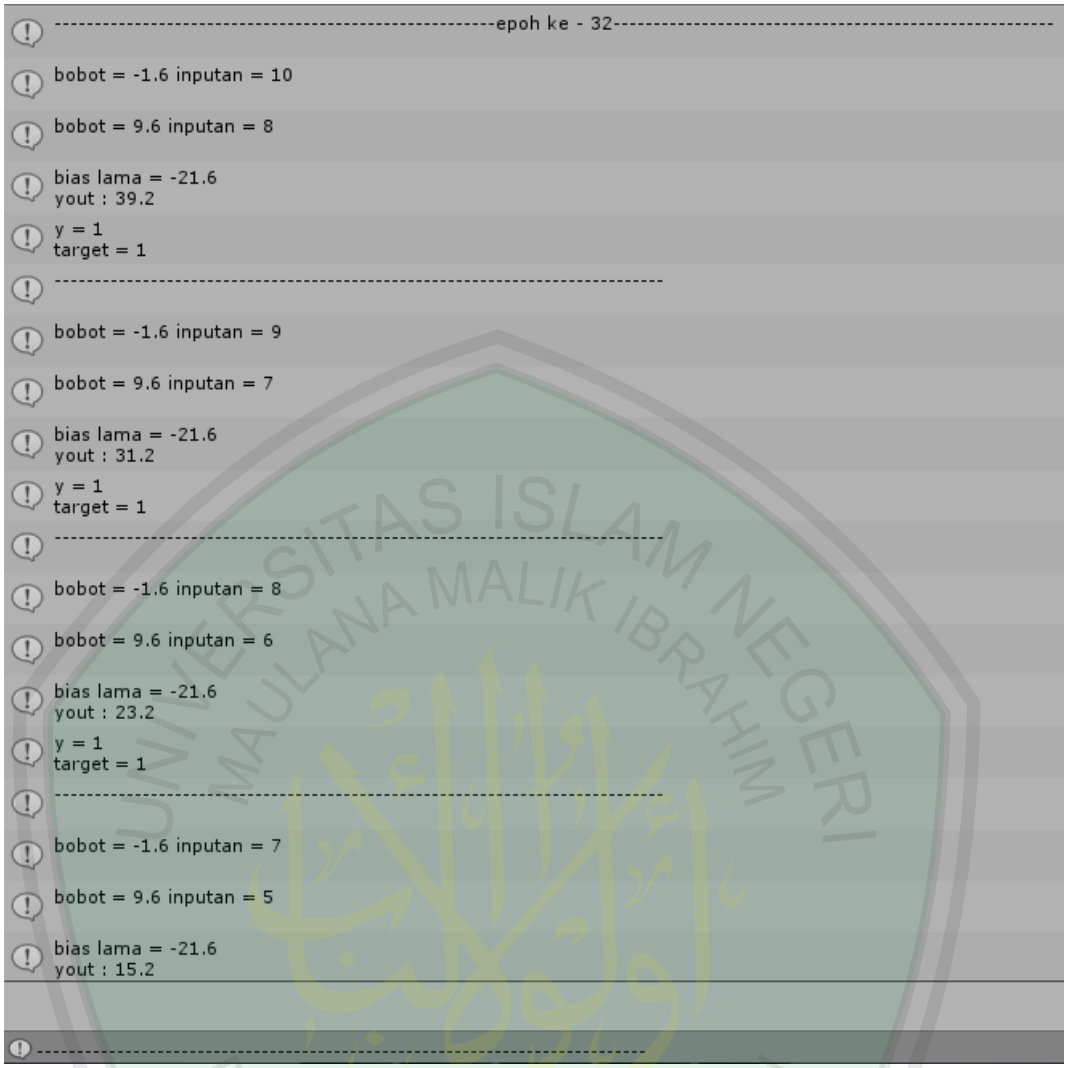
Berikut adalah output dari proses *training:*


----------epoh ke - 69----------
bobot = -19.2 inputan = 8
bobot = 28 inputan = 7
bias lama = 11.2
yout : 53.6000000000006
y = 1
target = 1
bobot = -19.2 inputan = 7
bobot = 28 inputan = 6
bias lama = 11.2
yout : 44.8000000000005
y = 1
target = 1
bobot = -19.2 inputan = 6
bobot = 28 inputan = 5
bias lama = 11.2
yout : 36.0000000000005
y = 1
target = 1
bobot = -19.2 inputan = 5
bobot = 28 inputan = 4
bias lama = 11.2
yout : 27.2000000000004

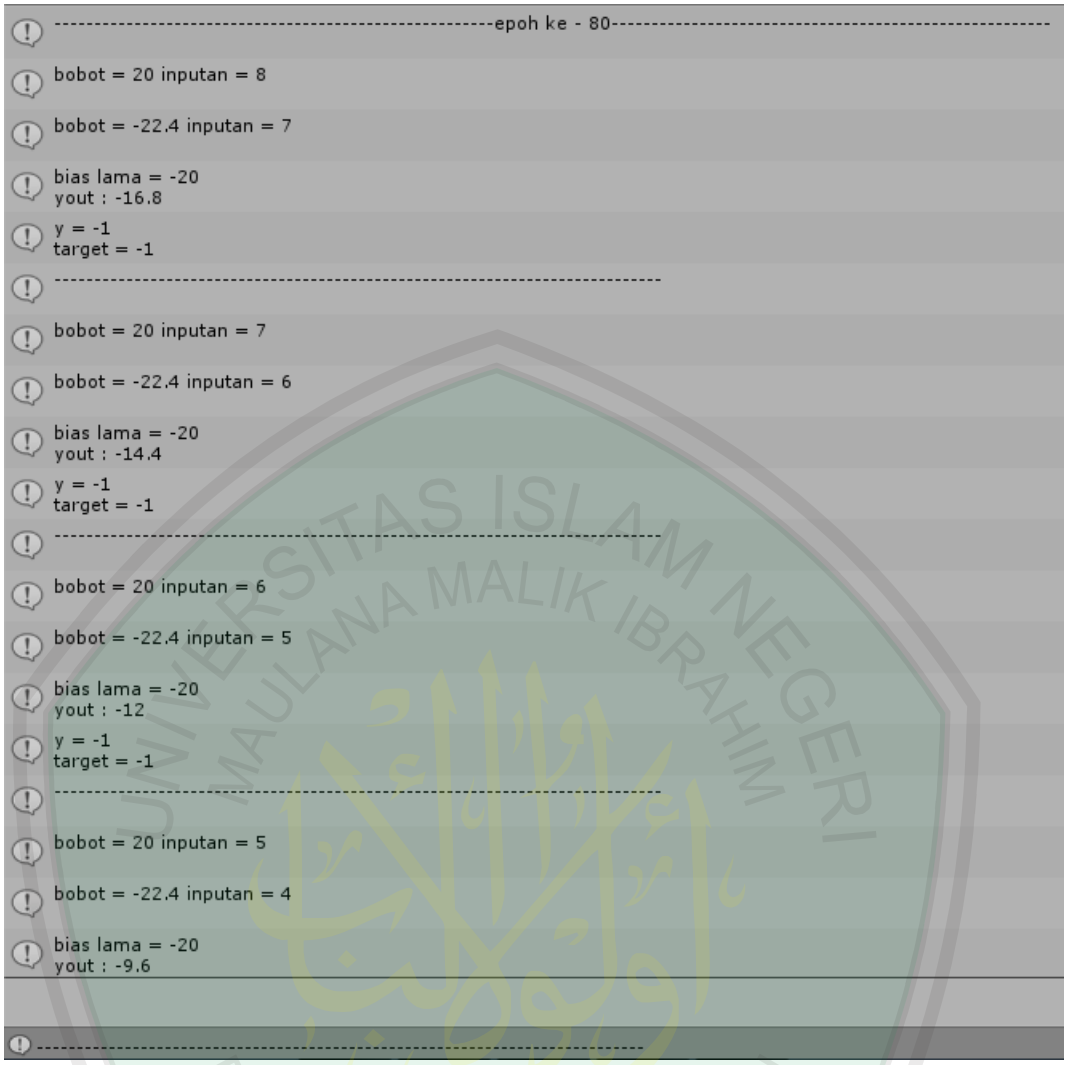
(a)



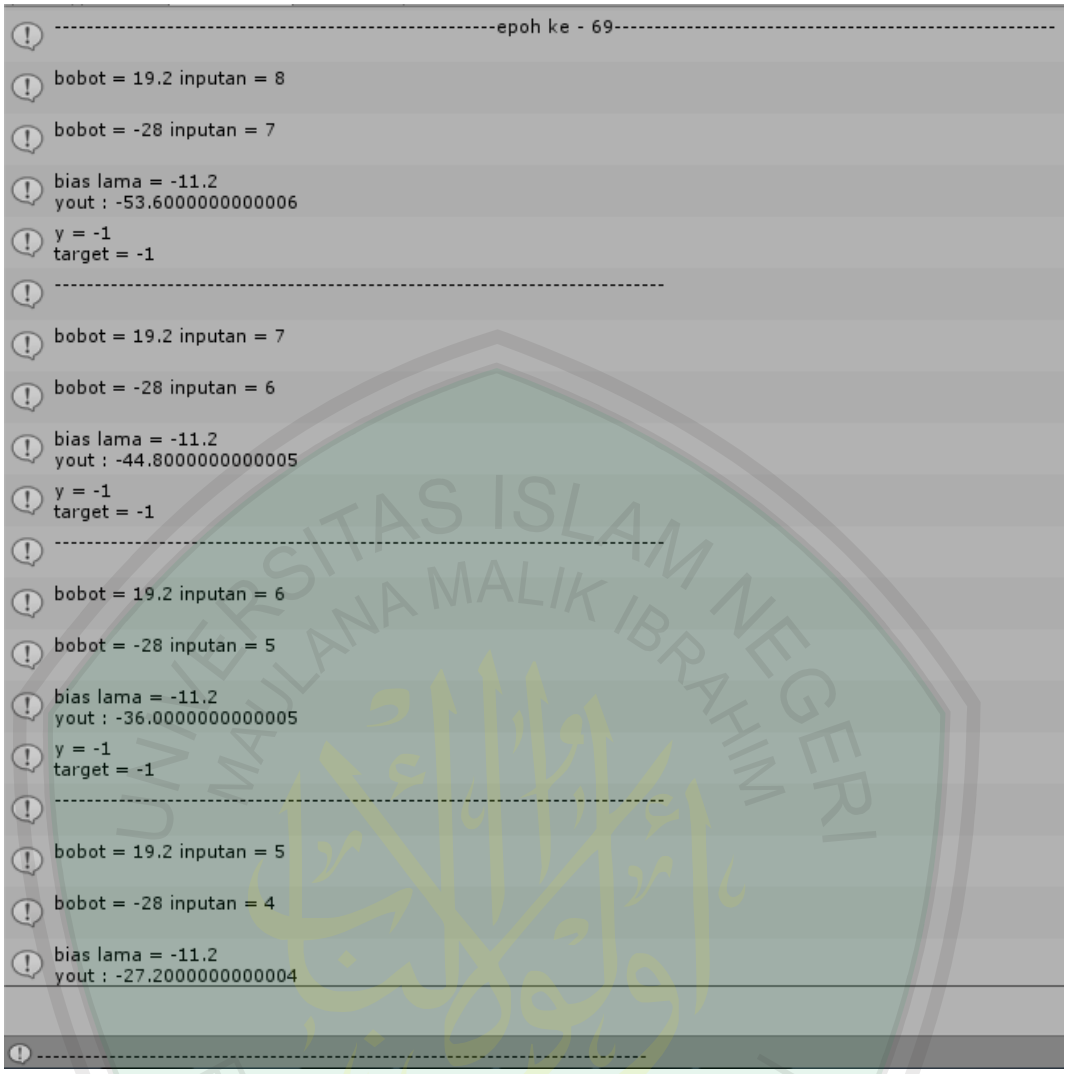
(b)



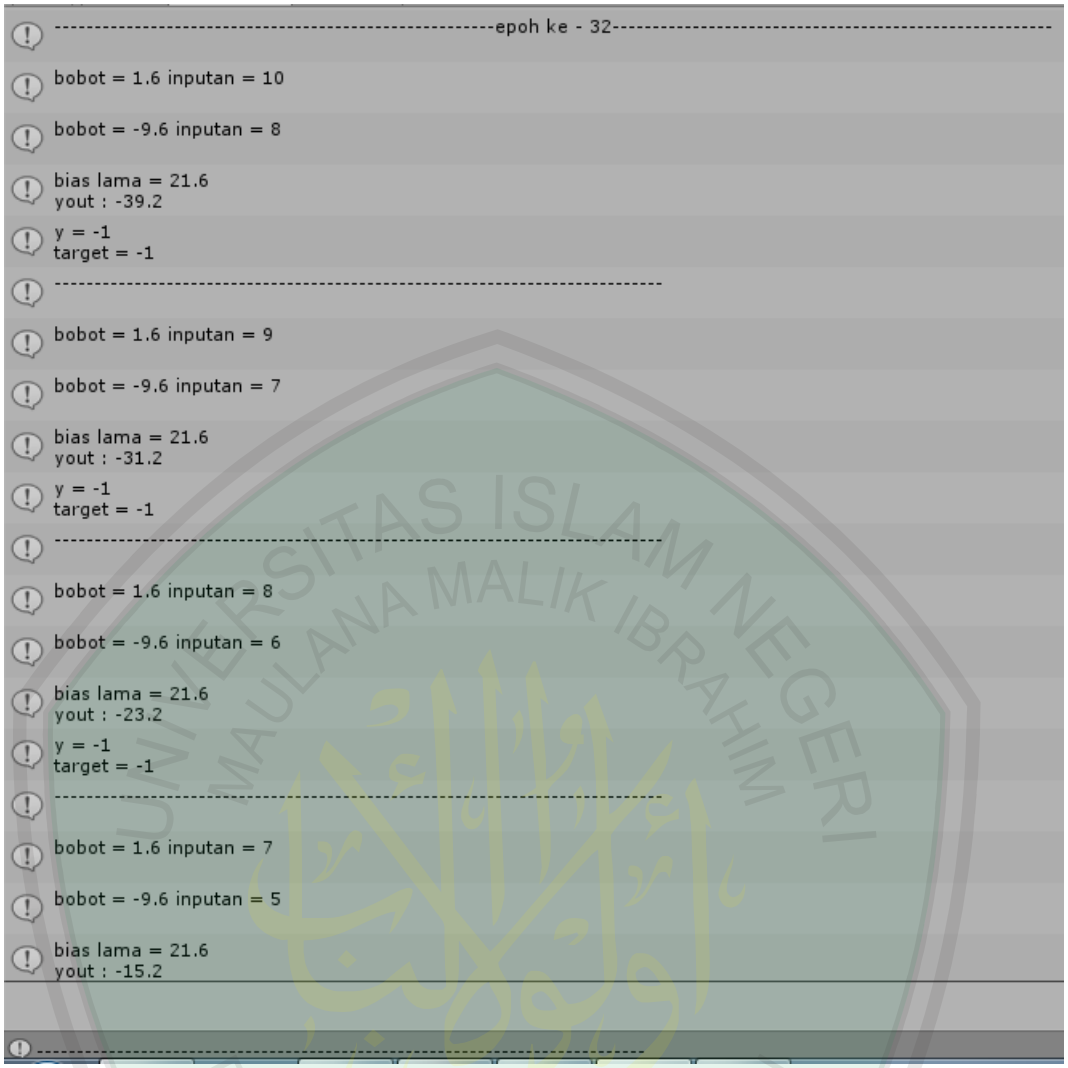
(c)



(d)


----------epoh ke - 69----------
bobot = 19.2 inputan = 8
bobot = -28 inputan = 7
bias lama = -11.2
yout : -53.6000000000006
y = -1
target = -1
bobot = 19.2 inputan = 7
bobot = -28 inputan = 6
bias lama = -11.2
yout : -44.8000000000005
y = -1
target = -1
bobot = 19.2 inputan = 6
bobot = -28 inputan = 5
bias lama = -11.2
yout : -36.0000000000005
y = -1
target = -1
bobot = 19.2 inputan = 5
bobot = -28 inputan = 4
bias lama = -11.2
yout : -27.2000000000004

(e)



(f)

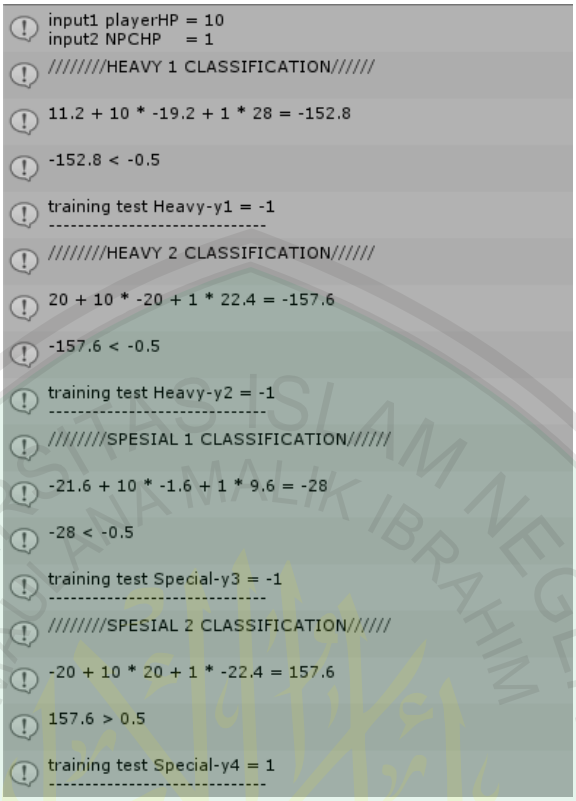
Gambar 4. 4 (a)heavy1, (b)heavy2, (c)spesial1, (d)spesial2, (e)weak1, (f)weak 2.

Gambar 4.5 menunjukkan *output* dari proses *training* pada masing-masing grup. Berikut adalah table hasil proses *training* pada masing-masing grup.

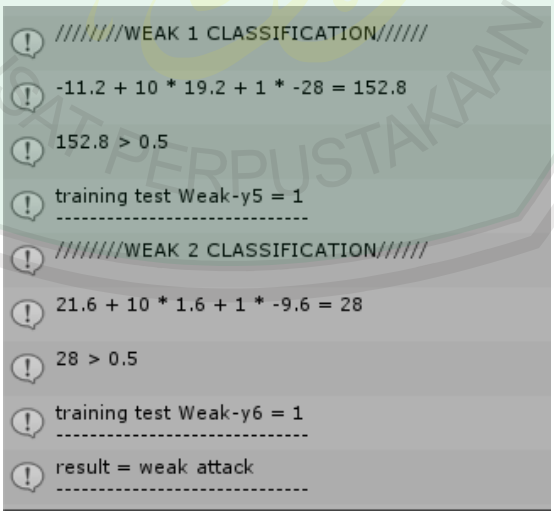
GROUP	BOBOT 1 AKHIR	BOBOT 2 AKHIR	BIAS AKHIR	EPOCH
Heavy1	-19.2	28	11.2	69 epoch
Heavy2	-20	22.4	20	80 epoch
Special1	-1.6	9.6	-21.6	32 epoch
Special2	20	-22.4	-20	80 epoch
Weak1	19.2	-28	-11.2	69 epoch
Weak2	1.6	-9.6	21.6	32 epoch

Tabel 4. 1 hasil dari proses training.

Berikut adalah output dari proses klasifikasi:



(a)

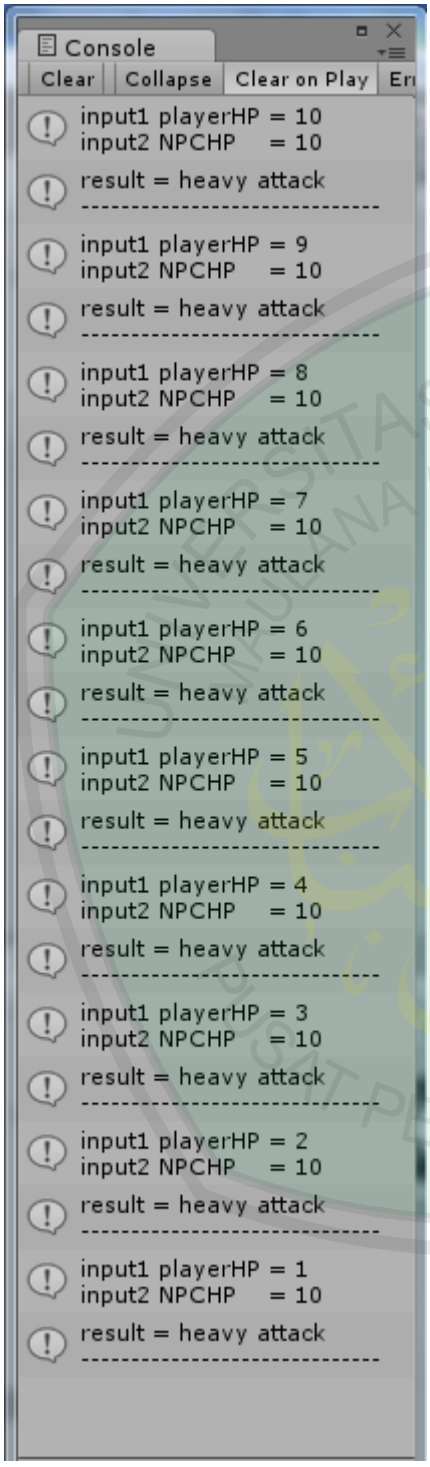


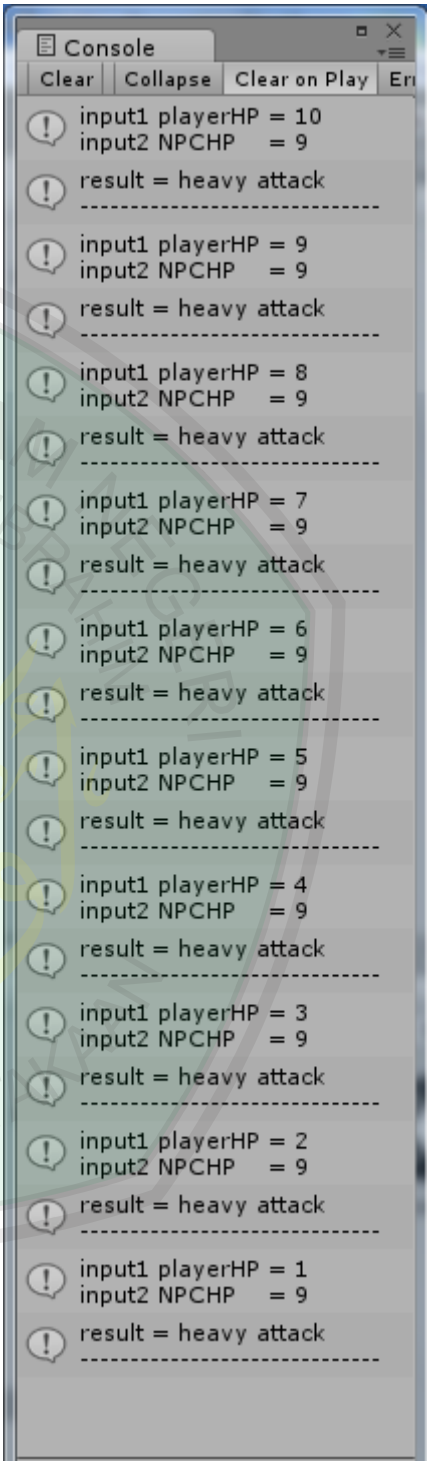
(b)

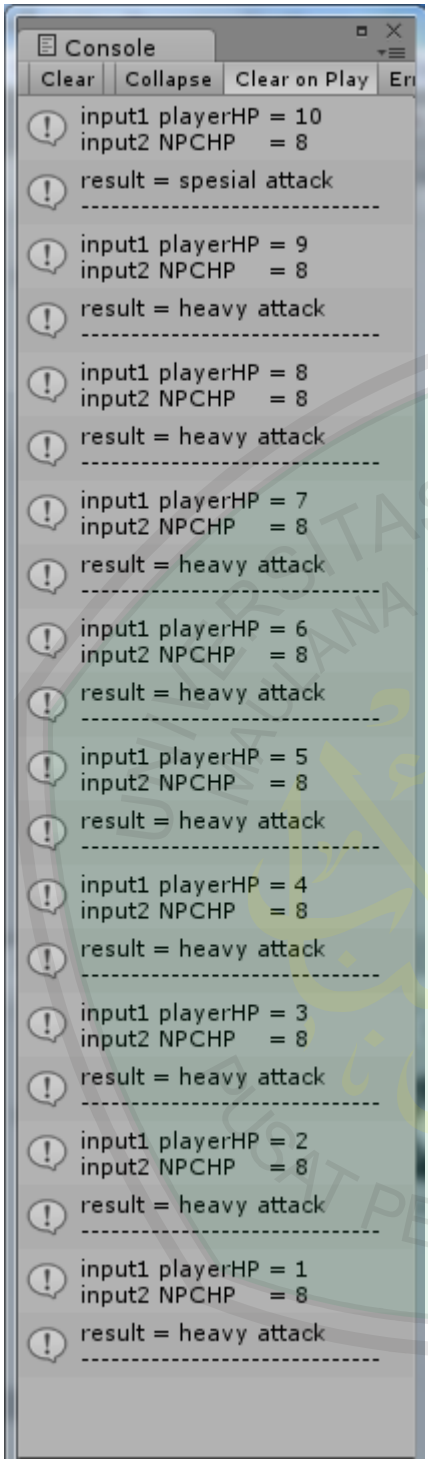
Gambar 4. 5 klasifikasi

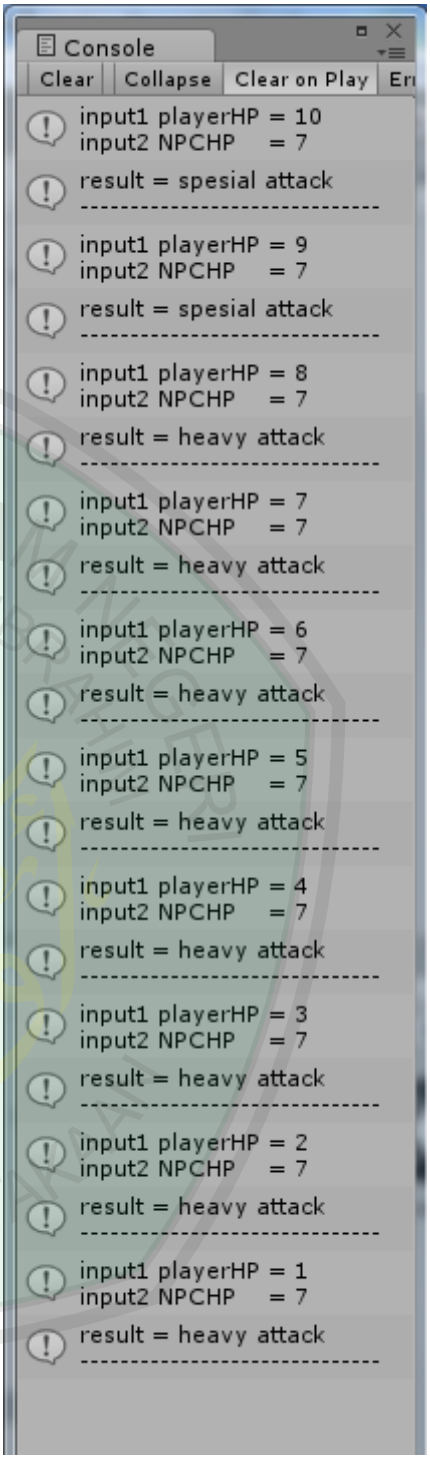
Gambar 4.6 menunjukkan proses klasifikasi untuk pemilihan serangan. Setelah mendapatkan bobot dan bias akhir untuk masing-masing group, langkah berikutnya adalah memasukkan nilai data input yang didapat selama permainan kedalam masing-masing persamaan group. Masing-masing persamaan akan menghasilkan nilai yang akan digunakan sebagai patokan pemilihan serangan. Seperti yang telah dibahas class yang memiliki nilai +1 lebih banyak adalah class yang terpilih. Masing-masing group mewakili suatu class. Pada gambar 4.6 nilai heavy-y1, nilai heavy-y2, special-y3, dan seterusnya adalah nilai output dari masing-masing group yang melambangkan suatu class. Heavy-y1 dan y2 adalah class heavy attack, special-y3 dan y4 adalah class special attack, dan weak-y5 dan y6 adalah class weak attack. Setelah memasukkan nilai input maka akan diperoleh nilai pada masing-masing group. Pada gambar 4.6 *class weak attack* memiliki nilai +1 lebih banyak dari pada class lain sehingga *weak attack* adalah *class* yang terpilih. Hal ini juga berlaku untuk *class* yang lain.

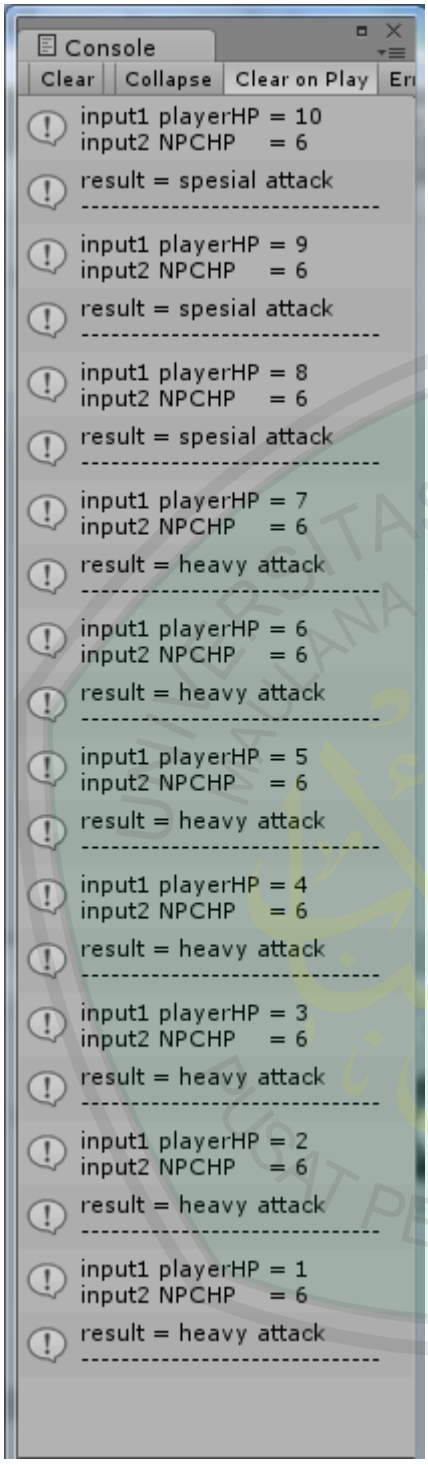
Berikut adalah hasil tes untuk keseluruhan data input:

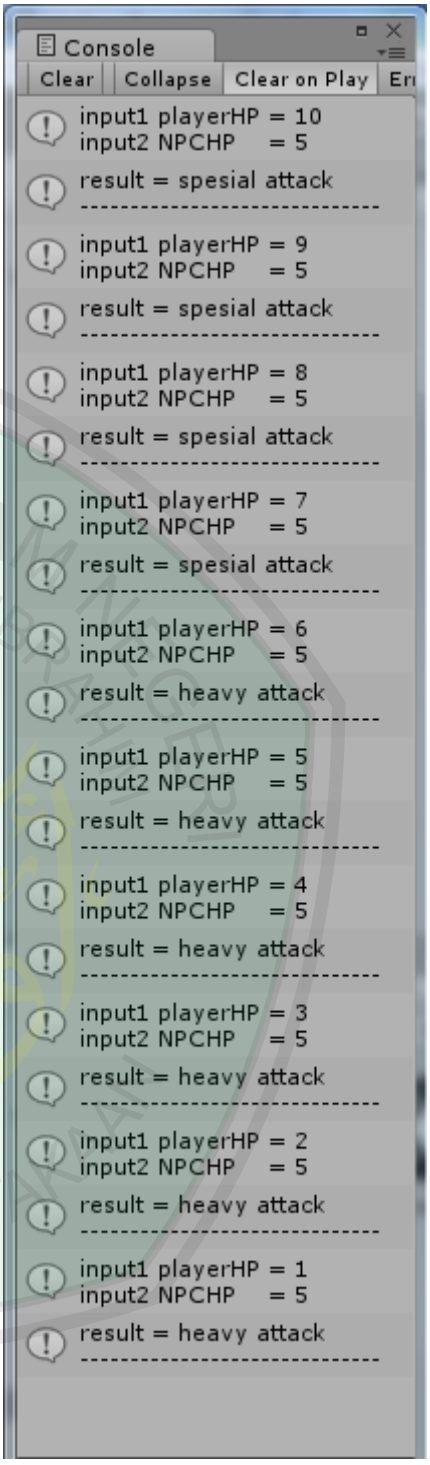


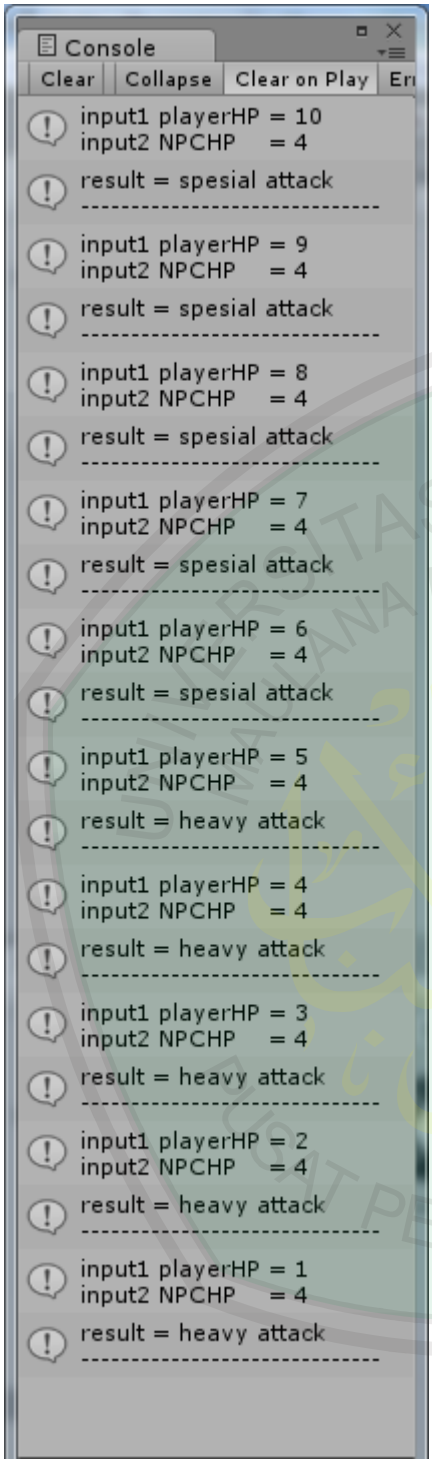


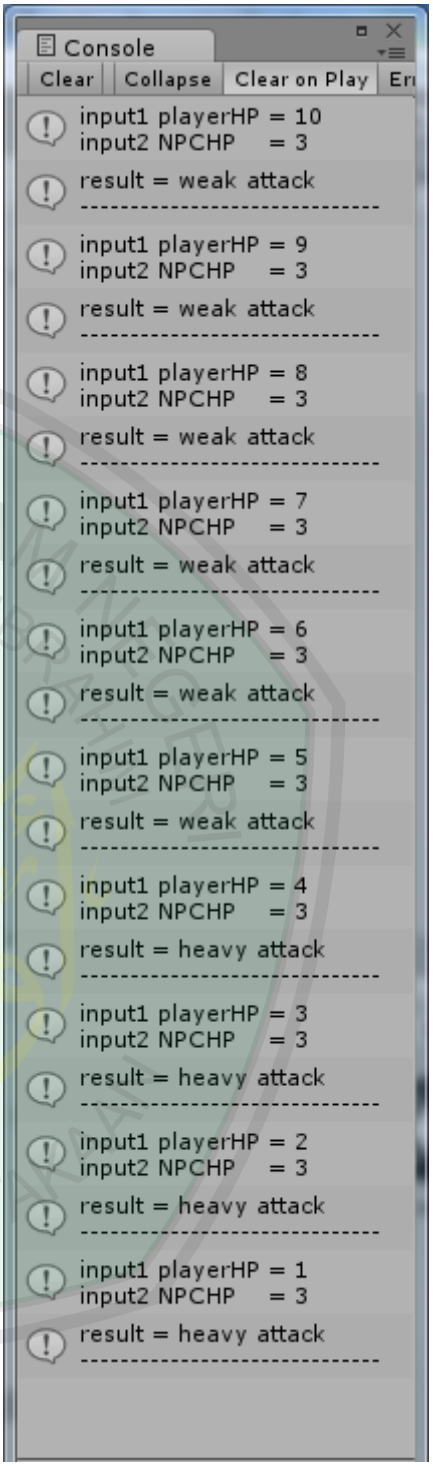

Console
Clear
Collapse
Clear on Play
input1 playerHP = 10
input2 NPCHP = 8
result = spesial attack
input1 playerHP = 9
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 8
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 7
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 6
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 5
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 4
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 3
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 2
input2 NPCHP = 8
result = heavy attack
input1 playerHP = 1
input2 NPCHP = 8
result = heavy attack

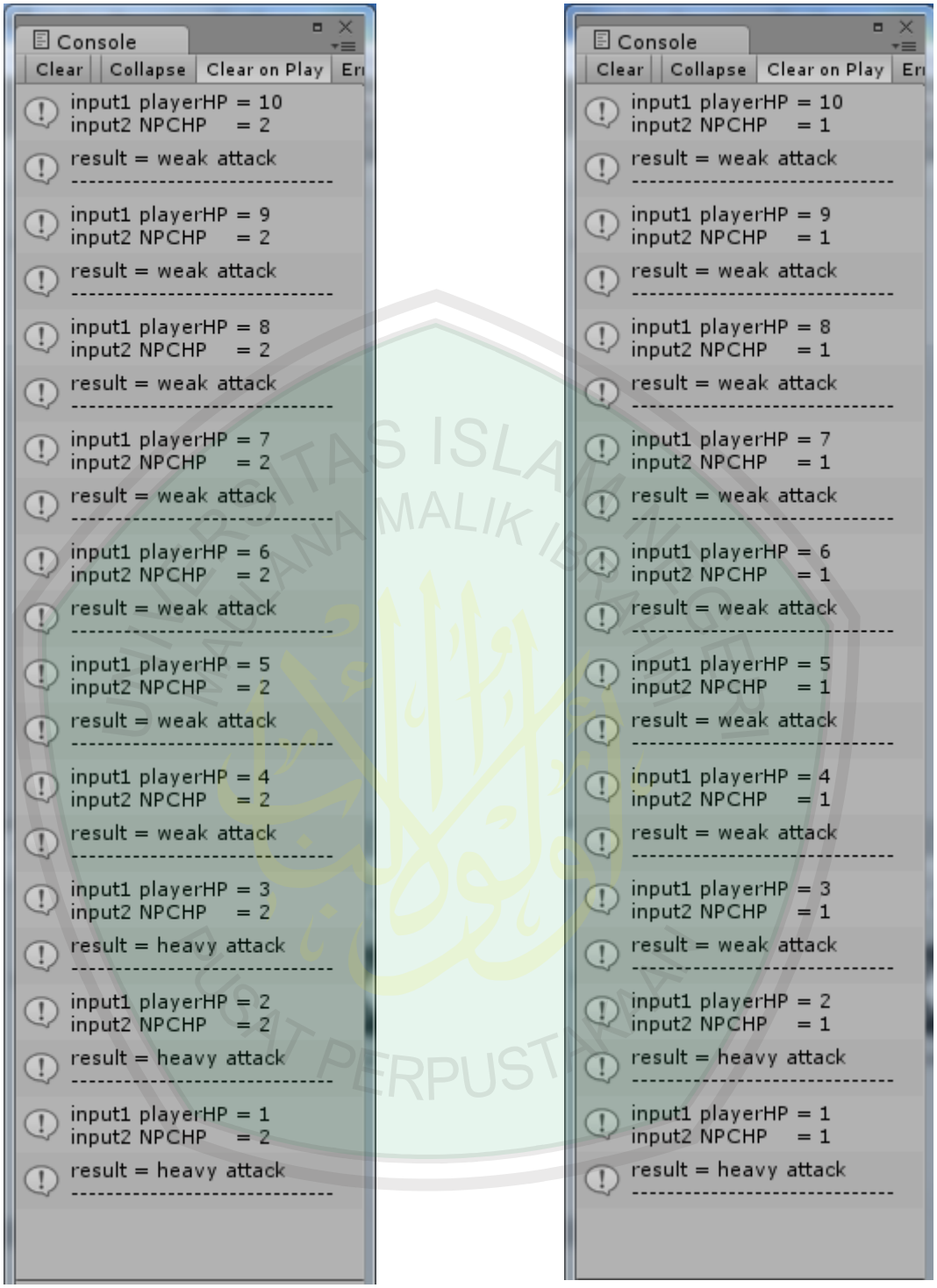

Console
Clear
Collapse
Clear on Play
input1 playerHP = 10
input2 NPCHP = 7
result = spesial attack
input1 playerHP = 9
input2 NPCHP = 7
result = spesial attack
input1 playerHP = 8
input2 NPCHP = 7
result = heavy attack
input1 playerHP = 7
input2 NPCHP = 7
result = heavy attack
input1 playerHP = 6
input2 NPCHP = 7
result = heavy attack
input1 playerHP = 5
input2 NPCHP = 7
result = heavy attack
input1 playerHP = 4
input2 NPCHP = 7
result = heavy attack
input1 playerHP = 3
input2 NPCHP = 7
result = heavy attack
input1 playerHP = 2
input2 NPCHP = 7
result = heavy attack
input1 playerHP = 1
input2 NPCHP = 7
result = heavy attack


Console
Clear
Collapse
Clear on Play
input1 playerHP = 10
input2 NPCHP = 6
result = spesial attack
input1 playerHP = 9
input2 NPCHP = 6
result = spesial attack
input1 playerHP = 8
input2 NPCHP = 6
result = spesial attack
input1 playerHP = 7
input2 NPCHP = 6
result = heavy attack
input1 playerHP = 6
input2 NPCHP = 6
result = heavy attack
input1 playerHP = 5
input2 NPCHP = 6
result = heavy attack
input1 playerHP = 4
input2 NPCHP = 6
result = heavy attack
input1 playerHP = 3
input2 NPCHP = 6
result = heavy attack
input1 playerHP = 2
input2 NPCHP = 6
result = heavy attack
input1 playerHP = 1
input2 NPCHP = 6
result = heavy attack


Console
Clear
Collapse
Clear on Play
input1 playerHP = 10
input2 NPCHP = 5
result = spesial attack
input1 playerHP = 9
input2 NPCHP = 5
result = spesial attack
input1 playerHP = 8
input2 NPCHP = 5
result = spesial attack
input1 playerHP = 7
input2 NPCHP = 5
result = spesial attack
input1 playerHP = 6
input2 NPCHP = 5
result = heavy attack
input1 playerHP = 5
input2 NPCHP = 5
result = heavy attack
input1 playerHP = 4
input2 NPCHP = 5
result = heavy attack
input1 playerHP = 3
input2 NPCHP = 5
result = heavy attack
input1 playerHP = 2
input2 NPCHP = 5
result = heavy attack
input1 playerHP = 1
input2 NPCHP = 5
result = heavy attack


Console
Clear
Collapse
Clear on Play
input1 playerHP = 10
input2 NPCHP = 4
result = spesial attack
input1 playerHP = 9
input2 NPCHP = 4
result = spesial attack
input1 playerHP = 8
input2 NPCHP = 4
result = spesial attack
input1 playerHP = 7
input2 NPCHP = 4
result = spesial attack
input1 playerHP = 6
input2 NPCHP = 4
result = spesial attack
input1 playerHP = 5
input2 NPCHP = 4
result = heavy attack
input1 playerHP = 4
input2 NPCHP = 4
result = heavy attack
input1 playerHP = 3
input2 NPCHP = 4
result = heavy attack
input1 playerHP = 2
input2 NPCHP = 4
result = heavy attack
input1 playerHP = 1
input2 NPCHP = 4
result = heavy attack


Console
Clear
Collapse
Clear on Play
input1 playerHP = 10
input2 NPCHP = 3
result = weak attack
input1 playerHP = 9
input2 NPCHP = 3
result = weak attack
input1 playerHP = 8
input2 NPCHP = 3
result = weak attack
input1 playerHP = 7
input2 NPCHP = 3
result = weak attack
input1 playerHP = 6
input2 NPCHP = 3
result = weak attack
input1 playerHP = 5
input2 NPCHP = 3
result = weak attack
input1 playerHP = 4
input2 NPCHP = 3
result = heavy attack
input1 playerHP = 3
input2 NPCHP = 3
result = heavy attack
input1 playerHP = 2
input2 NPCHP = 3
result = heavy attack
input1 playerHP = 1
input2 NPCHP = 3
result = heavy attack

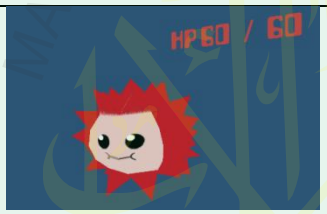
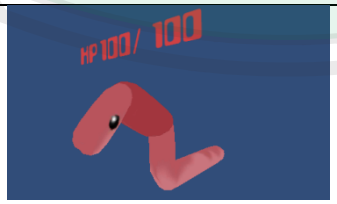
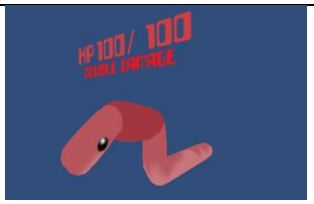


Gambar 4. 6 Hasil klasifikasi perceptron pada keseluruhan data.

4.2 Hasil Akhir Game

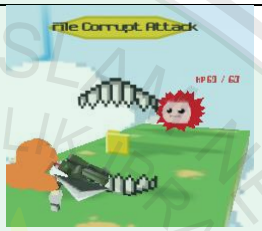
Dalam permainan nanti, akan ada empat *NPC malware* yang akan dihadapi *player*. *NPC* ini memiliki serangan dan *behavior* yang berbeda-beda. Berikut adalah detail dari *NPC Malware*:

Tabel data *NPC:*

Nama	Gambar	Behavior	Gambar behavior
Adware		-	-
Virus		Kekuatan serangan meningkat saat *HP* kurang dari 50	
Trojan		Kekuatan pertahanan meningkat saat *HP* kurang dari 50	
Worm		*Damage* menjadi dua kali lipat saat *HP* kurang dari 50	

Tabel 4. 2 daftar NPC Malware beserta behaviornya.

Tabel serangan *NPC:*

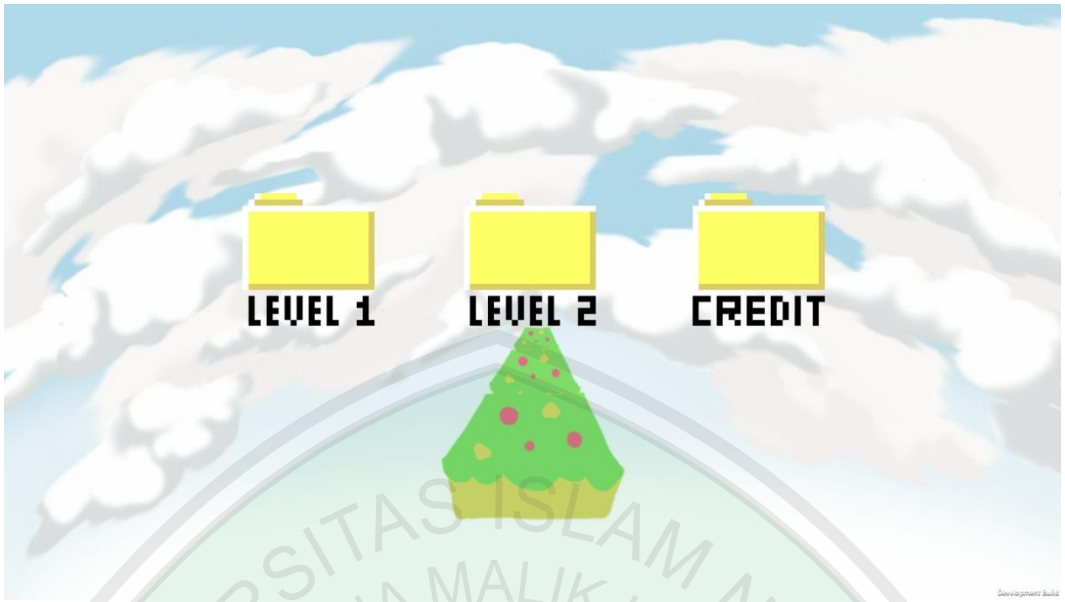
Nama	Heavy Attack	Special Attack	Weak Attack
Adware	 Multi pop-up attack	 Spyware attack	 Pop-up attack
Virus	 File hidden attack	 File corrupt attack	 File shortcut attack
Trojan	 BSOD attack	 PC control attack	 Steal Data attack
Worm	 Flame PC attack	 Payload PC attack	 Slow PC attack

Tabel 4. 3 daftar serangan NPC Malware.

Berikut adalah tampilan dari game pembelajaran malware:

Gambar 4. 7 main menu

Main menu adalah menu utama, menu yang pertama kali akan ditemui *player* ketika menjalankan permainan. *Player* hanya perlu men-*tap* layar *smartphonenya* untuk masuk ke menu selanjutnya.



Gambar 4. 8 stage select menu

Pada *stage select menu, player* dapat memilih *level stage* yang ingin dimainkan. Masing-masing *level stage* memiliki *boss NPC* yang berbeda untuk dikalahkan. *Player* cukup menekan *level stage* yang hendak dimainkan. Setelah memilih *level stage* yang ingin dimainkan berikutnya *player* akan masuk ke dalam *stage* yang telah dipilih.


OBJECTIVE:
• eliminate malware boss
• reach the end of
platform
JUMP

(a)

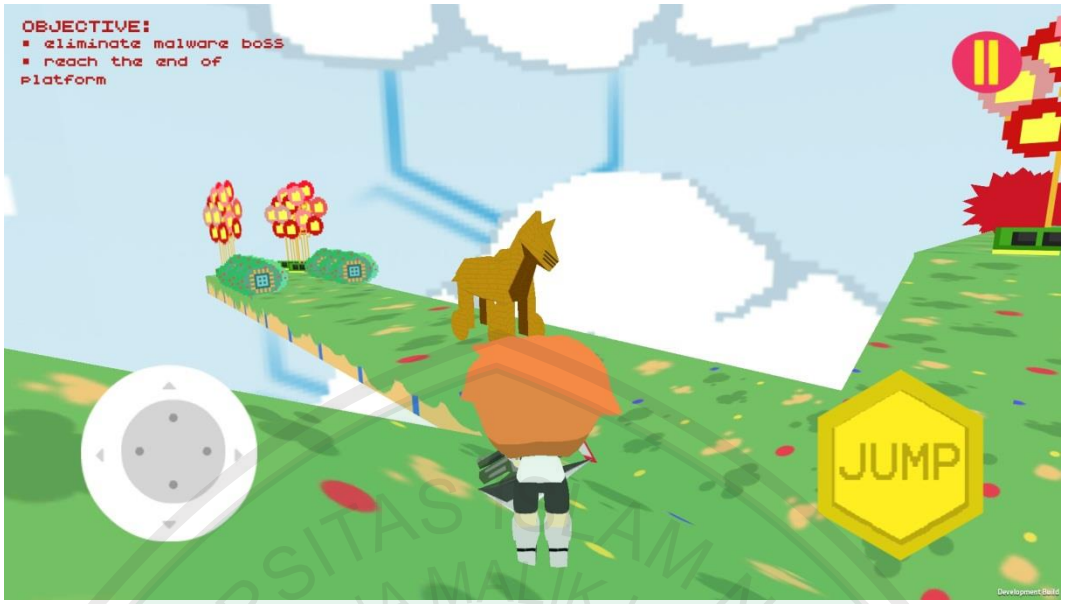

OBJECTIVE:
• eliminate malware boss
• reach the end of
platform
JUMP

(b)

(c)

Gambar 4. 9 (a) stage level 1, (b) stage level 2, (c) stage credit

Gambar 4.10 (a) dan (b) adalah stage dimana *player* akan bertarung melawan malware-malware selama perjalanan dari awal *platform* sampai akhir *platform*. Terdapat *jump button* yang dapat player gunakan untuk menghindari pertarungan dengan cara melompati musuh atau berpindah platform. Gambar 4.10 (c) adalah *stage credit,* yang mana dalam *stage credit* tidak terdapat musuh yang akan dihadapi. *Stage credit* menunjukkan nama dari *developer* game ini.

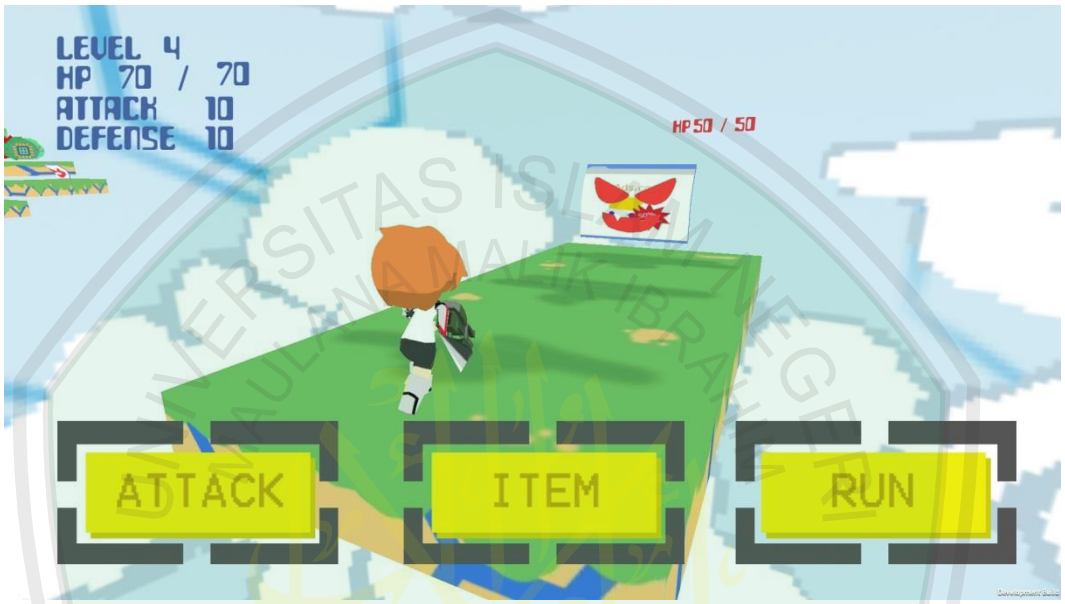


(a)

(b)

Gambar 4. 10 (a) stage level 1 boss NPC, (b) stage level 2 boss NPC

Pada gambar 4.11 (a) menunjukkan *boss NPC* pada *stage level 1* yaitu *Trojan*, sedangkan 4.11 (b) menunjukkan *boss NPC* pada *stage level 2* yaitu *Worm*. *Boss NPC* berada di ujung dari *platform* yang dilalui oleh *player*. *Player* tidak akan bisa lompat melewati *boss NPC*.

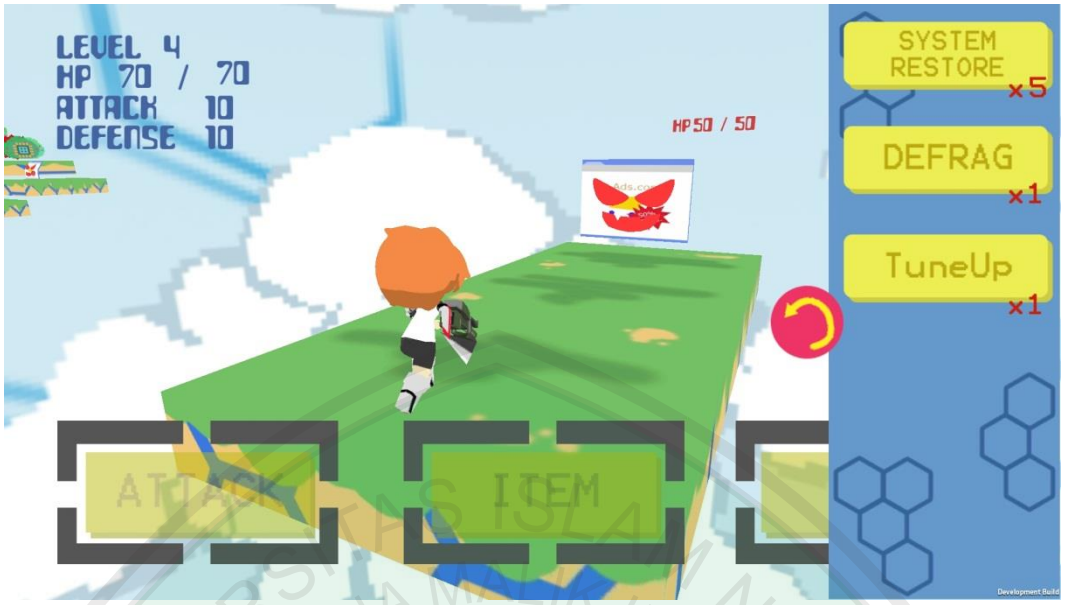


Gambar 4. 11 battle scene

Ketika melakukan kontak dengan *malware* saat berada di *platform*, maka *player* akan memasuki *battle scene*. Player akan bertarung menghadapi *malware* sampai salah satu diantaranya kalah. Dalam *battle scene* terdapat tiga *button* yang dapat digunakan oleh *player*, yaitu *attack*, *item*, dan *run*. *Button* ini adalah aksi yang akan dilakukan oleh *player*. Di sisi pojok kiri terdapat *player status* yang berisikan *level player*, *HP player*, *player power/attack*, dan *player defense*. Pada status *player HP*, nominal yang berada di sisi kanan adalah jumlah maksimum *HP* yang dapat dimiliki oleh *player*. Nominal yang berada di sisi kiri adalah jumlah *HP* yang dimiliki oleh *player* saat ini. *Player* juga dapat memantau *HP* dari malware yang dilawan.

Gambar 4. 12 battle scene attack selected

Saat *player* memilih aksi *attack*, maka akan muncul pilihan serangan yang dapat dilancarkan oleh player. Salah satu serangan yang terdapat pada pilihan tersebut merupakan kelemahan dari malware yang sedang dihadapi oleh *player*. Selain itu terdapat tombol *back* jika *player* ingin berganti menggunakan aksi yang lain.



Gambar 4. 13 battle scene item selected

Saat *player* memilih aksi *item* maka akan muncul pilihan *item* yang dapat digunakan oleh *player* seperti pada gambar 4.14. Dipojok kanan bawah *button* masing-masing *item* terdapat sebuah nominal yang melambangkan jumlah maksimum penggunaan *item.* Setiap *item* memberikan efek yang berbeda-beda kepada *player*.

Gambar 4. 14 Attack the weak spot

Ketika *player* melancarkan serangan kepada musuh, dan serangan tersebut adalah kelemahan dari musuh tersebut makan efek *weak* akan muncul. Ketika musuh menerima serangan yang merupakan kelemahannya, maka musuh tersebut akan menerima *damage* dua kali lipat lebih banyak. Dengan mengetahui kelemahan musuh dalam game diharapkan pengguna dapat mengetahui cara membasmi dan mencegah *malware* yang menyerang di kehidupan nyata.

Gambar 4. 15 Enemy terminated

Dalam *battle scene* ketika *player* membuat *NPC HP* menjadi nol maka *player* telah memenangkan pertarungan dan akan muncul *text enemy terminated*. Ketika berhasil mengalahkan musuh maka *player* akan mendapatkan kenaikan *level player*, maksimum *HP*, *attack*, dan juga *defense*. Maksimum *HP* yang dapat dimilik oleh *player* adalah 100 point. Maksimum *attack* dan *defense* yang dapat dimiliki oleh *player* adalah 30 point, dan maksimum *level player* yang dapat dicapai adalah 24 point. Sangat disarankan untuk meningkatkan *level player* terlebih dahulu sebelum melawan *boss NPC*, karena *boss NPC* memiliki attack, defense, dan maksimum *HP* yang sangat tinggi. Setelah berhasil mengalahkan musuh *player* akan dikembalikan ke dalam platform untuk melanjutkan perjalanan menuju ujung platform.

Gambar 4. 16 Game over

Dalam *battle scene* ketika *NPC* membuat *player HP* menjadi nol maka *text game over* akan muncul dan pengguna akan dibawa kembali ke *level stage select menu*.

Gambar 4. 17 Finish Line

Setelah *player* mengalahkan *boss NPC,* selanjutnya *player* akan mencapai ujung dari *platform.* Saat *player* berada di ujung *platform, text finish* akan muncul dan *jump button* berubah menjadi *return button*. ketika pengguna men-*tap return button* pengguna akan dibawa kembali ke *level stage select menu*.

4.3 Pengujian Permainan

Berikut ini adalah table pengujian aplikasi pada perangkat *mobile smartphone*:

No	Model Smartphone	Versi Android	CPU	RAM	Keterangan	Presentase
1	Lenovo A7000-a	5.0.2 (lollipop)	8 core, 1.7GHz	2.00 GB	Sistem berjalan dengan baik	100%
2	Samsung Galaxy Tab 2	4.1.2 (jellybean)	Dual core, 1GHz	1.00 GB	Sistem berjalan dengan baik	100%
3	Xiaomi Redmi 2	4.4.4 (kitkat)	Quad-core, 1.2GHz	1.00 GB	Sistem berjalan dengan baik	100%
4	Samsung Galaxy A5	5.1.1 (lollipop)	Quad-core, 1.2GHz	2.00 GB	Sistem berjalan dengan baik	100%
5	Lenovo S850	4.2.0 (jellybean)	Quad-core, 1.3GHz	1.00 GB	Sistem berjalan dengan baik	100%

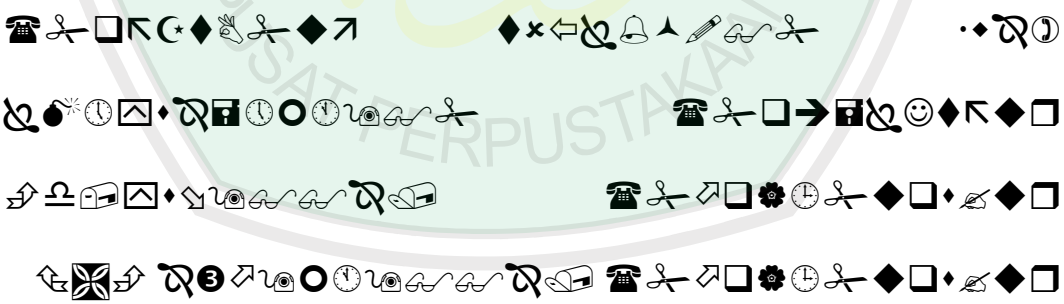
Tabel 4. 4 Pengujian pada perangkat mobile smartphone.

Pengujian pada perangkat mobile menunjukkan hasil sistem dapat berjalan dengan baik. Tampilan dan fungsi tombol berjalan sesuai dengan harapan.

4.3 Integrasi dengan Islam

Hampir semua aspek di kehidupan manusia melibatkan teknologi didalamnya. Salah satu teknologi yg sangat berpengaruh dalam kehidupan manusia adalah smartphone. Kehadiran teknologi ini telah mengubah gaya hidup manusia masa kini. Manusia tidak pernah bisa jauh dari smatphone milik mereka, karena kehadiran teknologi ini sangat membantu dalam berbagai sektor kehidupan manusia. Namun dibalik itu semua terdapat ancaman berbahaya yang mengancam data serta informasi penting yang terdapat di dalam smartphone, yaitu serangan malware. Malware-malware ini dapat merusak dan bahkan mencuri data pribadi dan informasi penting yang disimpan dalam smartphone. Akan dari itu pengetahuan akan malware ini sangatlah penting dalam rangka melindungi smartphone milik kita. Namun kurangnya pengetahuan masyarakat akan bahaya dari malware adalah masalah yang lain.

Dalam Al-quran surat Al-Ashr ayat 3:



3. kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran.

Surat Al-Ashr menceritakan tentang golongan manusia yang merugi, kecuali orang-orang yang beriman dan beramal shaleh. Dalam tafsirnya ibnu katsir menjelaskan maka dikecualikan dari jenis manusia yang terhindar dari kerugian, yaitu orang-orang yang beriman hatinya dan anggota tubuhnya mengerjakan amal-amal yang shaleh. Yakni menunaikan dan meninggalkan semua yang diharamkan. Yaitu tabah menghadapi musibah dan malapetaka serta gangguan yang menyakitkan dari orang-orang yang ia perintah melakukan kebajikan dan ia larang melakukan kemungkaran(ibnukatsironline,2015).

Ayat tersebut memberikan motivasi dan inspirasi bagi penulis untuk mengamalkan ilmunya. Setelah belajar banyak hal dalam bidang teknik informatika tentunya akan menjadi orang yang rugi bila penulis tidak dapat mengamalkan ilmunya atau mengajarkan ilmu yang diperolehnya. Ilmu yang penulis peroleh akan menjadi lebih bermanfaat bila orang lain juga dapat merasakan manfaat dari ilmu tersebut. Berhubungan dengan permasalahan tentang kurangnya pengetahuan masyarakat tentang malware, penulis ingin membagi atau mengajarkan ilmu yang dimilikinya tentang bagaimana cara menangani dan mencegah serangan dari malware.

Dalam teknik informatika terdapat satu bidang yang disebut multimedia. Salah satu wujud dari multimedia tersebut adalah permainan video, atau *video game*. Kita dapat membuat *video game* menjadi sebuah media yang lebih dari sekedar media hiburan, melainkan media yang membawa pendidikan dan pelajaran bagi siapapun yang memainkannya.

Dengan memanfaatkan *video game* kita dapat membuat media pembelajaran berbasis *game* yang mempelajari mengenai *malware* cara pencegahan, cara menghilangkan, dan macam-macam serangan *malware*. Dengan memanfaatkan *video game* kita dapat menarik minat masyarakat untuk belajar karena dengan bermain secara tidak langsung mereka juga belajar dan mereka juga merasa senang memainkannya.

BAB V

PENUTUP

5.1 Kesimpulan

Algoritma Perceptron dapat diimplementasikan ke dalam aplikasi *game* pembelajaran malware ini. Dengan *algoritma perceptron* ini, *NPC* dapat memilih serangan yang akan dilancarkan kepada *player* berdasarkan input *HP* yang dimiliki oleh *NPC* dan *player*. NPC mampu memilih satu dari tiga pilihan serangan yang ada yaitu *heavy attack, special attack*, dan *weak attack*. Semua uji coba aplikasi pada perangkat mobile menunjukkan hasil aplikasi dapat berjalan dengan baik dan berjalan sesuai dengan yang diinginkan.

5.2 Saran

Masih banyak kekurangan dan hal yang dapat dikembangkan dalam game ini di antaranya sebagi berikut:

1. Memberikan informasi lengkap dan tutorial cara memainkan game ini.
2. Memberikan informasi yang lebih lengkap pada malware yang akan dilawan, agar pengguna dapat mempelajarinya lebih jauh.
3. Menambahkan sound.
4. Mengembangkan lebih jauh *AI* dari *NPC*.
5. Menambahkan variasi *NPC*.

DAFTAR PUSTAKA

Kusumadewi, Sri. 2003. Artificial Intelligence(Teknik dan Aplikasinya). Yogyakarta: Graha Ilmu.

Murya, Yosef. 2014. Android Black Box:JasaKom.

Hermawan, Stephanus. 2011. Mudah Membuat Aplikasi Android.Yogyakarta: Penerbit Andi.

Sutojo, T., Mulyanto, E., Suhartono, V. (2011). Kecerdasan Buatan. Yogyakarta: Penerbit Andi.

Wahyuningsih, M.S. 2014. Penerapan Metode Perceptron Untuk Leveling Fitur Kuis Pada Aplikasi Pembelajaran Bahasa Korea. Universitas Islam Negeri Maulana Malik Ibrahim, Malang.

Susmikanti, Mike, Entin Hartini, dan Antonius Sitompul. 2007. Identifikasi Pengaruh Umur, Suhu Dan Radiasi Terhadap Strukturmikro Ferritic Steel Berbasis Kecerdasan Buatan. *Jurnal Sains Materi Indonesia Edisi Khusus Desember 2008, hal : 307 – 313,* Pusat Pengembagan Informasi Nuklir (PPIN) – BATAN Kawasan Puspiptek, Serpong 15314, Tangerang, Pusat Teknologi Bahan Industri Nuklir (PTBIN) – BATAN, Kawasan Puspiptek, Serpong 15314, Tangerang

Susmikanti, Mike, Arya Adhiyaksa. 2005. Identifikasi Huruf Menggunakan Metode Pembelajaran Perceptron Dalam Jaringan Neural. Prosiding Semiloka Teknologi Simulasi dan Komputasi serta Aplikasi 2005. Pusat Pengembangan Teknologi Informasi dan Komputasi – BATAN

Pujiyanta, Ardi. 2009. Pengenalan Citra Objek Sederhana Dengan Jaringan Saraf Tiruan Metode Perceptron. Jurnal Informatika Vol. 3, No. 1, Januari 2009. Universitas Ahmad Dahlan Yogyakarta.

Azmi, Zulfian. 2011. Aplikasi Jaringan Syaraf Tiruan Untuk Pengenalan Pola Pembukaan Permainan Catur. Jurnal SAINTIKOM Vol. 10 / No. 1 / Januari 2011. Universitas STMIK Triguna Dharma.

Brilliant, R.M.A. Analisis Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Perceptron Dalam Memprediksi Penyakit Jantung Koroner. Universitas Dian Nuswantoro. Semarang

Asri, Yessy. 2011. Penerapan Aturan Perceptron Pada Jaringan Saraf Tiruan Dalam Pengenalan Pola Penyakit Mata. Jurnal Pengkajian dan Penerapan Teknik Informatika Vol. 4/ No. 2/ September 2011. Sekolah Tinggi Teknik PLN.

Conversation, The. 2014. Video Games Are Good For Your Brain - Here's Why. Dikutip dari http://www.science20.com/the_conversation/video_games_are_good_for_your_brain_heres_why-149061 diakses pada tanggal 12/09/2015.

Purwanto, Eko. 2014. Cerita di Balik Software 3D Blender. Dikutip dari http://bpptik.kominfo.go.id/2014/05/12/419/cerita-di-balik-software-3d-blender diakses pada tanggal 12/09/2015.

Antara. 2013. Survei: 92 Persen Malware Serbu Android. Dikutip dari http://www.republika.co.id/berita/trendtek/aplikasi/13/06/26/mozw7d-survei-92-persen-malware-serbu-android diakses pada tanggal 12/09/2015.

Wijaya, Ketut Krisna. 2015. Android dan browser Opera dominasi pengguna mobile Indonesia selama 2014. Dikutip dari https://id.techinasia.com/android-opera-dominasi-smartphone-indonesia-2014 diakses pada tanggal 12/09/2015.

Azka, Rudi Abu. 2015. Tafsir Surat Al-'Asr, ayat 1-3. Dikutip dari http://www.ibnukatsironline.com/2015/10/tafsir-surat-al-asr-ayat-1-3.html diakses pada tanggal 26/06/2016.