

***GAME SHARRAF SOLITAIRE MENGGUNAKAN NEURAL
NETWORK BACKPROPAGATION UNTUK PENENTUAN
LEVEL PADA GAME PEMBELAJARAN TASHRIF
ISTHILAH DI DALAM ILMU SHARRAF***

SKRIPSI

oleh:

M. ARIF MUKHLISIN

NIM. 10650099



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

***GAME SHARRAF SOLITAIRE MENGGUNAKAN NEURAL
NETWORK BACKPROPAGATION UNTUK PENENTUAN
LEVEL PADA GAME PEMBELAJARAN TASHRIF
ISTHILAH DI DALAM ILMU SHARRAF***

SKRIPSI

oleh:
M.ARIF MUKHLISIN
NIM. 10650099



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

***GAME SHARRAF SOLITAIRE MENGGUNAKAN NEURAL
NETWORK BACKPROPAGATION UNTUK PENENTUAN
LEVEL PADA GAME PEMBELAJARAN TASHRIF
ISTHILAHI DALAM ILMU SHARRAF***

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:
M.ARIF MUKHLISIN
NIM. 10650099**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

LEMBAR PERSETUJUAN

***GAME SHARRAF SOLITAIRE MENGGUNAKAN NEURAL NETWORK
BACKPROPAGATION UNTUK PENENTUAN LEVEL PADA GAME
PEMBELAJARAN TASHRIF ISTHILAHY DALAM ILMU SHARRAF***

SKRIPSI**Oleh :**

Nama : M.Arif Mukkhlisin

NIM : 10650099

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji :

Tanggal :

Dosen Pembimbing I**Dosen Pembimbing II****Fresy Nugroho, M.T****Dr.H.Ahmad Barizi, M.A****NIP. 19710722 201101 1 001****NIP. 19731212 199803 1 001**

Mengetahui,

Ketua Jurusan Teknik Informatika**Dr. Cahyo Crysdiyan****NIP. 19740424 200901 1 008**

LEMBAR PENGESAHAN

***GAME SHARRAF SOLITAIRE MENGGUNAKAN NEURAL NETWORK
BACKPROPAGATION UNTUK PENENTUAN LEVEL PADA GAME
PEMBELAJARAN TASHRIF ISTHILAH I DALAM ILMU SHARRAF***

SKRIPSI

Oleh :

M.Arif Mukhlisin**NIM. 10650099**

Telah dipertahankan di depan dewan penguji skripsi
dan dinyatakan diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana komputer (S.Kom)

Tanggal : 11 Juli 2014

Susunan Dewan Penguji:**Tanda Tangan**

- | | | | |
|---------------------------|---|----------|----------|
| 1. Penguji Utama | : <u>Hani Nurhayati, M.T</u> | (|) |
| | NIP. 19780625 200801 2 006 | | |
| 2. Ketua Penguji | : <u>Fachrul Kurniawan, M.MT</u> | (|) |
| | NIP. 19771020 200901 1 001 | | |
| 3. Sekretaris | : <u>Fresy Nugroho, M.T</u> | (|) |
| | NIP. 19710722 201101 1 001 | | |
| 4. Anggota Penguji | : <u>Dr.H.Ahmad Barizi, M.A</u> | (|) |
| | NIP.19731212 199803 1 001 | | |

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian**NIP. 19740424 200901 1 008**

HALAMAN PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama : M.Arif Mukhlisin

NIM : 10650099

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : *Game Sharraf Solitaire Menggunakan Neural Network Backpropagation Untuk Penentuan Level Pada Game Pembelajaran Tashrif Isthilahi Dalam Ilmu Sharraf*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, 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, 07 Juli 2014
Yang membuat pernyataan,

M.Arif Mukhlisin
NIM. 10650099

MOTTO

*hablum minAllah,
hablum minannas,
hablum minal alam.*

PERSEMBAHAN

Segala Puji Syukur atas limpahan rahmat, taufik serta hidayah Allah SWT atas terselesaikannya skripsi ini. Tak lupa ucapan terimakasih sebanyak – banyaknya kepada :

- ❖ Ayah dan Ibu tercinta yang selalu memberi do'a, motivasi dan dukungan yang luar biasa untuk menyelesaikan studi di UIN Maliki.
- ❖ Saudariku Anita dan seluruh keluarga besar yang telah menyemangati dan memberikan dukungannya.
- ❖ Bapak Fressy Nugroho, M.T dan Bapak Dr.H.Ahmad Barizi M.A yang selalu membimbing dalam pengerjaan tugas akhir.
- ❖ Dosen-dosen di jurusan Teknik Informatika yang telah membimbing dan mendidik saya selama menjalani studi di jurusan Teknik Informatika UIN Maliki Malang.
- ❖ Ibu Dr.Hj.Mufidah Cholil M.Ag yang mengajarkan saya untuk jadi orang yang bermanfaat bagi orang lain.
- ❖ Sahabat-sahabat dekat seangkatan : Catur, Muiz, Wahyu, Maskal, Taufan, Arif, Agus, Bekti, Saiful, Saifudin, Sofyan, Afif, Acip, Zaenal, Fani, Asov, Ardi, Dita, Ade, Tita, Dewi, Puspa, dan seluruh INFINITY (TI 2010) yang sangat menginspirasi dan membantu dalam penyelesaian study S1.
- ❖ Teman-Teman Organisasi di IKAMARO (Ikatan Mahasiswa Bojonegoro) UIN Maliki, LMN (Liga Mahasiswa NasDem) Malang Raya, HMJ TI 2011 UIN Maliki, DEMA F Saintek 2013 UIN Maliki yang telah memberikan kesempatan untuk mengimplementasikan ilmu saya selama menjadi mahasiswa di UIN Maliki Malang.
- ❖ Teman – teman seperjuangan skripsi khususnya (Syafi'i, Saifudin, Rizky), terimakasih kerjasamanya telah memberikan semangat dan dukungan yang luar biasa semoga setelah perjuangan ini kita tetap saling bersilaturahmi.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah rabbil'Alamin penulis haturkan rasa syukur kehadiran Allah SWT yang telah memberikan rahmat, hidayah, dan ridha-Nya, sehingga penulis dapat menyelesaikan studi di jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan skripsi ini dengan baik dan lancar.

Selanjutnya penulis menghaturkan ucapan terima kasih kepada semua pihak yang telah memberikan doa, harapan, dan semangat untuk terselesaikannya skripsi ini. Ucapan terima kasih, penulis sampaikan 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. Hj. Bayyinatul Muchtaromah, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiان selaku ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Fressy Nugroho, M.T dan Dr.H.Ahmad Barizi, M.A selaku dosen pembimbing skripsi, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
5. Segenap civitas akademika jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
6. Ayahanda H.M.Sulkhan dan Ibunda Hj.Siti Mas'Amah tercinta yang senantiasa memberikan doa, spirit, tenaga, biaya, dan restunya kepada penulis dalam menuntut ilmu di kampus hijau tercinta.
7. Semua pihak yang ikut membantu dalam menyelesaikan skripsi ini baik berupa materi maupun moril.

Peneliti menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan peneliti berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi peneliti secara pribadi. *Amin Ya Rabbal Alamin.*

Wassalamu'alaikum Wr. Wb.

Malang, 07 Juli 2014

M.Arif Mukhlisin

DAFTAR ISI

HALAMAN JUDUL
HALAMAN PENGAJUAN	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
HALAMAN PERNYATAAN	iiii
MOTTO	v
PERSEMBAHAN	vi
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
ABSTRAK	xiii
ABSTRACT	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
1.6 Sistematika Penulisan	7
BAB II KAJIAN PUSTAKA	8
2.1 Pembelajaran Bahasa Arab Dalam Pandangan Islam	8
2.2 <i>Ilmu Sharraf</i>	9
2.3 Permainan (<i>Game</i>)	11
2.3.1 Peraturan Permainan	12
2.3.2 <i>Plot</i>	12
2.3.3 Tema	12
2.3.4 Karakter	13
2.3.5 Objek	13

2.3.6	Teks, Grafik, dan Suara	13
2.3.7	Animasi	13
2.3.8	<i>User Interface</i>	13
2.4	Permainan <i>Spider Solitaire</i>	15
2.5	<i>Neural Network Backpropagation</i>	16
2.5.1	Algoritma Pelatihan	17
2.5.2	Algoritma Aplikasi	19
2.6	<i>Android</i>	21
BAB III PERANCANGAN SISTEM		24
3.1	Analisa dan Perancangan Sistem	24
3.1.1	Keterangan Umum <i>Game</i>	24
3.1.2	<i>Story Board Game</i>	25
3.1.3	Penampilan Umum <i>Game</i>	27
3.1.4	Konten	28
3.2	Perancangan Kecerdasan Buatan (<i>Artificial Intelligence</i>)	29
3.2.1	Perancangan Metode <i>Neural Network Backpropagation</i>	29
3.3	Perancangan Aplikasi <i>Game</i>	35
3.3.1	Perancangan <i>Leveling</i>	35
3.3.2	Perancangan Alur / <i>Flowchart Game</i>	36
3.3.3	Kebutuhan Sistem	37
3.3.4	Kebutuhan <i>Device Minimum Pemain</i>	38
BAB IV HASIL DAN PEMBAHASAN		39
4.1	Implementasi	38
4.2	Implementasi <i>Algoritma Neural Network Backpropagation</i>	39
4.3	Implementasi Aplikasi <i>Game</i>	46
4.4	Pengujian algoritma <i>Neural Network Backpropagation</i>	49
4.5	Pengujian <i>Game</i>	52
4.6	Integrasi Dalam Islam	53
BAB V PENUTUP		55
5.1	Kesimpulan	55
5.2	Saran	55

DAFTAR PUSTAKA 56
LAMPIRAN



DAFTAR GAMBAR

Gambar 2.1 <i>Spider Solitaire</i>	15
Gambar 2.2 Arsitektur <i>Backpropagation</i>	17
Gambar 2.3 Algoritma <i>Backpropagation</i>	17
Gambar 2.4 Arsitektur <i>Android</i>	22
Gambar 3.1 Tampilan <i>Splash</i>	26
Gambar 3.2 Tampilan Menu	26
Gambar 3.3 Tampilan <i>About</i>	26
Gambar 3.4 Tampilan <i>Score</i>	26
Gambar 3.5 Tampilan <i>Help</i>	26
Gambar 3.6 <i>Level 1</i>	27
Gambar 3.7 <i>Level 2</i>	27
Gambar 3.8 <i>Level 3</i>	27
Gambar 3.9 Permainan <i>Sharraf</i>	27
Gambar 3.10 Konten <i>Wazan Fa'ala</i>	28
Gambar 3.11 Konten <i>Wazan Af'ala</i>	28
Gambar 3.12 Konten <i>Wazan Fa'lala</i>	28
Gambar 3.13 Konten <i>Wazan Ifta'ala</i>	29
Gambar 3.14 Perancangan Kecerdasan Buatan	29
Gambar 3.15 <i>Flowchart</i> Algoritma Pelatihan	31
Gambar 3.16 <i>Flowchart</i> Algoritma Aplikasi	34
Gambar 3.17 <i>Flowchart</i> Permainan	36
Gambar 4.1 Tampilan <i>Splashscreen</i>	46
Gambar 4.2 Tampilan Menu	46
Gambar 4.3 Tampilan Level 1	47
Gambar 4.4 Tampilan Level 2	47
Gambar 4.5 Tampilan Level 3	47
Gambar 4.6 Tampilan Permainan	48
Gambar 4.7 Tampilan <i>About</i>	48
Gambar 4.8 Tampilan <i>Score</i>	48
Gambar 4.9 Tampilan <i>Help</i>	49
Gambar 4.10 Tampilan <i>Win</i>	49

DAFTAR TABEL

Tabel 2.1 <i>Tashrif Istihlahi</i>	10
Tabel 3.1 Pola Pelatihan.....	31
Tabel 3.2 Kebutuhan <i>Device</i> Pemain	38
Tabel 4.1 Keterangan Algoritma <i>Neural Network Backpropagation</i>	45
Tabel 4.2 Hasil Pengujian 1	49
Tabel 4.3 Hasil Pengujian 2	51
Tabel 4.4 Hasil Pengujian <i>Game</i>	52



BAB I

PENDAHULUAN

1.1 Latar Belakang

Cintailah bahasa Arab, karena aku (Rasullullah) adalah turunan arab, *Al-Qur'an* berbahasa arab dan sekaligus bahasa penghuni surga kelak. (HR. Al-Baihaqie). Dalam mempelajari bahasa arab ada dua ilmu yakni *nahwu* dan *Sharraf*, karena pentingnya ilmu ini dalam mempelajari bahasa arab muncullah ungkapan :

الصرف ام العلوم والنحو ابوها

“ilmu *Sharraf* adalah induk segala ilmu dan *nahwu* bapaknya” (K.H.Moch.Anwar).

Ilmu *Sharraf* disebut induk segala ilmu sebab ilmu *Sharraf* itu melahirkan bentuk setiap kalimat sedangkan kalimat itu menunjukkan bermacam-macam ilmu. Kalau tidak ada kalimat tentu tidak ada tulisan, tanpa tulisan sukar mendapatkan ilmu. Adapun *nahwu* disebut juga dengan bapaknya ilmu, sebab ilmu *nahwu* itu untuk memperbaiki setiap kalimat dalam susunannya (K.H.Moch.Anwar, 2000). Mempelajari ilmu *Nahwu* dan *Sharraf* merupakan suatu keharusan, terutama bagi seorang muslim yang menjadikan *Al Qur'an* dan *As Sunnah* sebagai pedoman hidupnya. Hal ini sebagaimana difirmankan oleh Allah SWT dalam *Al Qur'an* :
Sesungguhnya Kami menurunkannya berupa Al Quran dengan berbahasa Arab, agar kamu

memahaminya (QS. Yusuf / 12:2). (Departement Agama RI, 2009). *Al Qur'an* yang diturunkan dalam bahasa Arab tidak mungkin dapat difahami apalagi diamankan isinya tanpa mengetahui ilmu *Nahwu* dan ilmu *Sharraf*. *Al Qur'an* yang diwahyukan dengan menggunakan bahasa Arab dianggap sulit difahami oleh mayoritas manusia, khususnya bangsa Indonesia, Padahal pada hakikatnya *Al Qur'an* diturunkan dalam bahasa arab untuk memudahkan dalam memahaminya. Dalam pelajaran ilmu *Sharraf*, ulama' *Sharraf* telah membagi *tashrif* menjadi dua macam, yaitu *TASHRIF ISTHILAH* dan *TASHRIF LUGHOWI*. *Tashrif isthilahi* adalah untuk mengetahui bentuk *sighot* dari suatu kalimat. Contoh : **فَعَلَ** (*Fa'ala*) dinamakan *Fi'il madhi*, artinya telah berbuat ia seorang laki-laki. *Fi'il madhi* maksudnya perbuatan yang telah dikerjakan. Dalam terjemah bahasa Indonesia biasanya diterjemahkan dengan telah (Maftuh Ahnan, 1999).

Game secara naluri adalah bagian dari kehidupan manusia. Hal ini didasarkan kenyataan bahwa manusia sangat menyukai sebuah permainan. Namun saat ini *game* dengan unsur edukasi di dalamnya sangat jarang ditemukan. Sebuah *game* yang ada dalam perangkat bergerak tentu dapat memberikan hiburan yang menarik kepada para pecinta *game*. Karena mereka dapat bermain *game* dimana saja secara praktis dan mudah. Kesederhanaan dalam bermain *game* namun tidak membosankan, serta dapat membuat pemain ingin kembali memainkan *game* tersebut. Oleh karena itu berbagai pendekatan terus dikembangkan untuk membuat

sebuah *game* yang dapat dimainkan dalam perangkat bergerak atau *mobile* yang menarik, salah satunya berbasis *android*.

Pada berita media *online merdeka.com* tanggal 7 oktober 2013 disebutkan bahwa *game-game* terkenal saat ini sudah dirilis dalam *platform android* dan kini *game* tersebut sudah siap mengudara secara *global*. *Android* adalah sistem operasi untuk perangkat bergerak yang berbasis *Linux*. Meski kemunculan sistem operasi besutan *Google* ini sudah ada sejak beberapa tahun lalu, namun *android* baru populer belakangan ini. Ini tidak lepas dari banyaknya ponsel berbasis *android* yang beredar di pasaran. *Android* merupakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri secara bebas. Sistem operasi *android* pada *smartphone* merupakan sistem operasi yang banyak diminati saat ini. Dan juga, salah satu kelebihan sistem operasi *android* bagi para *programmer* maupun *developer* adalah bersifat *open source* sehingga mereka memiliki kesempatan untuk membuat dan mengembangkan aplikasi-aplikasi sesuai yang diinginkan (Ryiadhy, 2012).

Secara umum *Neural Network* (NN) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. NN ini merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Secara sederhana NN adalah sebuah alat pemodelan data statistik

non-linear. NN dapat digunakan untuk memodelkan hubungan yang kompleks antara *input* dan *output* untuk menemukan pola-pola pada data. Secara mendasar, sistem pembelajaran merupakan proses penambahan pengetahuan pada NN yang sifatnya kontinuitas sehingga pada saat digunakan pengetahuan tersebut akan dieksploitasikan secara maksimal dalam mengenali suatu objek. Pada *beritateknologi.com* tanggal 11 oktober 2011 disebutkan bahwa peneliti Jepang bernama Hosamu Hasegawa menciptakan robot yang bisa melakukan tugas-tugas tertentu dengan sebuah bahasa pemrograman dia menggunakan *Self Organising Incremental Neural Network* atau disebut SOINN, sebuah algoritma yang memungkinkan robot untuk belajar dari pengalamannya.

Salah satu pengembangan arsitektur *Neural Network* adalah *Backpropagation* yaitu algoritma pembelajaran terawasi dan terutama digunakan oleh *Multi-layer-perceptron* untuk mengubah bobot yang terhubung dengan neuron *layer* tersembunyi jaringan. Algoritma *backpropagation* menggunakan hitungan *error output* untuk mengubah nilai bobot dalam arah mundur. Untuk mendapatkan *error* jaringan ini, fasa *forwardpropagation* harus telah dilakukan sebelumnya (Heri, 2011).

Kita sudah mengetahui bersama bahwasanya Rasulullah menyeru umatnya untuk mencintai bahasa arab sebagai bahasa *Al-Qur'an* dan bahasa penghuni surga. Ilmu *Nahwu Sharraf* merupakan disiplin ilmu bahasa arab yang sering diajarkan di pesantren-pesantren, pengajian rutin dan sekolah. Bahkan sudah dimasukkan dalam

kurikulum Nasional walaupun tidak secara mandiri dijadikan sebagai suatu mata pelajaran, namun ilmu *Nahwu* dan *Sharraf* merupakan bagian dari standar kompetensi dan kompetensi dasar yang harus dipelajari oleh peserta didik. Timbul pertanyaan di benak penulis sejauh mana lembaga pendidikan baik sekolah maupun pesantren berupaya dalam melaksanakan pembelajaran ilmu Nahwu dan *Sharraf* terhadap siswa maupun santri? Metode apa yang digunakan oleh guru, tentor, dosen dan ustadz dalam proses pembelajaran ilmu *Nahwu* dan *Sharraf* terhadap peserta didiknya?

Sabda Rasulluah dan Pertanyaan-pertanyaan di atas itulah yang melatarbelakangi penulis untuk mencintai bahasa arab dengan membuat sebuah permainan sebagai metode yang menyenangkan dalam mempelajari ilmu *Sharraf* yang diimplementasikan dalam bentuk permainan melalui *ponsel android* sebagai *smartphone* yang sangat diminati.

1.2 Rumusan Masalah

Dari latar belakang diatas dapat dirumuskan masalahnya adalah Bagaimana metode *Neural Network Backpropagation* dapat menentukan *level* dalam *game* edukasi pengenalan dan pembelajaran *Tashrif Isthilahi* Dalam Ilmu *Sharraf* ?

1.3 Batasan Masalah

Pada Aplikasi *game* ini diberikan pembatasan masalah sebagai berikut:

- a. *Tashrif Isthilahi* yang dijadikan objek dalam *game* dari wazan *fa'ala*, *af'ala*, *fa'lala* dan *ifta'ala* dalam *Sharraf* karena wazan tersebut adalah acuan yang banyak digunakan dalam pengenalan awal *tashrifan*.
- b. *Shighot*(bentuk kata) meliputi *Fi'il Madli*, *Fi'il Mudhori'*, *Masdar G.Mim*, *Masdar Mim*, *Isim Fa'il*, *Isim maf'ul*, *Fi'il Amar*, *Fi'il Nahi*, *Isim Z.Makan*, *Isim Alat*
- c. Bahasa Pemrograman Yang digunakan adalah *Java*.
- d. Aplikasi Permainan ini berbasis *mobile* yang diimplementasikan pada *platform Android OS mobile*.

1.3 Tujuan Penelitian

Tujuan dari dibuatnya *game* ini adalah :

- a. Membuat *game* dari permainan *spider solitaire* untuk pengenalan dan pembelajaran ilmu *Sharraf* menjadi permainan yang bisa dimainkan di *smartphone*.
- b. Mengimplementasikan metode *Neural Network Backpropagation* sebagai penentuan *level* berdasarkan 2 *inputan* yakni perpindahan kartu dan waktu.

1.5 Manfaat Penelitian

Manfaat pembuatan aplikasi *game* ini adalah memperkenalkan dan memberikan pengetahuan dalam mempelajari perubahan kata bahasa arab disetiap kalimat dalam pendidikan formal maupun non formal. Dan juga sebagai sarana pembelajaran ilmu *Sharraf tashrif isthilahi* melalui *game* mobile yang bisa dimainkan kapan saja, dimana saja, dan oleh siapa saja.

1.6 Sistematika Penulisan

Penulisan skripsi ini tersusun dalam 5 (lima) bab dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi uraian tentang latar belakang masalah yang mendasari pentingnya diadakan penelitian, identifikasi, pembatasan dan perumusan masalah, tujuan penelitian, kegunaan penelitian yang diharapkan serta sistematika penulisan.

BAB II KAJIAN PUSTAKA

Bab ini berisi tinjauan pustaka yang mendiskripsikan pengertian terkait dengan kebutuhan penelitian dan mengacu pada penelitian sebelumnya.

BAB III PERANCANGAN SISTEM

Menganalisa kebutuhan sistem dan pengumpulan data untuk membuat *game* meliputi spesifikasi kebutuhan *software* dan langkah-langkah pembuatan *game*.

BAB IV HASIL DAN PEMBAHASAN

Menjelaskan implementasi teori pada pengujian *game* dan metode yang diterapkan serta hasil uji coba *game* di berbagai versi *smartphone android*.

BAB V PENUTUP

Berisi uraian tentang pokok-pokok kesimpulan dan saran-saran yang perlu disampaikan kepada pihak-pihak yang berkepentingan dengan hasil penelitian.

BAB II

KAJIAN PUSTAKA

2.1 Pembelajaran Bahasa Arab Dalam Pandangan Islam

Belajar merupakan kewajiban bagi semua umat islam. Seperti yang dijelaskan dalam *Al Hadist* tentang menuntut ilmu :

طَلَبُ الْعِلْمِ فَرِيضَةٌ عَلَى كُلِّ مُسْلِمٍ

Artinya : “menuntut ilmu wajib bagi setiap muslim” (HR. Ibnu Abdil Bari).

Belajar memiliki tiga arti penting menurut *Al Quran*. *Pertama*, bahwa orang yang belajar akan mendapatkan ilmu yang dapat digunakan untuk memecahkan segala masalah yang dihadapinya di kehidupan dunia. *Kedua*, manusia dapat mengetahui dan memahami apa yang dilakukan karena setiap apa yang yang diperbuat akan dimintai pertanggung jawabannya. *Ketiga* dengan ilmu yang dimilikinya mampu mengangkat derajatnya di mata Allah. Belajar merupakan kebutuhan dan berperan penting dalam kehidupan manusia. Hal ini disebabkan manusia terlahir tidak mengetahui apa-apa, ia hanya dibekali potensi jasmani dan rohani (QS. Al-Nahl / 16:78). Maka sangat beralasan jika mengapa dan bagaimana manusia itu dipengaruhi oleh bagaimana ia belajar (William Berkson, 2003).

Belajar Bahasa Arab merupakan jalan untuk bisa memahami *Al-Qur'an*

إِنَّا أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَعَلَّكُمْ تَعْقِلُونَ

Artinya : “*Sesungguhnya Kami menurunkannya berupa Al Quran dengan berbahasa Arab, agar kamu memahaminya*” (QS. Yusuf / 12:2).

Dan Imam Syafi'i berkata: wajib pada tiap-tiap muslim untuk belajar bahasa arab kalau ingin sampai kepada kesungguhannya dalam melaksanakan *kefardhuannya*. Jika bukan karena mengamalkan *fardhu*, maka belajar bahasa arab hukumnya *sunnah*, selain yang ingin mengetahui seluk beluk Syari'at Islam, karena wajib bagi para alim syari'at belajar bahasa arab untuk memahami tentang *syari'at Qur'ani* atau *syari'at Haditsi*. Rasulullah SAW bersabda :

أحبوا العرب لثلاث لأني عربي ، والقرآن عربي ، وكلام أهل الجنة عربي

Artinya : “*Cintailah bahasa Arab karena tiga hal, yaitu karena aku (Nabi Muhammad SAW) adalah orang Arab, dan Al-Qur'an dengan bahasa Arab, dan bahasa penghuni surga adalah bahasa Arab*” (HR. Ath-Thabrani).

Salah satu cabang ilmu bahasa arab adalah ilmu *sharraf* yang perlu dipelajari untuk memudahkan menggunakan kata-kata dalam suatu kalimat, untuk mengetahui seluk bentuk perubahan kata dan memahami setiap perubahan kata.

2.2 Ilmu *Sharraf*

Sharraf menurut bahasa (*lughot*) yaitu berubah atau mengubah dari bentuk aslinya ke bentuk yang lain. Misalnya merubah bentuk bangunan rumah kuno menjadi bentuk bangunan rumah yang modern. Sedangkan menurut istilah (menurut kalangan ulama'*Sharraf*), yaitu berubahnya bentuk asal pertama (*fi'il madhi*) menjadi *fi'il mudhori*, dari *fi'il mudhori* menjadi

mashdar, menjadi *isim fa'il*, *isim maf'ul*, *fi'il amar*, *fi'il nahi*, *isim zaman*, *isim makan*, dan terakhir sampai pada *isim alat*. Maksud dan tujuan dari perubahan-perubahan bentuk tersebut adalah agar memperoleh makna/arti yang berbeda. Dengan demikian ilmu yang mempelajari berbagai macam bentuk perubahan kata, asal usul kata atau keadaannya itu dinamakan : ILMU SHARRAF.

Tashrif Isthilahi, yaitu *tashrifan* untuk mengetahui bentuk *shighat* dari suatu kalimat atau dari *shighat fi'il madhi* dan diakhiri *isim alat*. Contoh *tashrif isthilahi* : (Maftuh Ahnan, 1999)

Tabel 2.1 *Tashrif Isthilahi*

Isim Alat	Isim Z. Makan	Fi'il Nahi	Isim Maf'ul	Isim Maf'ul	Isim Fa'il	Masdar Mim	Masdar G.Mim	Fi'il Mudhori'	Fi'il Madli
مِفْعَلٌ	مَفْعَلٌ ٢	لَا تَفْعَلْ	أَفْعَلٌ	مَفْعُولٌ	فَاعِلٌ	مَفْعَلًا	فَعْلًا	يَفْعَلُ	فَعَلٌ

1. فَعَلٌ dinamakan *fi'il madhi*, artinya telah berbuat ia seorang laki-laki.
2. يَفْعَلُ dinamakan *fi'il mudhori'*, artinya sedang/akan berbuat ia seorang laki-laki.
3. مَفْعَلًا / فَعْلًا dinamakan *masdar*, artinya perbuatan. *Masdar* ini ada *ghoiru mim* (tanpa diawali mim) dan *masdar mim* (diawali dengan mim).
4. فَاعِلٌ dinamakan *isim fa'il*, artinya seorang laki-laki yang berbuat.

5. مَفْعُولٌ dinamakan *isim maf'ul*, artinya seorang laki-laki yang dibuat (dikenai pekerjaan) / obyek.
6. أَفْعَلٌ dinamakan *fi'il amar*, artinya hendaklah engkau seorang laki-laki berbuat (perintah).
7. لَا تَفْعَلُ dinamakan *fi'il nahi*, artinya janganlah engkau seorang laki-laki berbuat (larangan).
8. مَفْعَلٌ dinamakan *isim zaman*, artinya waktu berbuat.
9. مَفْعَلٌ dinamakan *isim makan*, artinya tempat berbuat.
10. مَفْعَلٌ dinamakan *isim alat*, artinya alat untuk berbuat. (Maftuh Ahnan, 1999).

2.3 Permainan (Game)

Teori permainan (*game*) pertama kali ditemukan oleh sekelompok ahli matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern yang berisi :

“Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan

kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi.”

Lam (2003) mengemukakan definisi *Mobile Game* merupakan sebuah permainan video yang dimainkan pada *mobile phone*, ataupun PDA (*Personal Digital Assistant*) dan bukan pada konsol permainan maupun mesin ditingdong. *Mobile game* adalah *game* yang tidak hanya dimainkan melalui *mobile phone*, namun dapat dikembangkan dalam berbagai macam *mobile handset* seperti PDA, Symbian OS, dan *Microsoft's Smartphone*.

Menurut Teresa Dillon dan Aeny (2010), elemen-elemen dasar sebuah *game* adalah :

2.3.1 Peraturan Permainan

Peraturan permainan merupakan sebuah aturan bagaimana menjalankan permainan dengan berbagai fungsi-fungsi objek dan karakter.

2.3.2 *Plot*

Plot merupakan informasi-informasi atau perintah-perintah yang dibutuhkan *user* pada sebuah permainan seperti bagaimana menyelesaikan sebuah permainan dan lain-lain.

2.3.3 Tema

Biasanya berisi pesan moral yang dibutuhkan.

2.3.4 Karakter

Karakter merupakan pemain-pemain yang memiliki pergerakan. Entah itu digerakkan *player* ataupun yang telah diberikan algoritma-algoritma.

2.3.5 Objek

Objek merupakan sesuatu yang dibutuhkan *player* dalam menyelesaikan sebuah permainan.

2.3.6 Teks, Grafik, dan Suara

Merupakan hal-hal penting yang harus terdapat dalam sebuah permainan sebagai pendukung permainan agar terlihat hidup.

2.3.7 Animasi

Animasi ini berisi gerakan-gerakan karakter yang ada pada sebuah permainan.

2.3.8 *User Interface*

Merupakan fitur-fitur dalam sebuah permainan sebagai media komunikasi dan interaksi antara pemain dengan aplikasi.

Menurut M.Husni penelitian tahun 2011 yang berjudul MEMBANGUN APLIKASI MULTIMEDIA EDUKATIF SEBAGAI ALAT BANTU AJAR ILMU *SHARRAF* BERDASARKAN METODE PEMBELAJARAN “*SHARRAF* KRAPYAK” di STIMIK AMIKOM Yogyakarta tentang Aplikasi Multimedia Pembelajaran Ilmu *Sharraf* Menggunakan Metode *Sharraf* Krapyak adalah sebagai berikut:

1. Multimedia dapat digunakan sebagai salah satu alternatif dalam menyampaikan informasi, khususnya di bidang pendidikan, dimana multimedia dapat mengemas informasi sehingga menjadi menarik dan lebih efektif.
2. Aplikasi berbentuk multimedia yang dihasilkan dalam pembuatan dan mendesainnya menggunakan beberapa *software* yaitu *Adobe Photoshop CS2*, *Corel Draw X3*, *Adobe Audition 1.0*, *Adobe Flash Professional CS4*, dan *Macromedia Director MX 2004*.
3. Dengan hasil pengujian sistem aplikasi multimedia interaktif ini melalui questioner yang diisi oleh responden, maka sistem yang baru dikatakan lebih menarik dan lebih mudah untuk dipahami sehingga membantu dalam memberikan dorongan dan motivasi dalam belajar.
4. Aplikasi multimedia ini mampu menjadi alternatif media informasi tambahan di samping media pembelajaran buku. Tetapi juga memiliki kekurangan bahwa aplikasi ini tidak dapat diterapkan dalam aplikasi *mobil* yang lebih praktis dan metode pembelajarannya mengacu di STIMIK AMIKOM saja.

2.4 Permainan *Spider Solitaire*



Gambar 2.1 *Spider Solitaire*

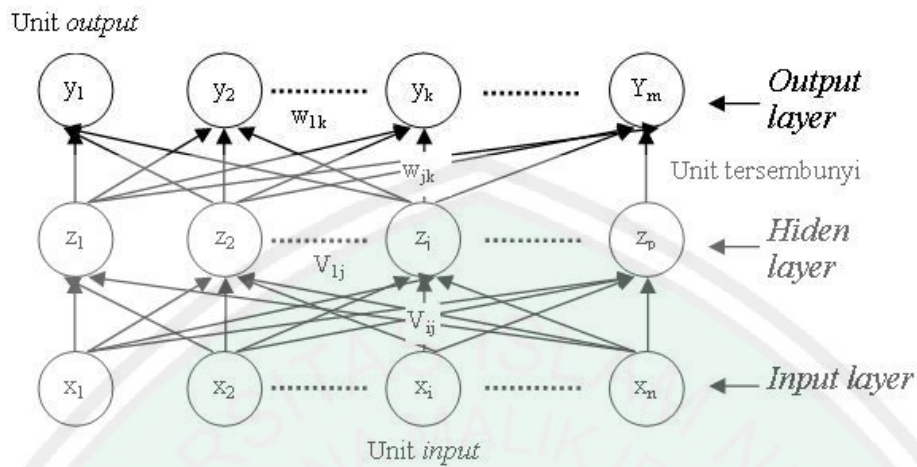
Spider Solitaire adalah suatu *game* yang cara mainnya hampir mirip dengan *solitaire*, mengurutkan suatu kartu dengan variasi *level game* yang berbeda-beda. *Spider Solitaire* sebenarnya menyenangkan, dan sangat terkenal, hanya saja membutuhkan kesabaran dan perencanaan yang cermat sehingga seringkali permainan ini mengambil lebih dari setengah jam pemikiran yang terus-menerus untuk tidak menghasilkan suatu langkah yang sia-sia. Tujuan dari *Spider Solitaire* adalah untuk membangun urutan kartu mulai dari king sampai as dan ketika sudah terurut, kadang-kadang bisa tampak hampir mustahil untuk menyelesaikan permainan yang terkadang menghabiskan waktu berjam-jam pemikirannya. Intinya, hal yang paling penting dalam *Spider Solitaire* adalah untuk menciptakan kolom kosong, dan mencoba dan menjaga mereka kosong.

2.5 Neural Network Backpropagation

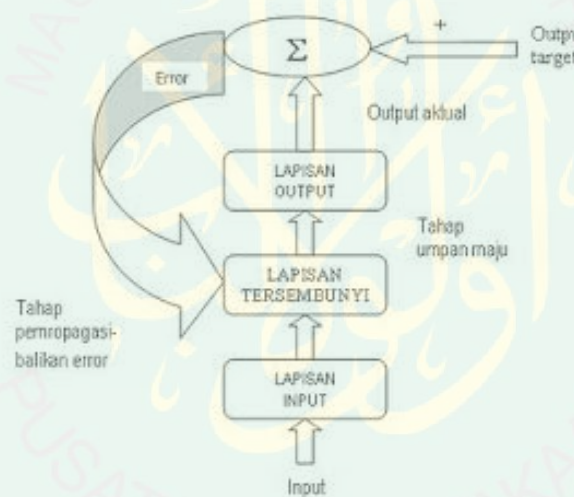
Backpropagation adalah algoritma pembelajaran terawasi dan terutama digunakan oleh *Multi-layer-perceptron* untuk mengubah bobot yang terhubung dengan neuron *layer* tersembunyi jaringan. Algoritma *backpropagation* menggunakan hitungan *error output* untuk mengubah nilai bobot dalam arah mundur. Untuk mendapatkan *error* jaringan ini, fasa *forwardpropagation* harus telah dilakukan sebelumnya. Saat propagasi dalam arah maju, neuron diaktivasi dengan menggunakan fungsi aktivasi *sigmoid*. Metode ini menurunkan gradien untuk meminimalkan penjumlahan *error* kuadrat *output* jaringan dengan aturan *generalized delta rule*. *Neural Network Backpropagation* terdiri dari banyak lapisan (Diyah, 2006).

1. Lapisan *input* yang terdiri dari neuron-neuron atau unit-unit *input* mulai dari unit 1 sampai n.
2. Lapisan tersembunyi yang terdiri dari unit-unit tersembunyi mulai dari unit 1 sampai p.
3. Lapisan *output* yang terdiri dari unit-unit *output* 1 sampai m.

Arsitektur dijelaskan pada gambar 2.2 dan algoritma dijelaskan pada gambar 2.3.



Gambar 2.2 Arsitektur *Backpropagation* (Diyah, 2006).



Gambar 2.3 Algoritma *Backpropagation* (Diyah, 2006).

Algoritma *backpropagation* dibagi menjadi 2 bagian yakni algoritma pelatihan dan algoritma aplikasi.

2.5.1 Algoritma pelatihan

0. Inisialisasi bobot-bobot.

Tentukan angka pembelajaran dan nilai toleransi *error*.

1. **While** dimana kondisi berhenti tidak terpenuhi **do** langkah ke-2 sampai langkah ke-9.

2. Untuk setiap pola pelatihan lakukan langkah ke-3 sampai ke-8.

Tahap Umpan Maju

3. Setiap unit *input* x_i (dari unit 1 sampai n lapisan *input*) mengirimkan sinyal *input* ke semua unit pada lapisan tersembunyi.

4. Pada setiap unit di lapisan tersembunyi z_j (dari unit 1 sampai unit p ; $i=1, \dots, n$; $j=1, \dots, p$) sinyal *output* lapisan tersembunyi dihitung dengan menerapkan fungsi aktivasi terhadap penjumlahan sinyal-sinyal *input* berbobot x_i :

$$z_j = f(v_{0j} + \sum_{i=1}^n x_i v_{ij})$$

Kemudian dikirim ke semua unit di lapisan *output*.

5. Kemudian unit di lapisan *output* y_k (dari unit 1 sampai unit ke- m ; $i=1, \dots, n$; $k=1, \dots, m$) dihitung sinyal *output*nya dengan menerapkan fungsi aktivasi terhadap penjumlahan sinyal-sinyal *input* berbobot z_j bagi lapisan ini :

$$y_k = f(w_{0k} + \sum_{j=1}^p z_j w_{jk})$$

Tahap Mempagasibalikkan Error

6. Setiap unit *output* y_k (dari unit ke-1 sampai unit ke- m ; $j=1, \dots, p$; $k=1, \dots, m$) menerima pola target t_k lalu informasi kesalahan lapisan *output* dihitung dan dikirim ke lapisan bawahnya dan digunakan untuk menghitung besar koreksi bobot antara lapisan tersembunyi dengan lapisan *output* :

$$\delta_k = (t_k - y_k) f'(w_{0k} + \sum_{j=1}^p z_j w_{jk})$$

$$\Delta W_{jk} = a \delta_k z_j$$

$$\Delta W_{0k} = a \delta_k$$

7. Pada setiap unit di lapisan tersembunyi (dari unit ke-1 sampai unit ke- p ; $i=1, \dots, n$; $j=1, \dots, p$; $k=1, \dots, m$) dilakukan perhitungan informasi kesalahan

lapisan tersembunyi kemudian digunakan untuk menghitung besar koreksi bobot antara lapisan *input* dan lapisan tersembunyi.

$$\delta_j = \left(\sum \delta_k w_{jk} \right)_{k=1}^m f'(v_{0j} + \sum x_i v_{ij})_{i=1}^n$$

$$\Delta v_{ij} = a \delta_j x_i$$

$$\Delta v_{0j} = a \delta_j$$

Tahap Peng-update-an bobot

8. Setiap unit *output* y_k (dari unit ke-1 sampai unit ke-m) dilakukan peng-update-an bobot ($j=0, \dots, p; k=1, \dots, m$) sehingga bobot yang baru menjadi :

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk}$$

Dari unit ke-1 sampai unit ke-p di lapisan tersembunyi juga dilakukan peng-update-an pada bobotnya ($i=0, \dots, n; j=1, \dots, p$);

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

9. Tes kondisi berhenti setelah melakukan *learning* dan mendapatkan bobot-bobot baru.

2.5.2 Algoritma Aplikasi

0. Inisialisasi bobot. Bobot ini diambil dari bobot-bobot terakhir yang diperoleh dari algoritma pelatihan.

1. Untuk setiap vektor *input*, dilakukan langkah ke-2 sampai ke-4.

2. Setiap unit *input* x_i (dari unit ke-1 sampai unit ke-n pada lapisan *input*; $i=1, \dots, n$) menerima sinyal *input* pengujian x_i dan menyiarkan sinyal x_i ke semua unit pada lapisan unit tersembunyi.

3. Setiap unit di lapisan tersembunyi z_j (dari unit ke-1 sampai unit ke-p; $i=1, \dots, n; j=1, \dots, p$) menghitung sinyal *output*nya dengan menerapkan fungsi

aktivasi terhadap penjumlahan sinyal-sinyal *input* x_i . Sinyal *output* dari lapisan tersembunyi kemudian dikirim kesemua unit lapisan di atasnya :

$$z_j = f(v_{0j} + \sum_{i=1}^n x_i v_{ij})$$

4. Setiap unit *output* y_k (dari unit ke-1 sampai unit ke- m ; $j=1, \dots, p$; $k=1, \dots, m$) menghitung sinyal *output*nya dengan menerapkan fungsi aktivasi terhadap penjumlahan sinyal-sinyal *input* bagi lapisan ini, yaitu sinyal-sinyal *input* z_j dari lapisan tersembunyi : (Diyah, 2006).

$$y_k = f(w_{0k} + \sum_{j=1}^p z_j w_{jk})$$

Menurut Achmad Hidayanto, R.Rizal Isnanto, Dian Kurnia Widya Buana penelitian tahun 2008 yang berjudul “Identifikasi Tanda-Tangan Menggunakan Jaringan Saraf Tiruan Perambatan-Balik (*Backpropagation*)” di Universitas Diponegoro Semarang tentang jaringan syaraf tiruan perambatan-balik yang dilatih dengan data citra tanda-tangan seseorang, mengenali pemiliki anda-tangan tersebut, untuk mendapatkan arsitekturan parameter jaringan saraf tiruan yang terbaik perlu memperhatikan jumlah simpul lapis tersembunyi, laju pembelajaran, momentum, toleransi galat, dan jumlah *epoch*. Dan semakin bertambahnya nilai momentum akan berpengaruh pada berkurangnya jumlah *epoch* yang menyebabkan proses pelatihan/pengujian jaringan akan semakin cepat (Hidayanto, 2008).

Muhammad Hanief, Metri Annisa, Narendi Mudandri, dan Kadarsyah Suryadi penelitian tahun 2009 yang berjudul “Prediksi Masa Studi Sarjana dengan *Artificial Neural Network*” di Institut Teknologi Bandung tentang prediksi lama masa studi dibutuhkan oleh manajemen perguruan tinggi dalam

menentukan kebijakan preventif terkait pencegahan dini kasus DO. Kriteria pemilihan model yang digunakan adalah minimasi *sum square error*. Model terbaik untuk memprediksi adalah model *Artificial Neural Network* dengan arsitektur *Multiplayer Perceptor*. Dari penelitian ini ditemukan bahwa lama studi dipengaruhi oleh IPK, jumlah mata kuliah yang diambil, jumlah mata kuliah mengulang dan jumlah pengambilan mata kuliah tertentu. Penelitian ini memiliki kelebihan dapat mengolah data dan mengambil sebuah kesimpulan dari data tersebut tetapi juga memiliki kekurangan karena tidak bisa menjamin kebenaran dari kesimpulan yang telah diambil (Hanief, 2009).

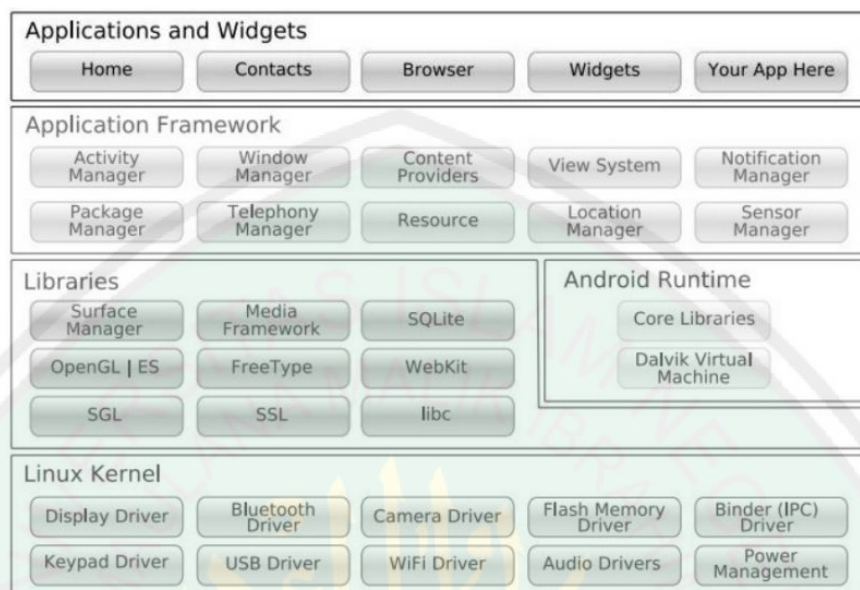
2.6 *Android*

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* diakuisisi oleh *google* pada Juli 2005, dan baru dirilis perdana pada 5 November 2007. *Android* berlisensi di bawah *GNU, General Public Lisensi Versi 2 (GPLv2)*, yang memperbolehkan pihak ketiga untuk mengembangkannya dengan menyertakan term yang sama. Pendistribusiannya di bawah Lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua dan seterusnya. (Safaat, 2011).

Android dirancang dengan arsitektur sebagai berikut :

1. *Application dan Widgets*
2. *Application Frameworks*
3. *Libraries*
4. *Android Run Time*

5. *Linux Kernel*



Gambar 2.4 Arsitektur *Android* (Safaat, 2011).

Beberapa keunggulan *Platform Android* adalah sebagai berikut :

1. Lengkap (*Complete Platform*). Para desainer dapat melakukan pendekatan yang komprehensif ketika sedang mengembangkan *platform Android*. *Android* menyediakan banyak *tools* dalam membangun *software* dan merupakan sistem operasi yang aman.
2. Terbuka (*Open Source Platform*). *Platform Android* disediakan melalui lisensi *open source*.
3. Bebas (*Free Platform*). *Android* merupakan platform atau aplikasi yang bebas untuk dikembangkan. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform Android*.

Menurut Ryiadhy Budhy Nugraha, Muhammad Danaparamita, Aldhy Nephyana penelitian tahun 2012 yang berjudul “APLIKASI GAME GARUDA BERBASIS ANDROID” di Binus University tentang

pembangunan aplikasi *game* edukasi berbasis *Android* yang berwawasan cinta tanah air dan kebudayaan daerah Indonesia. *Game* garuda mudah untuk dimainkan dan memiliki *gameplay* dan grafis yang menarik. Fitur-fitur yang terdapat dalam *game* garuda sesuai dengan harapan pemain dan unsur edukasi yang terdapat dalam *game* garuda dapat menambah pengetahuan pemain. *Game* ini memiliki kelebihan yang menarik karena selain bermain *game* juga dapat mempromosikan Indonesia baik dari segi budaya, wisata, dan sejarah. Tetapi *game* ini juga memiliki kekurangan karena masih banyak aspek-aspek penting terkait Indonesia yang belum dimasukkan (Budhy Ryiadhy, 2012).



BAB III

PERANCANGAN SISTEM

3.1 Analisa dan Perancangan Sistem

Game ini adalah *game mobil* berbasis *android*. Pada *game* ini terdapat empat jenis kartu *sharraf* dengan bentuk kata berbeda dan acak yang akan dijalankan dan disusun oleh pemain sesuai dengan kaidah ilmu *sharraf*. Metode penelitian dalam permainan ini adalah sebuah algoritma jaringan syaraf tiruan *backpropagation* untuk menentukan *level* pada *game* berdasarkan perpindahan kartu dan waktu yang telah ditentukan. Tingkat kesulitan pada setiap *level* adalah jenis kartu yang berbeda. Pada setiap *level* terdapat 80 kartu yang harus disusun.

3.1.1 Keterangan Umum *Game*

Game Sharraf Solitaire merupakan *game* edukasi yang dimainkan pada *smartphone android* dan dijadikan media untuk pengenalan dan pembelajaran kaidah ilmu *sharraf* khususnya pada *tashrif isthila*hi dalam ilmu *sharraf*. Sebagai alat untuk menjembatani pengenalan dan pembelajaran ilmu *sharraf*. Sasaran pengguna *game* ini adalah untuk umum dan memberikan edukasi tentang dasar ilmu *sharraf*. *Game* ini menerapkan permainan kartu modern yang terdapat pada *windows* yaitu *spider solitaire* menyusun kartu secara urut diimplementasikan sesuai dengan kaidah pembelajaran *sharraf*. *Game* ini dimainkan secara individu. *Game* ini memiliki 3 *level*. Perbedaan setiap *level* adalah *wazan* pada ilmu *sharraf* sebagai objek.

Sistem kemenangan akan ditentukan apabila pemain dapat mengurutkan atau menyusun kartu dengan benar di setiap *level*nya sesuai dengan kaidah ilmu *sharraf*. *Game* ini memiliki 3 *level*. Setiap *level* berisi kartu-kartu *sharraf* yang memiliki tingkat kesulitan berbeda-beda, Berikut penjelasannya :

- *Level 1* : 80 Kartu *Sharraf* dengan wazan فَعَلَ
- *Level 2* : 40 Kartu *Sharraf* dengan wazan فَعَلَ dan 40 Kartu *Sharraf* dengan wazan أَفْعَلَ
- *Level 3* : 20 Kartu *Sharraf* dengan wazan فَعَلَ , 20 Kartu *Sharraf* dengan wazan أَفْعَلَ , 20 Kartu *Sharraf* dengan wazan اِفْتَعَلَ dan 20 Kartu *Sharraf* dengan wazan فَعَّلَ

3.1.2 Story Board Game

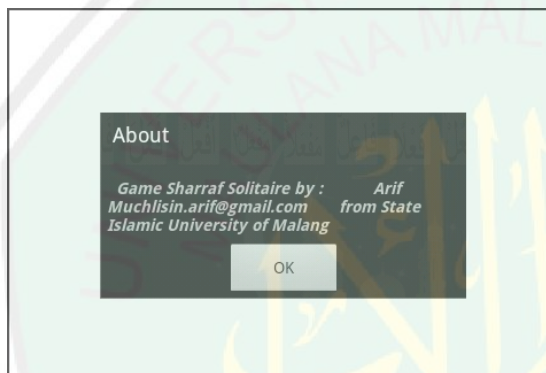
Berikut adalah gambar dari *storyboard game* yang akan ditunjukkan pada gambar 3.1 sampai 3.9 :



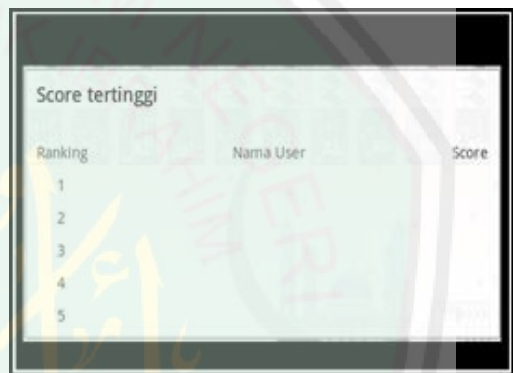
Gambar 3.1 Tampilan *Splash*



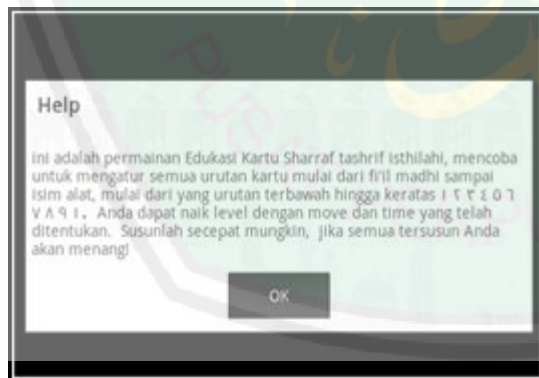
Gambar 3.2 Tampilan Menu



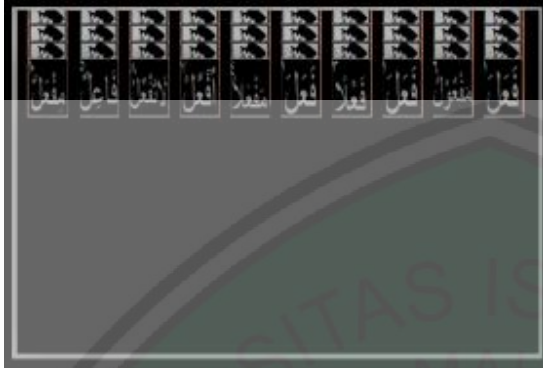
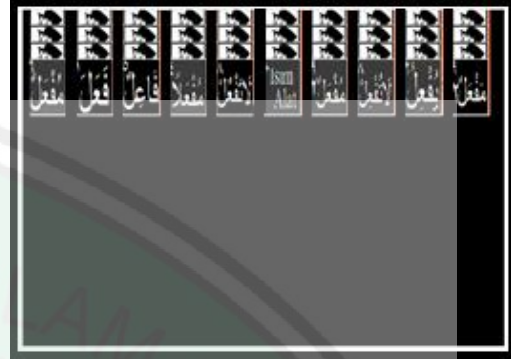
Gambar 3.3 Tampilan *About*



Gambar 3.4 Tampilan *Score*



Gambar 3.5 Tampilan *Help*

Gambar 3.6 *Level 1*Gambar 3.7 *Level 2*Gambar 3.8 *Level 3*Gambar 3.9 Permainan *Sharraf*

3.1.3 Penampilan Umum *Game*

Game ini dibangun dengan grafis 2 dimensi dengan rancangan yang menarik sehingga pengguna dapat menikmati permainan ini. Dalam permainan ini terdapat 4 konten selain grafis *background*.

3.1.4 Konten

A. Konten *wazan fa'ala* (ada di semua level)

فَاعِلٌ	مَفْعَلًا	فَعَلًا	يَفْعَلُ	فَعَلَ
مِفْعَلٌ	مَمْفَعَلًا	لَا تَفْعَلُ	أَفْعَلُ	مَمْفَعُولٌ

Gambar 3.10 Konten *wazan fa'ala*

B. Konten *wazan af'ala* (ada di level 2 dan level 3)

مِفْعَلٌ	مَمْفَعَلًا	أَفْعَالًا	يُفْعِلُ	أَفْعَلٌ
Isim Alat	مَمْفَعَلًا	لَا تَفْعِلُ	أَفْعِلُ	مَمْفَعَلٌ

Gambar 3.11 Konten *wazan af'ala*

C. Konten *wazan fa'lala* (ada di level 3)

مِفْعَلٌ	مَمْفَعَلًا	فَعْلَلَةٌ	يَفْعِلُّ	فَعْلَلٌ
Isim Alat	مَمْفَعَلًا	لَا تَفْعِلُّ	فَعْلِلٌ	مَمْفَعَلٌ

Gambar 3.12 Konten *wazan fa'lala*

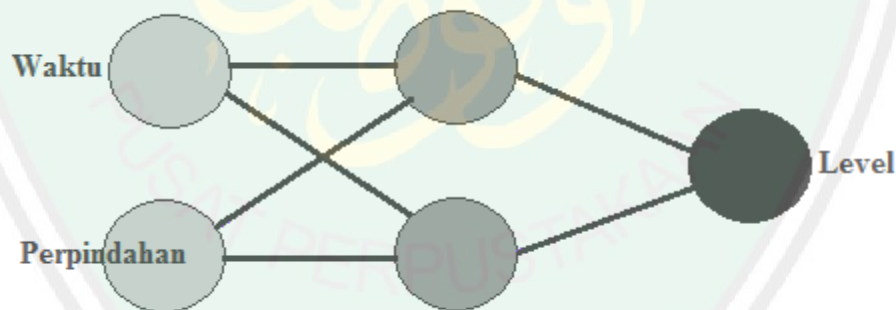
D. Konten *wazan ifta'ala* (ada di level 3)



Gambar 3.13 Konten *wazan ifta'ala*

3.2 Perancangan Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan yang ditanamkan pada aplikasi *game* ini adalah Metode *Neural Network Backpropagation* yang digunakan sebagai metode dalam menentukan *leveling* pada *game* berdasarkan 2 *inputan* yakni waktu dan perpindahan, dijelaskan pada gambar 3.14.



Gambar 3.14 Perancangan kecerdasan buatan

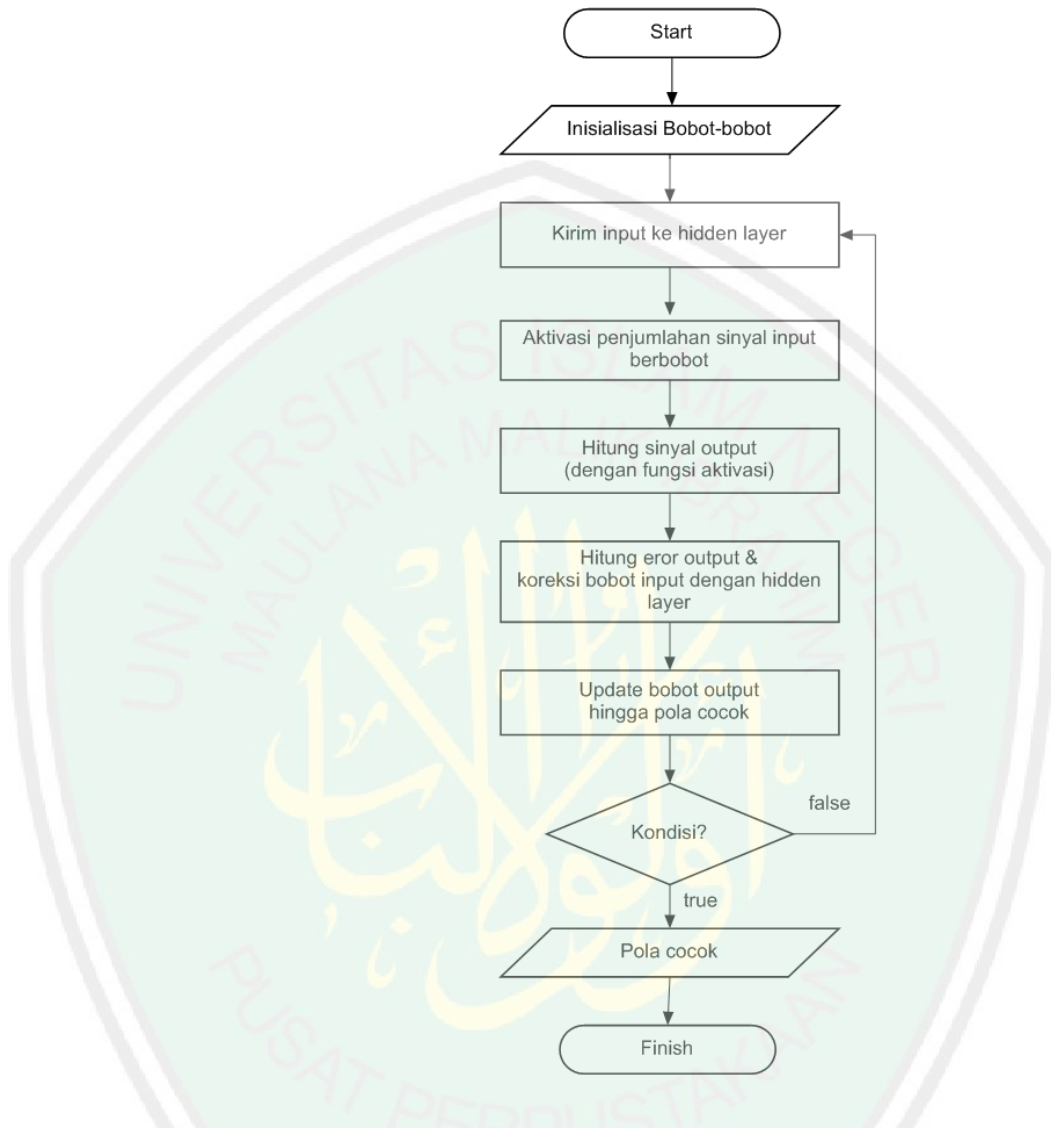
3.2.1 Perancangan Metode *Neural Network Backpropagation*

Perancangan metode terdapat pada *Leveling* yang menentukan apakah pemain bisa melanjutkan permainan ke *level* berikutnya atau malah turun dari *level* tersebut berdasarkan waktu dan jumlah perpindahan kartu, untuk

memperjelas algoritma dapat dibagi ke dalam 2 bagian yakni algoritma pelatihan dan algoritma aplikasi.

- Algoritma Pelatihan dijelaskan seperti berikut :
 - a. Memulai Permainan
 - b. Inisialisasi bobot-bobot.
 - c. Kirim sinyal *input* ke *hidden layer*.
 - d. Aktivasi terhadap penjumlahan sinyal *input* berbobot.
 - e. Hitung sinyal *output* dengan menerapkan fungsi aktivasi terhadap penjumlahan sinyal *input* berbobot.
 - f. Terima pola target, lalu hitung *error outputnya*, kirim ke lapisan bawahnya untuk mengkoreksi antara bobot dengan *hidden layer*.
 - g. Hitung kesalahan pada *hidden layer* untuk mengkoreksi bobot antara *input* dengan *hidden layer*.
 - h. *Update* bobot pada output hingga pola cocok.
 - i. Tes kondisi berhenti.

Dan dijelaskan dengan *flowchart* pada gambar 3.15



Gambar 3.15 Flowchart Algoritma pelatihan

Pola Pelatihan dijelaskan pada tabel 3.1

Tabel 3.1 Pola Pelatihan

Waktu (<i>Input</i>)	Perpindahan (<i>Input</i>)	Level (<i>Output</i>)
0 (waktu > 20 menit)	0 (perpindahan > 150 kali)	0 (Level 1)
0 (waktu > 20 menit)	1 (perpindahan <= 150 kali)	0 (Level 1)
1 (waktu <= 20)	0 (perpindahan > 150)	0 (Level 1)

menit)	kali)	
1 (waktu ≤ 20 menit)	1 (perpindahan ≤ 150 kali)	1 (Naik Level)

Mula-mula, nilai bobot diatur ke nilai acak: 0,62; 0,42; 0,55; -0,17 untuk *matrik* bobot 1 dan 0,35; 0,81 untuk *matrik* bobot 2.

Learning rate jaringan diatur ke 0,25 untuk memaksimalkan *looping*.

Berikutnya, nilai pola *input* (**0 1**) diatur ke neuron *layer input* (*output* dari *layer input* adalah sama dengan *inputnya*).

Neuron di *layer* tersembunyi diaktivasi:

$$\text{Input neuron tersembunyi 1: } 0 * 0,62 + 1 * 0,55 = 0,55$$

$$\text{Input neuron tersembunyi 2: } 0 * 0,42 + 1 * (-0,17) = -0,17$$

$$\text{Output neuron tersembunyi 1: } 1 / (1 + \exp(-0,55)) = 0,634135591$$

$$\text{Output neuron tersembunyi 2: } 1 / (1 + \exp(+0,17)) = 0,457602059$$

Neuron di *layer output* diaktivasi:

$$\text{Input neuron output: } 0,634135591 * 0,35 + 0,457602059 * 0,81 = 0,592605124$$

$$\text{Output neuron output: } 1 / (1 + \exp(-0,592605124)) = 0,643962658$$

Hitung nilai *error* dengan mengurangi *output* dari target: $0 - 0,643962658 = -0,643962658$

Setelah mendapatkan *error output*, lakukan *backpropagation*.

Dimulai dengan mengubah bobot pada *matrik* bobot 2:

$$\begin{aligned} \text{Perubahan bobot 1: } & 0,25 * (-0,643962658) * 0,634135591 * 0,643962658 * (1- \\ & 0,643962658) = \\ & -0,023406638 \end{aligned}$$

$$\begin{aligned} \text{Perubahan bobot 2: } & 0,25 * (-0,643962658) * 0,457602059 * 0,643962658 * (1- \\ & 0,643962658) = \\ & -0,016890593 \end{aligned}$$

$$\text{Bobot 1: } 0,35 + (-0,023406638) = 0,326593362$$

$$\text{Bobot 2: } 0,81 + (-0,016890593) = 0,793109407$$

Sekarang ubah *matrik* bobot 1:

$$\begin{aligned} \text{Perubahan bobot 1: } & 0,25 * (-0,643962658) * 0 * 0,634135591 * (1-0,634135591) \\ & = 0 \end{aligned}$$

$$\begin{aligned} \text{Perubahan bobot 2: } & 0,25 * (-0,643962658) * 0 * 0,457602059 * (1-0,457602059) \\ & = 0 \end{aligned}$$

$$\begin{aligned} \text{Perubahan bobot 3: } & 0,25 * (-0,643962658) * 1 * 0,634135591 * (1-0,634135591) \\ & = \\ & -0,037351064 \end{aligned}$$

$$\begin{aligned} \text{Perubahan bobot 4: } & 0,25 * (-0,643962658) * 1 * 0,457602059 * (1-0,457602059) \\ & = \\ & -0,039958271 \end{aligned}$$

$$\text{Bobot 1: } 0,62 + 0 = 0,62$$

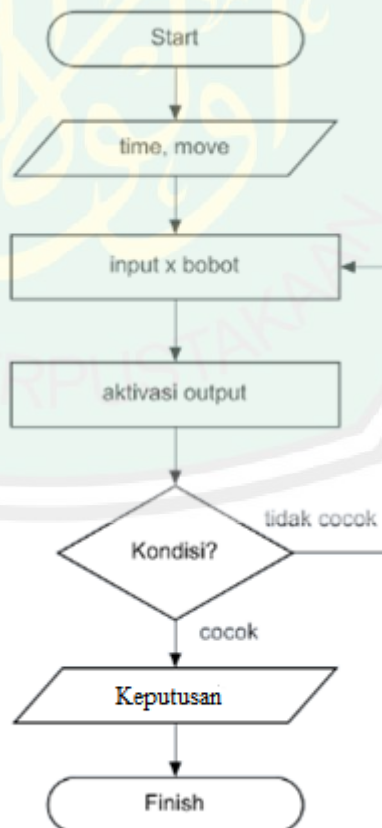
$$\text{Bobot 2: } 0,42 + 0 = 0,42$$

$$\text{Bobot 3: } 0,55 + (-0,037351064) = 0,512648936$$

$$\text{Bobot 4: } -0,17 + (-0,039958271) = -0,209958271$$

Pola input (**0 1**) telah dipropagasi melalui jaringan. Prosedur yang sama digunakan untuk pola *input* yang lain, tetapi dengan nilai bobot yang sudah diubah.

- Algoritma Aplikasi dengan *input* waktu dan perpindahan kartu dan *output*nya adalah penentuan *level*, dijelaskan seperti berikut :
 - a. Masukkan *input*an waktu dan perpindahan kartu.
 - b. Masukkan bobot dari *training*.
 - c. Aktivasi nilai *output*.
 - d. Apabila sesuai dengan data *training* maka naik *level*, jika tidak maka tetap di *level* tersebut. Dan dijelaskan dengan *flowchart* pada gambar 3.16



Gambar 3.16 *Flowchart* Algoritma aplikasi

3.3 Perancangan Aplikasi *Game*

Berikut ini menjelaskan tentang perencanaan aplikasi *game* berupa *interface*, *flowchart* aplikasi *game*, dan kebutuhannya.

3.3.1 Perancangan Leveling

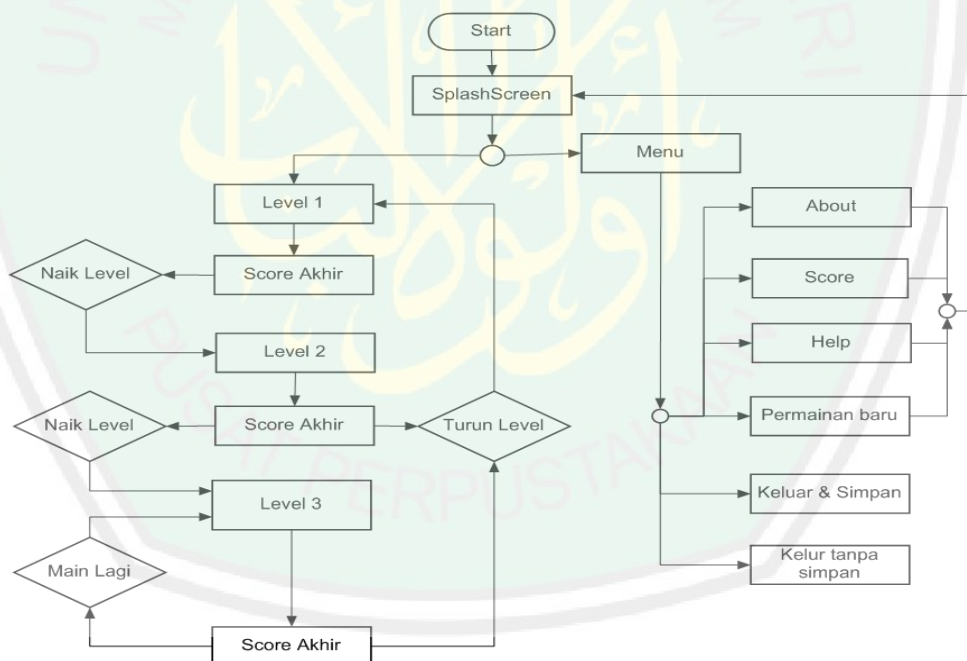
Leveling pada *game* ditentukan pada waktu dan perpindahan kartu setelah menyelesaikan permainan. Apabila *user* sudah berhasil menyusun keseluruhan kartu, apakah *user* bisa melanjutkan *level* selanjutnya ataukah tetap pada *level* tersebut tergantung pada berapa lama waktu yang ditempuh pada *level* tersebut dan banyaknya perpindahan kartu dengan ketentuan seperti berikut :

- i. Dari *level* 1, Apabila *user* menyelesaikan permainan dengan waktu lebih dari 20 menit, maka tetap di *level* 1. Apabila *user* menyelesaikan permainan dengan perpindahan lebih dari 150 kali maka tetap di *level* 1.
- ii. Dari *level* 1, Apabila *user* menyelesaikan permainan dengan waktu kurang dari sama dengan 20 menit dan *user* menyelesaikan permainan dengan perpindahan kurang dari sama dengan 150 kali maka *user* naik ke *level* 2.
- iii. Dari *level* 2, Apabila *user* menyelesaikan permainan dengan waktu lebih dari 20 menit, maka turun ke *level* 1. Apabila *user* menyelesaikan permainan dengan perpindahan lebih dari 150 kali, maka turun ke *level* 1.
- iv. Dari *level* 2, Apabila *user* menyelesaikan permainan dengan waktu kurang dari sama dengan 20 menit dan *user* menyelesaikan permainan dengan perpindahan kurang dari sama dengan 150 kali maka *user* naik ke *level* 3.

- v. Di *level 3*, Apabila *user* menyelesaikan permainan dengan waktu lebih dari 20 menit, maka turun ke *level 1*. Apabila *user* menyelesaikan permainan dengan perpindahan lebih dari 150 kali, maka turun ke *level 1*.
- vi. Di *level 3*, Apabila *user* menyelesaikan permainan dengan waktu kurang dari sama dengan 20 menit dan *user* menyelesaikan permainan dengan perpindahan kurang dari sama dengan 150 kali maka *user* akan menang.

3.3.2 Perancangan Alur/ *Flowchart Game*

Perancangan *game* ini dijelaskan dengan menggunakan *flowchart*. Seperti pada gambar 3.17



Gambar 3.17 *Flowchart* Permainan

Flowchart diatas menjelaskan tentang keseluruhan alur yang akan ditanamkan pada *game* ini. Pada awal *user* membuka permainan dan dihadapkan pada *splashscreen* kemudian menuju permainan *level 1*, setelah selesai menyelesaikan setiap *level*, akan menunjukkan *score* akhir dan dapat naik *level*

maupun turun ke *level* 1. Pada setiap *level* terdapat menu *about*, *score*, *help*, permainan baru, keluar dan simpan, keluar tanpa simpan.

3.3.3 Kebutuhan Sistem

Pada bagian spesifikasi kebutuhan system ini, diulas tentang kebutuhan *system* perangkat keras (*hardware*) maupun perangkat lunak (*software*) yang mendukung dalam pembuatan maupun saat pengoperasian aplikasi.

A. Kebutuhan Perangkat Keras (*Hardware*)

Kebutuhan perangkat keras yang digunakan pada proses pembuatan *game* ini adalah sebagai berikut : *Processor* Corei3, RAM (*Random Acces Memory*) 3 GB, VGA *Intel(R) HD Graphics* Total Memory 1275 MB, *Hardisk* 300 GB, LCD resolusi 1366 x 768 (32bit) (60Hz), Keyboard, Mouse, *Mobile phone Android* minimum *android* versi 2.3.

B. Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan untuk mendukung pembuatan *game* antara lain:

1. JDK, *Java Development Kit* adalah program *development environment* untuk menulis *Java applets* dan aplikasi.
2. ADT, *Android Development Tools* adalah *plugin* Eclipse IDE untuk membangun aplikasi *Android*.
3. *Inkscape*, adalah perangkat lunak yang bersifat *freeware* yang digunakan untuk membuat desain grafis pada *game*.
4. *Paint* adalah program *graphics painting* sederhana yang terintegrasi dengan hampir seluruh versi *Microsoft Windows*, sejak perilisan

pertamanya. Sering dirujuk sebagai *MS Paint* atau *Microsoft Paint*. Program ini dapat membuka dan menyimpan gambar dalam berbagai format, yaitu BMP, JPEG, GIF, PNG, dan TIFF.

3.3.4 Kebutuhan *Device* Minimum Pemain

Berikut ini merupakan daftar spesifikasi *device* yang harus dipenuhi untuk memainkan *game* ini.

Tabel 3.2 Kebutuhan *Device* Pemain

Kebutuhan	Spesifikasi Minimum	Spesifikasi Rekomendasi
Versi <i>Android</i>	2.3	2.3 keatas
GPU	<i>Power</i> VR SGX 100	<i>Power</i> VR SGX 530
Memori	1 GB	2GB

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada bab ini membahas tentang implementasi dari perencanaan yang telah dibuat. Serta melakukan pengujian terhadap *game* untuk mengetahui apakah *game* tersebut telah berjalan sesuai dengan yang diharapkan.

4.2 Implementasi Algoritma *Neural Network Backpropagation*

Algoritma *Neural Network Backpropagation* dalam *game* ini akan diimplementasikan dengan *source code* seperti berikut :

```
//data learning
public static void main(String haripinter[]){
    GameActivity BP = new GameActivity();
    BP.init_static();
    BP.learn_static();
    double ax[] = {0,0};
    double bx[] = {0,1};
    double cx[] = {1,0};
    double dx[] = {1,1};
    BP.test(ax);
    BP.test(bx);
    BP.test(cx);
    BP.test(dx);
}
//save score, and win dialog
public void gameOver(int score) {
    newScore = score;
    showDialog(WIN_DIALOG_ID);
    menit=mUrutan.menit;
    detik=mUrutan.detik;
    double isimove;
    double isitime;
    if (move > 10){
        isimove=0;
    }else{
        isimove=1;
    }
    if (menit > 1){
        isitime=0;
    }else{
```

```

        isitime=1;
    }
    if(isimove==0 && isitime==0){
        index=0;
    }else if(isimove==0 && isitime==1){
        index=1;
    }else if(isimove==1 && isitime==0){
        index=2;
    }else if(isimove==1 && isitime==1){
        index=3;
    }
    init_static();
    learn_static();
    double idi[]={isimove,isitime};
    test(idi);
}

// kondisi
double x[][];
// target
double t[];

int unit_input;
int unit_hidden;
int unit_output;

// learning rate
double alfa;
// maximum loop
int maxloop;
// target error
double ERR;
// last error
double ERX;

// bobot input - hidden
double v[][];
// bobot bias - hidden
double v0[];
// bobot hidden - output
double w[][];
// bobot bias - output
double w0[];

void init_static(){
    double init_x[][] = {{0,0},{0,1},{1,0},{1,1}}; //kondisi
    double init_t[] = {0,0,0,1}; //target
    int init_uinput = init_x[index].length;
    int init_uhidden = 4;
    int init_uoutput = 1;
    double init_alfa = 0.25; //learning
    int init_maxloop = 1000;
    double init_ERR = 0.01; //target error
    double init_v[][] = {{0.62,0.42,0.55,-0.17},{0.62,0.42,0.55,-

```

```

0.17}}; //bobot input-hidden
double init_v0[] = {1,1,1,1}; //bobot bias hidden
double init_w[][] = {{0.35,0.81,0.35,0.81}}; //bobot hidden
output
double init_w0[] = {1}; //bobot bias output

this.x = init_x;
this.t = init_t;
this.unit_input = init_uinput;
this.unit_hidden = init_uhidden;
this.unit_output = init_uoutput;
this.alfa = init_alfa;
this.maxloop = init_maxloop;
this.ERR = init_ERR;
this.v = init_v;
this.v0 = init_v0;
this.w = init_w;
this.w0 = init_w0;
}

void learn_static(){
double data[][] = this.x;
double target[] = this.t;
int jumlah_data = data.length;
int jumlah_input = this.unit_input;
int jumlah_hidden = this.unit_hidden;
int jumlah_output = this.unit_output;
// do it for learn
int loop = 0;
this.maxloop = 1000;

do{
for(int h=0; h<jumlah_data; h++){
double z[] = new double[data.length];
for(int j=0; j<jumlah_hidden; j++){
//sigma xi vij
double z_in[] = new double[z.length];
double jum_xv=0;
for(int i=0; i<jumlah_input; i++){
double tmp=x[h][i]*v[i][j];
jum_xv=jum_xv+tmp;
}
z_in[j] = v0[j]+jum_xv;
z[j] = 1/(1+(double)Math.exp(-z_in[j]));
}

//y_in dan y (output)
double y[] = new double[jumlah_output];
for(int k=0; k<jumlah_output; k++){
double y_in[] = new double[y.length];
double jum_zw=0;
for(int j=0; j<jumlah_hidden; j++){
double tmp=z[j]*w[k][j];
jum_zw=jum_zw+tmp;
}
}
}
}

```

```

    }
    y_in[k]=w0[k]+jum_zw;
    y[k]=1/(1+(double)Math.exp(-y_in[k]));
}

//error output dan delta bias dan delta bobot
double Err_y[] = new double[jumlah_output];
double Aw[][] = new
double[this.w.length][this.w[0].length];
double Aw0[] = new double[this.w0.length];
for(int k=0; k<jumlah_output; k++){
    //error output
    Err_y[k]=(t[h]-y[k])*y[k]*(1-y[k]);
    for(int j=0; j<jumlah_hidden; j++){
        //delta bobot h0
        Aw[k][j] = alfa*Err_y[k]*z[j];
        //delta bias h0
        Aw0[k] = alfa*Err_y[k];
    }
}

//hidden dan delta bias dan delta bobot
double Err_in[] = new double[jumlah_hidden];
double Err_z[] = new double[jumlah_hidden];
double Av[][] = new
double[this.v.length][this.v[0].length];
double Av0[] = new double[this.v0.length];
for(int j=0; j<jumlah_hidden; j++){
    double tmp=0;
    for(int k=0; k<jumlah_output; k++){
        tmp = tmp + (Err_y[k]*this.w[k][j]);
    }
    // eror sebelum output / setelah hidden
    Err_in[j]=tmp;
    // eror hidden (t[h]-y[k])*y[k]*(1-y[k]);
    Err_z[j]=Err_in[j]*(z[j])*(1-z[j]);

    for(int i=0; i<jumlah_input; i++){
        //delta bobot iH
        Av[i][j]=this.alfa*Err_z[j]*this.x[h][i];
    }
    //delta bias hidden
    Av0[j]=this.alfa*Err_z[j];
}

//update bobot bias output
for(int j=0; j<jumlah_hidden; j++){
    for(int k=0; k<jumlah_output; k++){
        this.w[k][j]=this.w[k][j]+Aw[k][j];
    }
}
for(int k=0; k<jumlah_output; k++){
    this.w0[k]=this.w0[k]+Aw0[k];
}

```

```

        //update bobot bias hidden
        for(int i=0; i<jumlah_input; i++){
            for(int j=0; j<jumlah_hidden; j++){
                this.v[i][j]=this.v[i][j]+Av[i][j];
            }
        }
        for(int j=0; j<jumlah_hidden; j++){
            this.v0[j]=this.v0[j]+Av0[j];
        }
    }
    loop++;
}while(is_stop()>this.ERR && loop<this.maxloop);
System.out.println("err : "+ERX);
System.out.println("loop : "+loop);
}

//penentuan berhenti atau lanjut
double is_stop(){
    int jumlah_input = this.unit_input;
    int jumlah_hidden = this.unit_hidden;
    int jumlah_output = this.unit_output;
    int jumlah_data = this.x.length;
    double akumY=0;

    // z_in dan z
    for(int h=0; h<jumlah_data; h++){
        double z[] = new double[jumlah_hidden];
        for(int j=0; j<jumlah_hidden; j++){
            // sigma xi vij
            double z_in[] = new double[z.length];
            double jum_xv=0;
            for(int i=0; i<jumlah_input; i++){
                double tmp=this.x[h][i]*this.v[i][j];
                jum_xv=jum_xv+tmp;
            }
            z_in[j]=this.v0[j]+jum_xv;
            z[j]=1/(1+(double)Math.exp(-z_in[j]));
        }
    }

    // y_in dan y (output)
    double y[] = new double[jumlah_output];
    for(int k=0; k<jumlah_output; k++){
        double y_in[] = new double[y.length];
        double jum_zw=0;
        for(int j=0; j<jumlah_hidden; j++){
            double tmp=z[j]*this.w[k][j];
            jum_zw=jum_zw+tmp;
        }
        y_in[k]=this.w0[k]+jum_zw;
        y[k]=1/(1+(double)Math.exp(-y_in[k]));
        akumY += Math.pow((t[h]-y[k]),2);
    }
}

```

```

    }
    double E = akumY/this.x[0].length;
    ERX = akumY;
    return E;
}

void test(double data[]){
    int jumlah_input = this.unit_input;
    int jumlah_hidden = this.unit_hidden;
    int jumlah_output = this.unit_output;

    // pada hidden
    double z[] = new double[jumlah_hidden];
    for(int j=0; j<jumlah_hidden; j++){
        double z_in[] = new double[z.length];
        double tmp = 0;
        for(int i=0; i<data.length; i++){
            tmp = tmp + (data[i] * this.v[i][j]);
        }
        z_in[j] = this.v0[j] +tmp;
        z[j] = 1/(1+(double)Math.exp(-z_in[j]));
    }

    //pada out
    double y[] = new double[jumlah_output];
    for(int k=0; k<jumlah_output; k++){
        double y_in[] = new double[y.length];
        double tmp = 0;
        for(int j=0; j<jumlah_hidden; j++){
            tmp = tmp + z[j] * this.w[k][j];
        }
        y_in[k] = this.w0[k] +tmp;
        y[k] = 1/(1+(double)Math.exp(-y_in[k]));

        if(y[k]>0.5)
            y[k]=1;
        else
            y[k]=0;
        System.out.println((double)y[k]);
        Toast.makeText(getApplicationContext(), ""+(int)y[k],
Toast.LENGTH_LONG).show();

        if ((double)y[k]==0){
            GameActivity.Level=1;
            SharedPreferences pref =
this.getPreferences(MODE_PRIVATE);
            mUrutan.nsuit= pref.getInt("Level",
GameActivity.Level);
            mUrutan.initReady();
            mUrutan.setNewGame();
            mUrutan.stoptimer();
        }
        if ((double)y[k]==1){
            SharedPreferences pref =

```

```

this.getPreferences(MODE_PRIVATE);
        mUrutan.nsuit= pref.getInt("Level",
    MainActivity.Level+=1);
        mUrutan.initReady();
        mUrutan.setNewGame();
        mUrutan.stoptimer();
    }

}
}

```

Berikut akan dijelaskan penggunaan *method* pada tabel 4.1

Tabel 4.1 Keterangan Algoritma *Neural Network Backpropagation*

No	Method	Keterangan
1.	public static void main(String haripinter[])	Method yang menampung <i>data learning</i>
2.	public void gameOver(int score)	Menunjukkan bahwa metode diletakkan pada <i>leveling</i> setelah akhir permainan dan menunjukkan nilai waktu dan perpindahan sebagai <i>inputan</i>
3.	void init_static(){	Digunakan untuk menginisialisasi bobot-bobot disetiap unit baik dalam <i>input, hidden, dan output</i> .
4.	void learn_static()	Proses perhitungan <i>learning</i> , mengaktifasi bobot-bobot koreksi bobot-bobot, menghitung <i>error output</i> dengan fungsi aktivasi, <i>update</i> bobot-bobot.
5.	double is_stop()	Sebuah kondisi dari <i>method</i> sebelumnya dan menentukan apakah proses berhenti ataukah lanjut.
6.	void test(double data[]){	Mencocokkan kondisi pada <i>hidden</i> dan <i>output</i> dari bobot-bobot yang diperoleh dan menghasilkan <i>output</i> berupa keputusan.
7.	Toast. <i>makeText</i> (getApplicationContext(), ""+(int)y[k], Toast.LENGTH_LONG).show();	Menampilkan hasil <i>output</i> berupa angka 0 atau 1 yang diperoleh setelah akhir permainan.
8.	if ((double)y[k]==0){	Menunjukkan jika <i>output</i> = 0

	<code>GameActivity.Level=1;</code>	maka akan menuju <i>level 1</i>
9.	<pre> if ((double)y[k]==1){ SharedPreferences pref = this.getPreferences(MODE_PRIVATE); mUrutan.nsuit= pref.getInt("Level", GameActivity.Level+1); </pre>	Menunjukkan jika <i>output = 1</i> maka akan naik <i>level</i>

4.3 Implementasi Aplikasi Game

Berikut adalah tampilan *game* yang telah selesai dibuat



Gambar 4.1 Tampilan *Splashscreen*



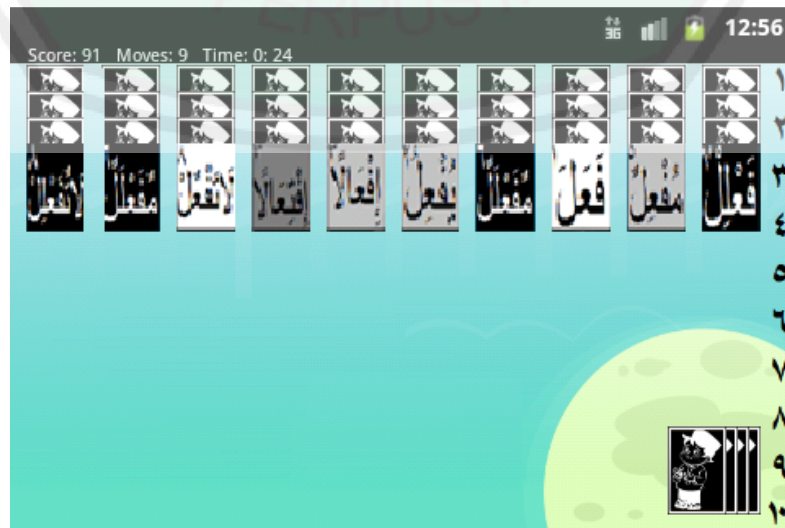
Gambar 4.2 Tampilan Menu



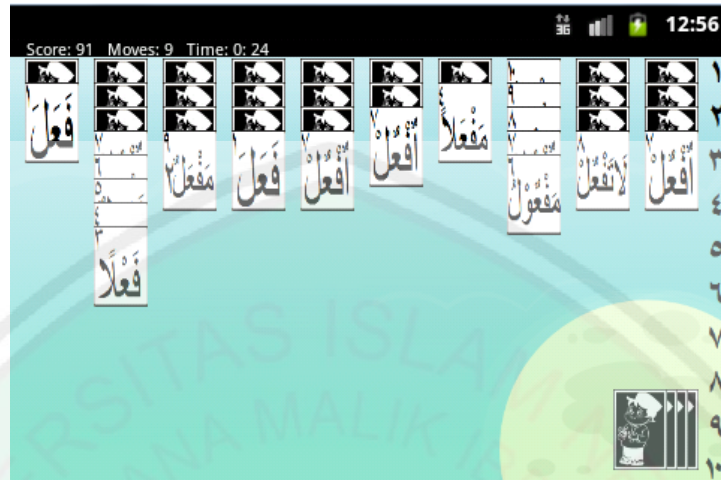
Gambar 4.3 Tampilan Level 1



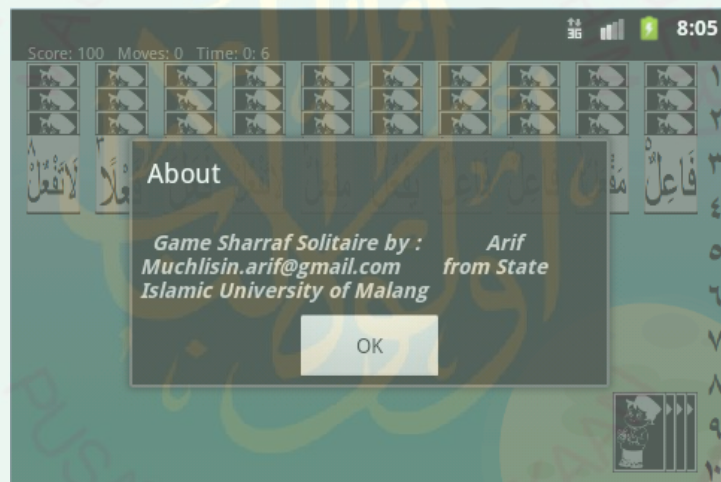
Gambar 4.4 Tampilan Level 2



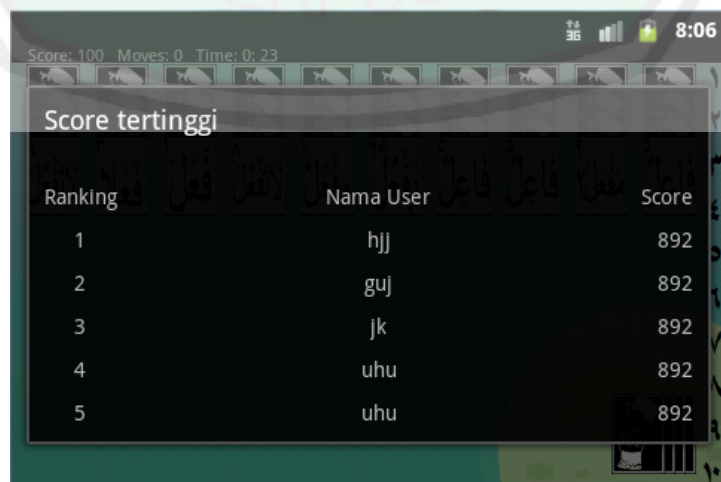
Gambar 4.5 Tampilan Level 3



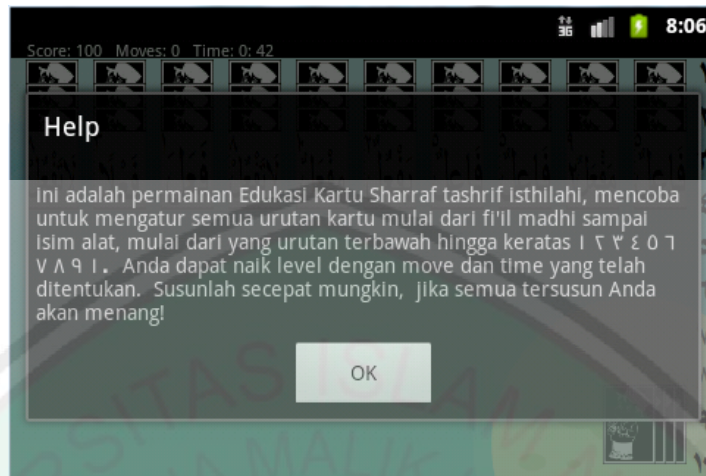
Gambar 4.6 Tampilan Permainan



Gambar 4.7 Tampilan About



Gambar 4.8 Tampilan Score



Gambar 4.9 Tampilan *Help*



Gambar 4.10 Tampilan *Win*

4.4 Pengujian Algoritma *Neural Network Backpropagation*

Berikut akan dijelaskan tentang hasil pengujian dari algoritma *Neural Network Backpropagation* dalam bentuk tabel pada tabel 4.2 dan tabel 4.3

Tabel 4.2 Hasil Pengujian 1 (dengan bobot 0 dan *learning* 1)

Aplikasi	<i>Game Sharraf Solitaire</i>	Keterangan
Algoritma	<i>Neural Network Backpropagation</i>	
<i>Input</i>	{{0,0},{0,1},{1,0},{1,1}}	
<i>Output</i>	0, 1	
<i>Target</i>	{0,0,0,1}	
<i>Learning</i>	1	

Bobot <i>input-hidden</i>	{0,0,0,0} , {0,0,0,0}	
Bobot <i>hidden-output</i>	{0,0,0,0}	
Waktu Eksekusi	3 detik	
<i>Maxloop</i>	1000	
<i>Loop</i>	320	
<i>Error</i>	0.01988834865416264	
<i>Error Output</i>	<i>Error Output</i> : -0.0037037109752488043 <i>Error Output</i> : -0.0037037109752488043 <i>Error Output</i> : -0.0037037109752488043 <i>Error Output</i> : -0.0037037109752488043 <i>Error Output</i> : 0.011178073763772467 <i>Error Output</i> : 0.011178073763772467 <i>Error Output</i> : 0.011178073763772467 <i>Error Output</i> : 0.011178073763772467	nilai <i>error</i> terakhir sebelum <i>output</i>
<i>Update Bobot</i>	<i>Update Bobot</i> : -3.142299262984355 <i>Update Bobot</i> : -3.142299262984355 <i>Update Bobot</i> : -3.142299262984355 <i>Update Bobot</i> : -3.142299262984355 <i>Update Bobot</i> : -3.1408511043286653 <i>Update Bobot</i> : -3.1408511043286653 <i>Update Bobot</i> : -3.1408511043286653 <i>Update Bobot</i> : -3.1408511043286653	<i>update bobot</i> terakhir sebelum <i>output</i>
<i>Output</i>	0 0 0 1	

Tabel 4.3 Hasil Pengujian 2 (dengan bobot *random* dan *learning* 0,25)

Aplikasi	<i>Game Sharraf Solitaire</i>	Keterangan
Algoritma	<i>Neural Network Backpropagation</i>	
<i>Input</i>	{{0,0},{0,1},{1,0},{1,1}}	
<i>Output</i>	0 , 1	
<i>Target</i>	{0,0,0,1}	
<i>Learning</i>	0,25	
Bobot <i>input-hidden</i>	{{0.62,0.42,0.55,-0.17},{0.62,0.42,0.55,-0.17}}	
Bobot <i>hidden-output</i>	{{0.35,0.81,0.35,0.81}}	
Waktu Eksekusi	5 detik	
<i>Maxloop</i>	1000	
<i>Loop</i>	1000	
<i>Error</i>	0.03515275712176967	
<i>Error Output</i>	<i>Error Output : -0.007732434707460579</i> <i>Error Output : -0.007732434707460579</i> <i>Error Output : -0.007732434707460579</i> <i>Error Output : -0.007732434707460579</i> <i>Error Output : 0.016084985756781703</i> <i>Error Output : 0.016084985756781703</i> <i>Error Output : 0.016084985756781703</i> <i>Error Output : 0.016084985756781703</i>	nilai <i>error</i> terakhir sebelum <i>output</i>
<i>Update Bobot</i>	<i>Update Bobot : 0.9096405365748391</i> <i>Update Bobot : 1.9416736568169473</i> <i>Update Bobot : 0.7217730305775777</i> <i>Update Bobot : -6.2590389423942865</i> <i>Update Bobot : 0.9132538352850043</i> <i>Update Bobot : 1.9450894930971991</i> <i>Update Bobot : 0.7253279718616239</i> <i>Update Bobot : -6.258444961390617</i>	<i>update bobot</i> terakhir sebelum <i>output</i>
<i>Output</i>	0 0 0 1	

4.5 Pengujian Game

Berikut akan dijelaskan tentang hasil pengujian dari *game system* pada beberapa *smartphone* dalam bentuk tabel pada tabel 4.4

Tabel 4.4 Hasil Pengujian Game

No	Versi	GPU	Processor	RAM	Keterangan
1.	v4.1.2 (<i>Jelly Bean</i>)	Mali – 300	1 GHz	512 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
2.	v4.0 (<i>Ice Cream Sandwich</i>)	Mali – 400	1 GHz	512 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
3.	v4.3 (<i>Jelly Bean</i>)	Power VR SGX5 44MP2	Dual-core 1.2 GHz	1 GB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik
4.	V2.3.5(<i>Gingerbread</i>)	Mali – 300	ARM v7 Processor	512 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
5.	v4.1 (<i>Jelly Bean</i>)	Broadcom	832 MHz	290 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
6.	v4.1 (<i>Jelly Bean</i>)	Adreno 203	Dual Core 1,2 GHz	512 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
7.	v4.2 (<i>Jelly Bean</i>)	Power VR	Dual Core 1,2 GHz	1 GB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
8.	v4.1.2 (<i>Jelly Bean</i>)	Adreno 200	1 GHz Cortex-A5	512 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
9.	v4.1 (<i>Jelly Bean</i>)	Adreno 320	Dual-core 1.7 GHz Krait	1 GB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
10.	v4.1 (<i>Jelly Bean</i>)	Adreno 203	Qualcomm MSM 8625 Dual Core 1GHz ARMv7	512 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.

11.	v4.2.2 (<i>Jelly Bean</i>)	<i>Power VR SGX 531</i>	ARMv7 Processor (VFPv3, NEON)	231.4 MB RAM	Sistem berjalan dengan baik. Semua tombol berfungsi. Tampilan Baik.
12.	V2.3.3(<i>Gingerbread</i>)	<i>Adreno 200</i>	600 MHz ARM 11	168 MB RAM	Sistem dapat berjalan tetapi dengan lambat. Semua tombol berfungsi. Tampilan Baik.

4.6 Integrasi Dalam Islam

Game edukasi merupakan salah satu alternatif untuk proses pembelajaran yang menyenangkan, sebagaimana belajar merupakan suatu hal yang sangat penting bagi umat manusia. Kewajiban untuk selalu menuntut ilmu telah difirmankan Allah dan disabdakan Nabi Muhammad. Apalagi dengan pembelajaran yang berkaitan erat dengan ilmu *sharraf* yang merupakan bagian dari *Ulumul Arabiyyah*, yang bertujuan untuk menjaga dari kesalahan pengucapan maupun tulisan. Untuk menjaga lisan dari kesalahan bahasa *Al-Quran* dan *Hadits* di mana keduanya adalah primer (pokok) ajaran Islam dan kandungan kedua sumber ajaran Islam ini harus diamankan.

Sehubungan dengan itu, Untuk mengamalkan kandungan di dalam bahasa arab, maka haruslah dipelajari dan dikuasai ilmu tentang bahasa arab, diantaranya ilmu *Nahwu* dan *Sharraf*. Alasan penulis mengatakan begitu kerana seperti yang kita ketahui pada masa sekarang banyak yang kurang faham dengan masalah-masalah agama disebabkan mempelajarinya dari buku-buku terjemahan yang terkadang-kadang masih banyak kesalahannya daripada teks sehingga menyebabkan kesalahan penerapan tersebut dalam kehidupan akan lebih baik bagi kita untuk mempelajari ilmu agama langsung dari sumber yang menggunakan

bahasa arab yaitu *Al-Qur'an*, melalui mempelajari ilmu *Nahwu* dan *Sharraf* kerana keduanya adalah bapak dan ibunya ilmu.

Dengan bermain *game* ini, pemain diharapkan mampu memahami dasar ilmu *sharraf tashrif isthilahi* melalui *Shighot*(bentuk kata) yang meliputi *Fi'il Madli*, *Fi'il Mudhori'*, *Masdar G.Mim*, *Masdar Mim*, *Isim Fa'il*, *Isim amf'ul*, *Fi'il Amar*, *Fi'il Nahi*, *Isim Z.Makan*, *Isim Alat* dan dapat meningkatkan minat belajar bahasa arab.



BAB V

PENUTUP

5.1 Kesimpulan

Hasil kesimpulan yang didapat setelah meninjau hasil analisis, perancangan, implementasi, serta hal-hal yang telah dibahas pada bab-bab sebelumnya bahwasannya aplikasi *game* ini lebih mendukung *android* versi *Jelly Bean*. Algoritma *Neural Network Backpropagation* merupakan algoritma pembelajaran tersupervisi yang dapat melakukan *learning* terhadap *inputan* dan menghasilkan *output* berupa sebuah keputusan dan cocok untuk diterapkan dalam *game* ini dengan tujuan agar setiap permainan dapat memainkan sesuai dengan *level* pemahamannya dan tidak jenuh untuk memahami *wazan* yang terdapat pada masing-masing *level*, pemain juga dapat merasa tertantang untuk menyelesaikan seluruh *level*. Dalam menjalankan proses perhitungan algoritma *Neural Network Backpropagation* membutuhkan waktu 3-5 detik hingga mendapatkan sebuah keputusan. Hal ini dikarenakan proses perhitungan propagasi balik yang terus berjalan dan *update* bobot hingga mendapatkan pola yang cocok untuk menghasilkan *output*. Dari hasil pengujian algoritma *Neural Network Backpropagation* lebih efektif menggunakan bobot *random* dan *learning* 0,25 untuk memaksimalkan *looping* pada saat proses perhitungan.

5.2 Saran

Penulis sadar, dalam pembuatan *game* ini masih memiliki kekurangan yang nantinya perlu untuk dilakukan pengembangan untuk menjadikan aplikasi

semakin baik dan diminati, saran-saran merupakan unsur penting dalam perwujudannya. Saran-saran untuk pengembangan *game sharraf solitaire* antara lain :

1. Menambah jumlah *level* permainan dan *wazan* sehingga permainan menjadi lebih menarik dan pemain mendapatkan lebih banyak pembelajaran.
2. Mengingat bahwa *game* ini adalah *game* edukasi yang diterapkan sebagai media pembelajaran, jadi diharapkan dalam pengembangan nantinya *game* ini bisa menjadi sarana belajar oleh siswa-siswa MI, MTs, MA maupun *madrasah diniyah*.

Daftar Pustaka

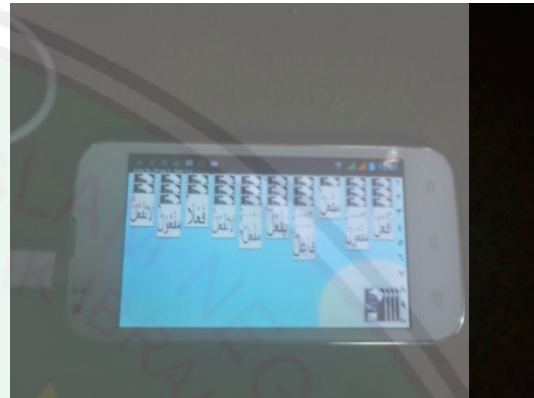
- K.H.Moch.Anwar.2000.*Revisi Ilmu Shorof Terjemahan Mantan Kailani dan Nadham Al-Maksud berikut penjelasannya*.Bandung : Sinar Baru Algensingo.
- Departemen Agama RI.2009. *Al-Qur'an Terjemah*.Surabaya : Fajar Mulya.
- Maftuh Ahnan.1999. *Metode Belajar ILMU SHOROF*.Surabaya: Terbit Terang.
- Riyadhy B.N, Danaparamita Muhammad, Dan paramita Muhammad, Nephyana Aldhy, Galih Afan ST, M.Si.2012.*Aplikasi Game Garuda Berbasis Android*.Jakarta : BINUS University
- Heri.2011.*Backpropagation*.Semarang : Universitas Stikubank.
- William Berkson, John Wettersten.2003.*Psikologi Belajar dan Filsafat Ilmu Karl Popper. Terjemahan oleh Ali Noer Zaman*.Yogyakarta : Qalam.
- Husni M.2011.*Membangun Aplikasi Multimedia Edukatif Sebagai Alat Bantu Ajar Ilmu Shorof Berdasarkan Metode Pembelajaran "Shorof Krapyak"*.Yogyakarta : STIMIK AMIKOM
- Puspitaningrum Diyah.2006.*Jaringan Saraf Tiruan*.Yogyakarta : ANDI
- Hidayanto Achmad, Isnanto R.Rizal, Kurnia Dian.2008. *Identifikasi Tanda-Tangan Menggunakan Jaringan Saraf Tiruan Perambatan-Balik (Backpropagation)*.Semarang : Universitas Diponegoro
- Hanief M., Annisa Metri, Mudandri Narendi, Suryadi Kadarsyah.2009.*Prediksi Masa Studi Sarjana Dengan Artificial Neural Network*.Bandung : Institut Teknologi Bandung
- Safaat H, Nazruddin. 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.

Lampiran

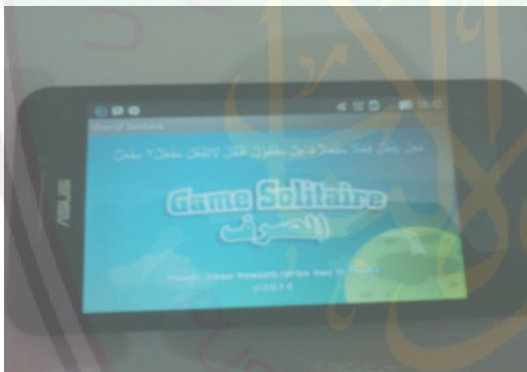
Pengujian Game Pada Smartphone



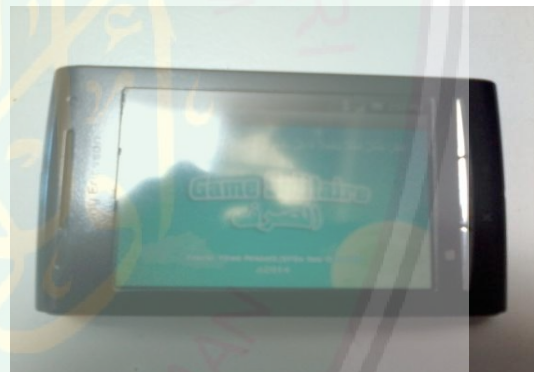
v4.1.2



v4.0

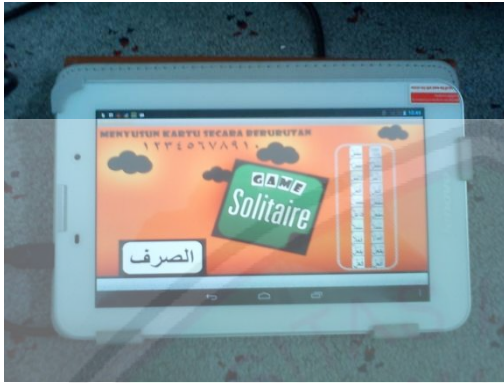


v4.3

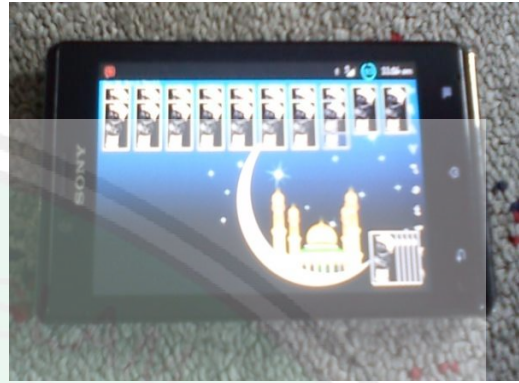


v2.3.3

Pengujian Game Pada Smartphone Dengan Desain Berbeda



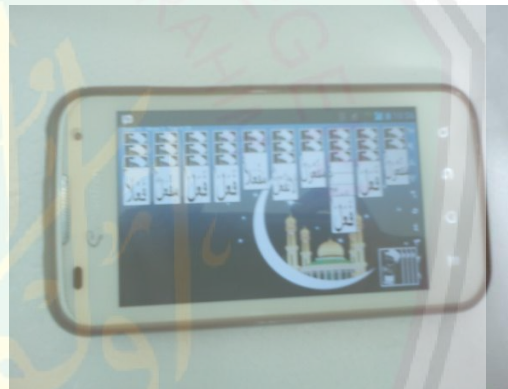
v4.2



v4.1.2



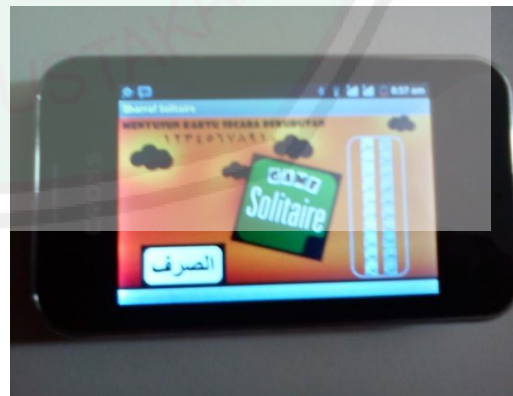
v4.1



v4.1



v4.1



v2.3.5

Tampilan Di Google play

