

ALGORITMA PENCARIAN *BEAM A UNTUK MENENTUKAN
RUTE ARAH GERAK POTONGAN KATA PADA
GAME PUZZLE BAHASA ARAB**

SKRIPSI

Oleh:
AHMAD WAHYU ROSYADI
10650031



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

ALGORITMA PENCARIAN *BEAM A UNTUK MENENTUKAN
RUTE ARAH GERAK POTONGAN KATA PADA
GAME PUZZLE BAHASA ARAB**

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:
AHMAD WAHYU ROSYADI
10650031**

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

LEMBAR PERSETUJUAN

ALGORITMA PENCARIAN *BEAM A UNTUK MENENTUKAN
RUTE ARAH GERAK POTONGAN KATA PADA
GAME PUZZLE BAHASA ARAB**

SKRIPSI

Oleh :

Nama : Ahmad Wahyu Rosyadi
NIM : 10650031
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji :

Tanggal 08 Juli 2014

Dosen Pembimbing I

Dosen Pembimbing II

Fresy Nugroho, MT
NIP. 19710722 201101 1 001

Fachrul Kurniawan, M. MT
NIP. 19771020 200901 1 001

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

**ALGORITMA PENCARIAN BEAM A* UNTUK MENENTUKAN
RUTE ARAH GERAK POTONGAN KATA PADA
GAME PUZZLE BAHASA ARAB**

SKRIPSI

Oleh :
Ahmad Wahyu Rosyadi
NIM. 10650031

Telah dipertahankan di depan dewan penguji skripsi
dan dinyatakan diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana komputer (S.Kom)
Tanggal : 11 September 2014

Susunan Dewan Penguji:

Tanda Tangan

- | | | | | | |
|------------------------------|--|---|--|--|---|
| 1. Penguji Utama | : <u>Yunifa Miftachul Arif, M.T</u> | (| | |) |
| | NIP. 19830616 201101 1 004 | | | | |
| 2. Ketua Penguji | : <u>Dr. Suhartono, M.Kom</u> | (| | |) |
| | NIP. 19680519 200312 1 001 | | | | |
| 3. Sekretaris Penguji | : <u>Fresy Nugroho, M.T</u> | (| | |) |
| | NIP. 19710722 201101 1 001 | | | | |
| 4. Anggota Penguji | : <u>Fachrul Kurniawan, M.MT</u> | (| | |) |
| | NIP. 19771020 200901 1 001 | | | | |

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

HALAMAN PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama : Ahmad Wahyu Rosyadi
NIM : 10650031
Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika
Judul Penelitian : Algoritma pencarian beam a* untuk menentukan rute arah gerak potongan kata pada game puzzle bahasa arab

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan 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, 16 September 2014
Yang membuat pernyataan,

Ahmad Wahyu Rosyadi
NIM. 10650031

MOTTO

**“Tidak perlu menjadi yang terbaik
untuk melakukan yang terbaik”**

-----dan-----

**“Ada satu hal yang bisa mengalahkan bakat, yaitu
keinginan dan semangat yang kuat”**



PERSEMBAHAN

Dengan penuh rasa syukur serta mengharap rahmat dan ridlo Allah SWT.

Kupersembahkan karya ini kepada **Bapak Iskandar** dan **Ibu Aminah Ulfah** yang selalu mendukung dan mencurahkan kasih sayang, perhatian, Doa, dan bimbingannya dalam setiap langkahku.

Serta adik-adikku **Maya, Dinda** dan **Dimas**, kakek dan nenek **Pakde Mad, Pakde Wujud, Mak Tuni, Makde Ya**, semua keluarga dan tante-tante dan paman-pamanku khususnya **Yuk Pit** dan **Cak Bas** yang selalu mendukungku.

Terima kasih untuk Bapak dan Ibu Dosen **Pak Fresy, Pak Fachrul, Pak Cahyo, Pak Zainal, Bu Ila Ifawati**. Terima kasih juga untuk dosen dan juga atasan di Lab **Pak Aziz, Pak Gianto, Bu Ivana**, dan juga admin TI **Mbak Rusiana** dan **Mbak Citra**

Teman-teman yang menemaniku daftar sampai masuk dan belajar di UIN **Hasan** dan **Anas**. Teman-teman ma'had **Habibi, Ropik, Shofa**.

Teman-teman yang selalu ngerjain tugas bareng, **Fuad, M_Bagus, Rifa, Muiz, Haris**.

Teman-teman kontrakan **Rizqi, Ichal Da'u', Dapid, Tipul, Hizbun, Catur, Syaifudin, Barep, Coko, Pakde Triyono, Irfan, Akmal, Febri**.

Teman seperjuangan **Anggoro, Arif, Ipul, Ardi, Ghulam, Cahyo, Rizqi Senyap, Kiki, Listya, Yudis**.

Teman-teman **PKPBA E3, ANTIKOMA, INFINITY**.

Praktikan-praktikanku **Fauzan, Ulfah, Anis, Emil, Dyan, Yasir, Taufiqi, Hakim** dan lainnya.

Keluarga dan teman-teman **LPM Mentaraman Shonhaji, Harum, Wulan, Lilik, Ayu, Linda**.

Dan juga untuk teman-teman sejurusan semuanya, Bapak dan Ibu Dosen dan semuanya yang pernah menemani hari-hariku di UIN.

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT yang telah melimpahkan rahmat, hidayah serta inayah-Nya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “*Algoritma Pencarian Beam A* Untuk Menentukan Rute Arah Gerak Potongan Kata Pada Game Puzzle Bahasa Arab*”.

Shalawat serta salam semoga tercurah kepada Nabi Agung Muhammad SAW yang telah membimbing kita dari gelapnya kekufuran menuju yang terang-benderang yakni agama Islam.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, oleh karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan 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 Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Fresy Nugroho, M. T. dan Fachrul Kurniawan M. MT selaku dosen pembimbing I dan II yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan dalam pengerjaan skripsi ini.
5. Segenap sivitas akademika Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
6. Bapak dan Ibuku tercinta, adikku dan seluruh keluarga besar yang senantiasa memberikan doa dan restunya kepada penulis dalam menuntut ilmu serta dalam menyelesaikan skripsi ini.
7. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan, penulis ucapkan terima kasih yang sebesar-besarnya.

Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna, untuk itu peneulis selalu menerima segala kritik dan saran dari pembaca. Harapan penulis, semoga karya ini bermanfaat bagi kita semua.

Malang, Juli 2014

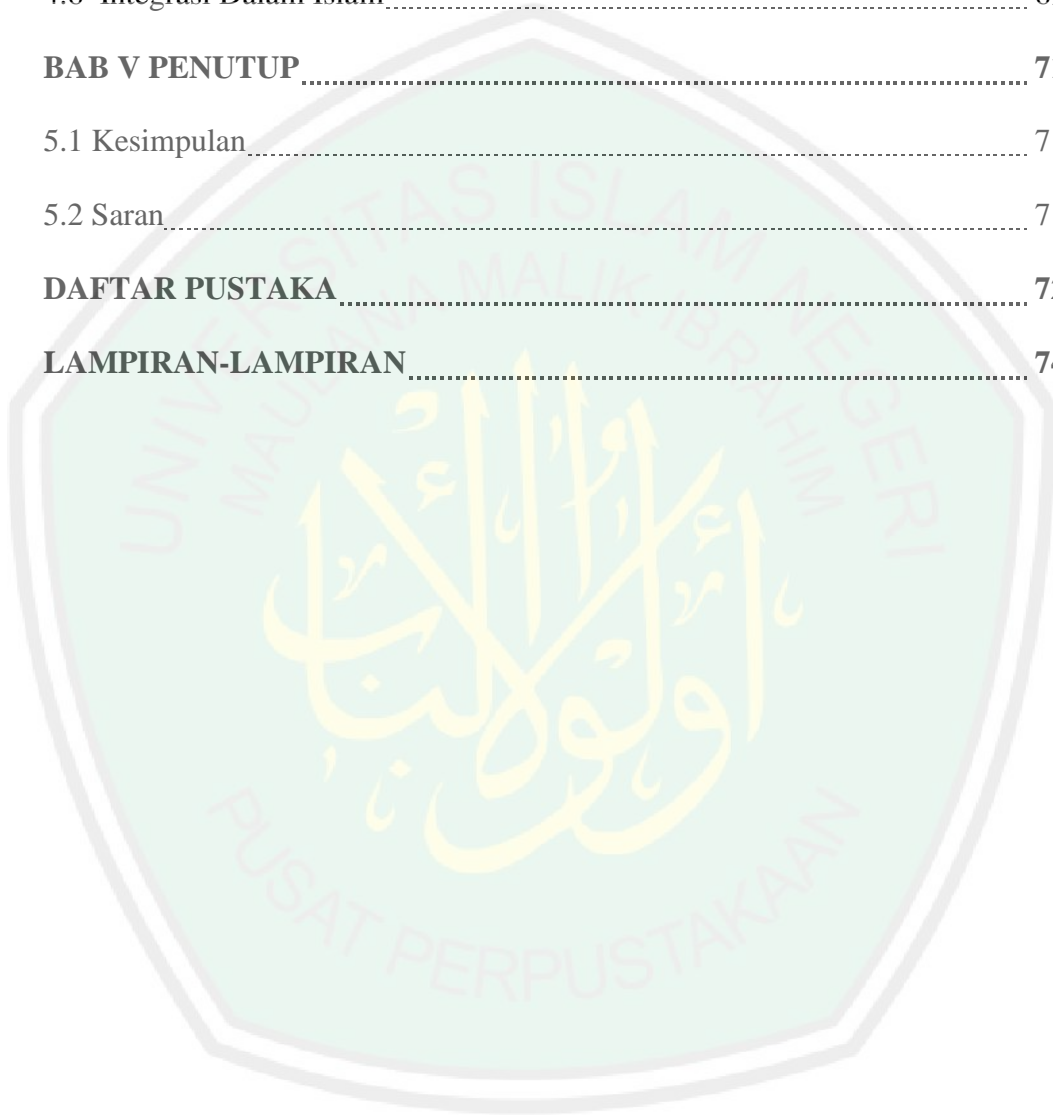
Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
HALAMAN PERNYATAAN	v
MOTTO	vi
PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xvi
ABSTRAK	xvii
ABSTRACT	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terkait.....	6

2.2 Game	7
2.2.1 Pengertian Game	7
2.2.2 Komponen Dasar Game	8
2.2.3 Jenis-jenis Game	9
2.3 Android	9
2.4 Algoritma Beam A* (BA*)	10
BAB III ANALISA DAN PERANCANGAN SISTEM	22
3.1 Metode Penelitian	22
3.2 Desain Sistem	24
3.2.1 Deskripsi Aplikasi	24
3.2.2 Storyline	24
3.2.3 Gameplay	24
3.2.4 Cara mengatur posisi buah	27
3.2.5 Scoring	27
3.2.6 Konten-konten pada aplikasi	28
3.2.7 Skenario Game	31
3.2.8 Storyboard	33
3.2.9 Penerapan algoritma Beam A*	36
3.3 Kebutuhan Sistem	37
BAB IV HASIL DAN PEMBAHASAN	38
4.1 Implementasi Algoritma Beam A* (BA*)	38
4.2 Pengujian Algoritma Beam A*	39
4.3 Implementasi aplikasi game	57

4.4 Pengujian Game.....	62
4.5 Kuesioner Pengujian Game.....	64
4.6 Integrasi Dalam Islam.....	69
BAB V PENUTUP.....	71
5.1 Kesimpulan.....	71
5.2 Saran.....	71
DAFTAR PUSTAKA.....	72
LAMPIRAN-LAMPIRAN.....	74



DAFTAR GAMBAR

Gambar 2.1 Bagan Arsitektur android.....	10
Gambar 2.2 Langkah 1 pencarian arah gerak huruf ۞.....	13
Gambar 2.3 Langkah 2 pencarian arah gerak huruf ۞.....	13
Gambar 2.4 Langkah 3 pencarian arah gerak huruf ۞.....	14
Gambar 2.5 Langkah 4 pencarian arah gerak huruf ۞.....	14
Gambar 2.6 Langkah 5 pencarian arah gerak huruf ۞.....	15
Gambar 2.7 Langkah 6 pencarian arah gerak huruf ۞.....	15
Gambar 2.8 Langkah 7 pencarian arah gerak huruf ۞.....	16
Gambar 2.9 Langkah 8 pencarian arah gerak huruf ۞.....	16
Gambar 2.10 Langkah 9 pencarian arah gerak huruf ۞.....	17
Gambar 2.11 Langkah 10 pencarian arah gerak huruf ۞.....	17
Gambar 2.12 Langkah 11 pencarian arah gerak huruf ۞.....	18
Gambar 2.13 Langkah 12 pencarian arah gerak huruf ۞.....	18
Gambar 2.14 Langkah 13 pencarian arah gerak huruf ۞.....	19
Gambar 2.15 Langkah 14 pencarian arah gerak huruf ۞.....	19
Gambar 2.16 Langkah 15 pencarian arah gerak huruf ۞.....	20
Gambar 2.17 Langkah 16 pencarian arah gerak huruf ۞.....	20
Gambar 2.18 Rute arah gerak huruf ۞.....	20
Gambar 2.19 Langkah 1 pencarian arah gerak huruf ”م”.....	21
Gambar 2.20 Langkah 2 pencarian arah gerak huruf ”م”.....	21
Gambar 2.21 Rute arah gerak huruf ”م”.....	21
Gambar 3.1 Tahap-tahap penelitian.....	23

Gambar 3.2 <i>Flowchart</i> untuk <i>Game puzzle</i> bahasa Arab.....	26
Gambar 3.3 <i>Finite state machine</i> permainan.....	31
Gambar 3.4 Tampilan misi.....	33
Gambar 3.5 Tampilan <i>puzzle</i> awal.....	33
Gambar 3.6 Tampilan <i>game</i> menampilkan perintah.....	34
Gambar 3.7 Tampilan <i>puzzle</i> tersusun.....	34
Gambar 3.8 Tampilan terjemahan bahasa Indonesia.....	34
Gambar 3.9 Tampilan semut mengumpulkan <i>puzzle</i> yang benar.....	35
Gambar 3.10 Tampilan <i>puzzle</i> tambahan.....	35
Gambar 3.11 Tampilan kata bonus.....	35
Gambar 3.12 Tampilan kata bonus yang diacak.....	36
Gambar 3.13 Tampilan koleksi level.....	36
Gambar 4.1 flowchart implementasi Beam A*(menyusun kata dan mengacak kata).....	38
Gambar 4.2 Matriks awal.....	40
Gambar 4.3 Langkah1, 2, dan 3.....	40
Gambar 4.4 Langkah 4 (selesai).....	40
Gambar 4.5 Matriks awal.....	42
Gambar 4.6 Langkah 1,2, dan 3.....	42
Gambar 4.7 Langkah 4 dan 5 (selesai).....	42
Gambar 4.8 Matriks awal.....	44
Gambar 4.9 Langkah 1, 2, dan 3.....	44
Gambar 4.10 Langkah 4 (selesai).....	44

Gambar 4.11 Matriks awal.....	46
Gambar 4.12 Langkah 1, 2, dan 3.....	46
Gambar 4.13 Langkah 4 (selesai).....	46
Gambar 4.14 Matriks awal.....	51
Gambar 4.15 Langkah 1,2, dan 3.....	51
Gambar 4.16 Langkah 4, 5, dan 6 (selesai).....	52
Gambar 4.17 Proses pengacakan puzzle penalti.....	55
Gambar 4.18 Tampilan menu.....	57
Gambar 4.19 Tampilan <i>help</i>	57
Gambar 4.20 Tampilan pilihan level.....	57
Gambar 4.21 Tampilan misi.....	58
Gambar 4.22 Tampilan daftar kata beserta terjemahan.....	58
Gambar 4.23 Tampilan level 1.....	58
Gambar 4.24 Tampilan level 2.....	59
Gambar 4.25 Tampilan level 3.....	59
Gambar 4.26 Tampilan jawaban benar.....	59
Gambar 4.27 Tampilan <i>penalty</i>	60
Gambar 4.28 Tampilan menggunakan bantuan atau <i>hint</i>	60
Gambar 4.29 Tampilan ketika <i>hint</i> berjalan.....	60
Gambar 4.30 Tampilan ketika semua kata telah dijawab.....	61
Gambar 4.31 Tampilan kata bonus.....	61
Gambar 4.32 Tampilan <i>puzzle</i> kata bonus.....	61
Gambar 4.33 Tampilan ketika permainan telah selesai.....	62

Gambar 4.34 Percobaan Hp.....	63
Gambar 4.35 Percobaan Hp.....	63
Gambar 4.36 Percobaan Hp.....	64
Gambar 4.37 Form kuesioner.....	65
Gambar 4.38 Hasil pernyataan 1.....	66
Gambar 4.39 Hasil pernyataan 2.....	66
Gambar 4.40 Hasil pernyataan 3.....	67
Gambar 4.41 Hasil pernyataan 4.....	67
Gambar 4.42 Hasil pernyataan 5.....	68
Gambar 4.43 Hasil pernyataan 6.....	68
Gambar 4.44 Hasil pernyataan 7.....	69

DAFTAR TABEL

Tabel 3.1 Tabel materi kata bahasa Arab.....	28
Tabel 4.1 Tabel hasil pengujian algoritma beam a* huruf خ.....	39
Tabel 4.2 Tabel hasil pengujian algoritma beam a* huruf ج.....	40
Tabel 4.3 Tabel hasil pengujian algoritma beam a* huruf ا.....	43
Tabel 4.4 Tabel hasil pengujian algoritma beam a* huruf ح.....	45
Tabel 4.5 Tabel hasil pengujian algoritma beam a* huruf ة.....	47
Tabel 4.6 Tabel kecepatan pengujian algoritma Beam A* untuk menyusun kata.....	52
Tabel 4.7 Tabel hasil pengujian algoritma beam a* untuk mengacak kata yang memiliki panjang 3 huruf.....	53
Tabel 4.8 Tabel hasil pengujian algoritma Beam A* untuk menyelesaikan kata penalty yang diacak.....	54
Tabel 4.9 Tabel kecepatan pengujian algoritma Beam A* untuk menyelesaikan kata penalty yang diacak.....	56

ABSTRAK

Rosyadi, Ahmad Wahyu. 2014. **Algoritma Pencarian *Beam A** Untuk Menentukan Rute Arah Gerak Potongan Kata Pada *Game Puzzle Bahasa Arab***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Fresy Nugroho, MT (II) Fachrul Kurniawan, M. MT

Kata Kunci: *Bahasa Arab, Game Slide Puzzle, Path finding, Beam A**

Bahasa Arab adalah salah satu bahasa yang sangat penting di dunia ini, karena bahasa Arab telah dijadikan sebagai bahasa resmi oleh beberapa negara seperti Saudi Arabia, Kuwait, Irak, Syria, Libanon, dan Mesir. Saat ini telah banyak metode-metode atau cara untuk pembelajaran bahasa Arab. Namun hal terpenting yang harus ditumbuhkan adalah rasa senang terhadap bahasa Arab itu sendiri. Untuk mencapai tujuan tersebut telah diciptakan berbagai media alternatif sebagai penunjang dalam proses pembelajaran, salah satunya adalah *game*. *Game* ini merupakan *slide puzzle game* yaitu pemain bertugas untuk menyusun huruf-huruf yang teracak menjadi kata bahasa Arab kemudian pemain memilih terjemahannya dalam bahasa Indonesia.

Berdasarkan hasil pengujian dapat disimpulkan bahwa algoritma *Beam A** cocok digunakan untuk menentukan rute arah gerak potongan kata karena algoritma ini dapat menemukan rute dengan waktu yang cukup cepat yakni 59,438 milisecond per huruf dan dengan keberhasilan mencapai 100%. Selain itu penggunaan algoritma *Beam A** untuk mengacak kata dalam *game* ini terbukti mampu untuk mengacak kata yakni 100% berhasil walaupun untuk kasus dengan jumlah 7 huruf masih menghasilkan acakan yang kurang optimal dikarenakan acakan bisa disusun lebih cepat dari batas geseran.

ABSTRACT

Rosyadi, Ahmad Wahyu. 2014. **Beam A* Search Algorithm To Determine Direction Motion of word pieces In Arabic Puzzle Game.** Thesis. Informatics Engineering Programme Faculty of Science and Technology. The State of Islamic University Maulana Malik Ibrahim Malang.

Promotor: (I) Fresy Nugroho, MT
(II) Fachrul Kurniawan, M. MT

Arabic is one of the most important languages in the world, because Arabic has been used as an official language by some countries such as the State of Saudi Arabia, Kuwait, Iraq, Syria, Lebanon, and Egypt. There are many ways or methods of learning Arabic. But the most important thing that must be grown is the pleasure of the Arabic language itself. To achieve it, a variety of alternative learning media has been created as a support in the learning process, one of which is gaming. This game is a slide puzzle game that player must arrange the scrambled letters into words and then the player chooses Arabic translation in Indonesian.

Based on the test results it can be concluded that the Beam A * algorithm suitable for determining the direction of motion of the fragments because it can find a route with a fast enough time 59,438 milliseconds per letter and achieve 100% success. In addition, this algorithm to scramble the words in this game proved to be able to scramble the words 100% successful even for cases with 7 letters the result is less than optimal because the scrambled can be arranged faster than the boundary shear.

Keywords: Arabic language, Game Slide Puzzle, Path finding, Beam A*

مخلص

رشادي، أحمد وحي. 2014. خوارزمية طلب الشعار A* لتعيين جهة تحريك من قطعة الكلمة في لعب اللغة العربية. البحث. شعبة تقنيات المعلوماتية كلية العلوم والتك نولوجيا جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج.

المشرق: (1) فرشى نوكر وهو، الماجستير(2) فخر الكرنياوان، الماجستير

الكلمة الرئيسية: اللغة العربية، لعب لغز الشريحة، النتيجة مسار، شعار A*

اللغة العربية هي إحدى اللغة الأهمية فى العالم، لأنها قد جعلها لغة رسمية بعض البلاد المملكة السعودية، وكويت، وعرق، وسوريا، ولبنان، ومصر مثالا. كثرت المناهج او الكيفيات لتعليم اللغة العربية الآن. لكن الأهم هي ان تزرع شعر الفرغ للغة العربية نفسها. لكي تصل الهدف تصنع وسيلة الخيارية مساعدا لعملية التعليمية، أحدها اللعب. إن اللعب لغز الشريحة هي اللاعب يوظف ان يرتب أحرفه بغير نظام لى تصير كلمة اللغة العربية فيختار ترجمها في اللغة الإندونيسية.

تأسيسا بنتيجة الإمتحان تلخص أن خوارزمية الشعار A* توافق أن تستعمل لتعيين جهة تحريك لقطعة الكلمة لأنها تجد جهة بالوقت التي تكفى سرعتها يعنى تسعة وخمسون، وأربعمائة وثلاثة وثمانين ميلي ثانية حرفا بحرف وبنتيجتها تصل مائة فى المائة، وغير ذلك استعمالها فى هذا اللعب مثبتة بدليل أنها تستطيع كى لاتنظم الكلمة يعنى مائة فى المائة نتيجة، ولو فى مسألة عدد سبعة أحرف مادامت أن تنتج غير نظام التي ينقص أمثلها، لأنها تستطيع أن ترتب سرعتها فاضلا من حد إزاحتها .

BAB I

PENDAHULUAN

1.1 Latar Belakang

Bahasa Arab adalah salah satu bahasa yang sangat penting di dunia ini, karena bahasa Arab telah dijadikan sebagai bahasa resmi oleh beberapa negara seperti Negara Saudi Arabia, Kuwait, Irak, Syria, Jordania, Libanon, dan Mesir (Kusno, 2011). Bahasa Arab juga telah menjadi salah satu bahasa resmi di organisasi PBB (Perserikatan Bangsa-Bangsa) yang diresmikan pada tanggal 18 Desember 1973 (Henry, 2011). Selain itu bahasa Arab juga digunakan sebagai bahasa al-Qur'an, oleh karena itu bahasa Arab menjadi bahasa yang penting, khususnya bagi umat Islam.

Dikarenakan pentingnya bahasa Arab saat ini telah banyak lembaga-lembaga pendidikan baik formal maupun informal yang mengajarkan bahasa Arab. Namun untuk bisa bahasa Arab itu tidaklah mudah karena terdapat banyak aturan-aturan dan kata-kata bahasa Arab yang harus diketahui. Sehingga banyak yang menganggap bahasa Arab itu sebagai salah satu momok yang menakutkan.

Saat ini telah banyak metode-metode atau cara-cara untuk pembelajaran bahasa Arab. Namun hal terpenting yang harus ditumbuhkan adalah rasa senang terhadap bahasa Arab itu sendiri, karena kalau hal itu telah terwujud proses pembelajaran akan lebih mudah. Untuk mencapai tujuan tersebut telah diciptakan berbagai media-media alternatif sebagai penunjang dalam proses pembelajaran, tak terkecuali melalui media teknologi berbasis *mobile* yang saat ini semakin

banyak penggunanya, salah satunya adalah *handphone*, *handphone* yang dulu hanya bisa digunakan untuk mengirim/menerima pesan dan menerima/melakukan panggilan, saat ini telah berkembang menjadi *smartphone* yang memiliki banyak fungsi, diantaranya adalah internet, pemutar musik dan video, bermain *game*, alat untuk membaca e-book, dan lain sebagainya. Bahkan saat ini *smartphone* telah digunakan oleh sebagian orang untuk melakukan pekerjaannya. *Smartphone* saat ini dibedakan berdasarkan sistem operasinya, beberapa sistem operasi yang telah digunakan oleh *smartphone* adalah iOS, Android, Windows phone, dan Blackberry, Seri 40, Open WebOS, Sailfish, Ubuntu, Firefox, Tizen, Smartisan (Didno,2013).

Dari beberapa sistem operasi *smartphone* tersebut Android merupakan sistem operasi *smartphone* yang paling banyak digunakan di Indonesia. Berdasarkan hasil riset *statcounter* setahun terakhir tentang sistem operasi yang digunakan *mobile device* dan *tablet* di Indonesia. Android mengalami peningkatan yang cukup besar dan menempati urutan teratas dengan 46,88 %, dilanjutkan Series 40 dengan 20,29 %, SymbianOS dengan 9,19 %, sementara iOS hanya mencapai 6,12 % dan Blackberry dengan 4,51 %.

Menurut Herry S.W (2013) merek, harga, dan tipe ponsel Android yang bervariasi menjadi faktor penting penyebab pasar Android masih tinggi. Selain faktor-faktor tersebut Herry S.W juga menyatakan bahwa banyaknya aplikasi gratis juga mempengaruhi pasar Android, karena orang Indonesia lebih menyukai aplikasi yang gratis meskipun ponsel mereka tergolong mahal. Tidak hanya itu,

kemudahan dan praktisnya penggunaan *smartphone* Android membuat *smartphone* Android semakin banyak penggunanya.

Kemudahan dan praktisnya penggunaan *smartphone* Android mengakibatkan pengguna lebih suka untuk melakukan sebagian pekerjaan melalui *smartphone* mereka. Saat ini *smartphone* tidak hanya digunakan oleh orang dewasa, bahkan anak-anak kecil ataupun remaja sudah banyak yang menggunakan *smartphone* walaupun hanya untuk bermain *game* atau melakukan aktifitas ringan lainnya. Beragam *game* telah tersedia di internet, *game-game* tersebut bisa memberikan pengaruh terhadap perkembangan anak, *game* juga dapat mengembalikan dan mengembangkan motivasi untuk belajar (Randel,1991). Namun untuk *game* yang berjenis pendidikan masih belum terlalu banyak. Oleh karena itu penulis akan membuat *game puzzle* tentang kosa kata bahasa Arab untuk *smartphone* Android.

1.2 Rumusan Masalah

1. Apakah *game* ini dapat membantu pengguna untuk menghafal kosa kata bahasa Arab beserta artinya?
2. Apakah minat belajar bahasa Arab pengguna bisa bertambah dengan menggunakan *game* ini?

1.3 Batasan Masalah

1. Kata-kata bahasa Arab yang digunakan dalam *game* merupakan kata-kata yang terdapat dalam kitab al ‘Arabiyah baina yadaik jilid satu dari bab satu sampai bab tiga.
2. *Game* berbasis *mobile* dengan OS Android dengan versi 3.0 ke atas

1.4 Tujuan Penelitian

Untuk membuat *game puzzle* yang dapat membantu pengguna untuk menghafal kata-kata bahasa Arab beserta artinya sehingga dapat menambah perbendaharaan kata bahasa Arab bagi pengguna.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah diharapkan bisa membuat media belajar alternatif yang menarik sehingga minat untuk belajar bahasa Arab menjadi meningkat.

1.6 Sistematika Penulisan

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

BAB I PENDAHULUAN

Pendahuluan, berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penyusunan skripsi.

BAB II TINJAUAN PUSTAKA

Tinjauan pustaka berisi tentang teori-teori yang melandasi penyusunan skripsi.

BAB III ANALISA DAN PERANCANGAN SISTEM

Menganalisa kebutuhan sistem dalam proses perancangan sampai dengan pembuatan *game*.

BAB IV HASIL DAN PEMBAHASAN

Menjelaskan tentang hasil dari pengaplikasian *game* setelah *game* selesai dibuat.

BAB V PENUTUP

Berisi kesimpulan dan saran





BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Gonzalo Mejia dan Nicholas G. Odrey tentang *an approach using petri nets and improved heuristic search for manufacturing system scheduling* pada tahun 2006. Pada penelitian tersebut terbukti bahwa dengan menggunakan *Beam A* Search* dapat menghasilkan solusi dan waktu komputasi yang menjanjikan.

Penelitian yang dilakukan oleh Dimas Maulana Ibrahim pada tahun 2010 tentang penerapan algoritma *Breadth First Search* dan *Depth First Search* Pada Permainan *Slide Puzzle*. Kedua algoritma tersebut digunakan untuk menentukan jalur atau urutan gerak *puzzle* agar *puzzle* dapat diselesaikan jika pemain menyerah. Algoritma tersebut dapat menyelesaikan *puzzle* akan tetapi algoritma tersebut juga membutuhkan memori yang cukup besar.

Penelitian yang dilakukan oleh Chandra Usman dan Kenedy Candra pada tahun 2013 tentang rancang bangun *game slide puzzle* berbasis android menggunakan metode heuristik dengan teknik *Best First Search*. Metode tersebut digunakan sebagai pembantu pemain yang akan menunjukkan *puzzle* mana yang sebaiknya dipindahkan. Dan dalam pengaplikasiannya metode tersebut terbukti mampu menentukan jalur terpendek dan membantu pemain menyelesaikan *puzzle*.

Penelitian yang dilakukan oleh Aldy Mereyano Iqbal tentang analisis performasi algoritma *beam* pada pencarian rute terpendek di Kota Tasikmalaya

pada tahun 2013. Pada penelitian tersebut algoritma beam dapat menemukan hasil pencarian rute tergantung dengan nilai heuristik dan lebar beamnya.

2.2 Game

2.2.1 Pengertian *Game*

Game merupakan salah satu sarana untuk mengisi waktu luang. Banyak orang yang suka untuk memainkan *game* karena *game* memiliki tantangan atau tingkat kesulitan yang berbeda-beda. Saat ini *game* juga telah dikemas dengan efek dan grafis yang sangat baik, bahkan beberapa *game* telah dapat dimainkan oleh banyak orang sekaligus dan dalam tempat atau bahkan Negara yang berbeda-beda.

Berikut adalah beberapa kriteria permainan yang menyenangkan untuk dimainkan antara lain (Putra, Muslim, Naba, 2013):

- ✓ Pemain harus bersenang-senang, bukan programmer atau designer.
- ✓ Permainan dimulai dengan pembukaan yang bagus.
- ✓ Gameplay yang bagus adalah aliran keputusan yang menarik yang harus diselesaikan oleh pemain.
- ✓ Tempatkan pemain dalam posisi seorang pahlawan.

Proses pengembangannya (Putra, Muslim, Naba, 2013):

- Prototype dan bermain game untuk sementara waktu.
- Selanjutnya minta orang lain untuk mencoba permainan.
- Secara cermat mengamati pemain.
- Secara terus-menerus beri penghargaan pemain dalam permainan.

- Ujicoba ke penguji baru.
- Membuat permainan lebih sukar sampai pemain meminta ampun.

1.2.2 Komponen Dasar *Game*

Di dalam sebuah *game*, terdapat beberapa komponen dasar yang cukup penting (Putra, Muslim, Naba, 2013):

✓ Grafik

Grafik merupakan suatu hal yang penting dalam *game*, dimana grafik ini akan membuat *game* menjadi daya tarik user untuk memainkan *game*. Pada awalnya *game* menggunakan grafik monokrom dengan layar hitam putih sampai sekarang sudah menggunakan teknologi 3D, semakin bagus grafik maka *game* tersebut semakin seperti dalam dunia nyata.

✓ Suara

Suara menjadi komponen *game* karena dengan adanya suara pada *game* maka *game* tersebut akan menarik.

✓ AI (kecerdasan Buatan)

Kecerdasan buatan atau AI (artificial Intelligence) menjadi komponen *game* karena dengan hal tersebut *game* menjadi lebih menantang.

✓ Skenario *Game*

Skenario *game* menjadi komponen *game* karena si pemain seolah-olah bermain sesuai dengan petunjuk yang ada di *game* tersebut.

✓ *Multiplayer*

Multiplayer adalah sebuah mode dalam game dimana game dimainkan oleh pemain lainnya sehingga bermain game secara bersama-sama. Era ini multiplayer game cukup populer.

1.2.3 Jenis-jenis *Game*

Game mempunyai 9 jenis, yaitu *Edutainment, First Person Shooter, Real Time Strategy, Fighting, Adventure, Role Playing Game, Construction and Management Simulation, Vehicle Simulation, Leveling* (Putra, Muslim, Naba, 2013).

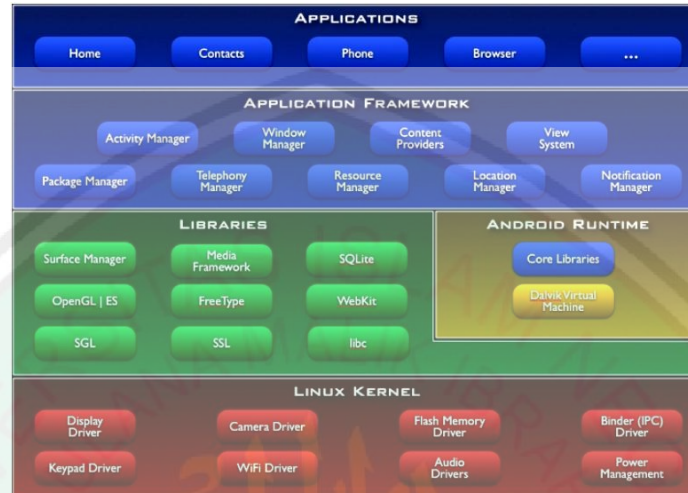
2.3 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi (Safaat, 2012). Android pertama kali dikembangkan oleh Android Inc yang kemudian pada tahun 2005 Android diakuisisi oleh google. Android merupakan *open source* sehingga banyak *developer* yang membuat aplikasi-aplikasi untuk Android baik berbayar maupun gratis.

Berikut ini merupakan arsitektur android (Safaat, 2012):

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

5) *Linux Kernel*



Gambar 2.1 Bagan Arsitektur android (Zechner, 2011)

1.4 Algoritma *Beam A** (*BA**)

Algoritma ini memberikan variasi pada *A**, yaitu dengan membatasi simpul yang bisa disimpan di dalam *OPEN*. Ketika jumlah simpul di *OPEN* sudah melebihi batas tertentu, maka simpul dengan nilai f terbesar akan dihapus. Sedangkan jumlah simpul di dalam *CLOSED* tetap dibiarkan tanpa batasan karena simpul yang di dalam *CLOSED* memang tidak mungkin dihapus. Dengan membatasi jumlah simpul di *OPEN*, maka pencarian menjadi lebih terfokus seperti sinar(*beam*). Sehingga variasi ini dinamakan *Beam A** (Suyanto, 2011).

Implementasi algoritma *BA** sama dengan *A**, tetapi ada sedikit fungsi tambahan untuk pengecekan dan penghapusan simpul terburuk di *OPEN*.

Langkah-langkah:

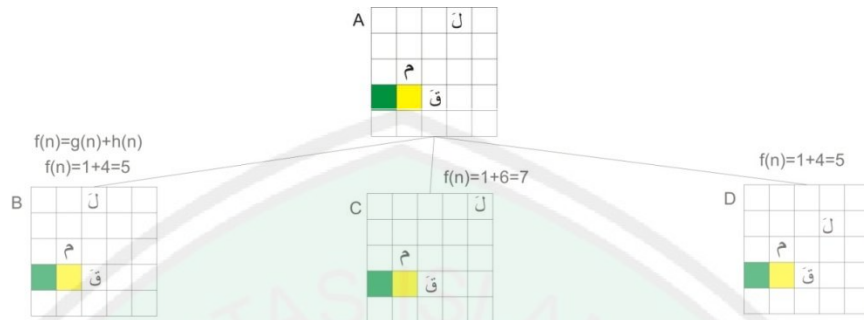
1. Membuat maksimal untuk open, Masukkan simpul awal ke array open, membuat array closed.

2. Loop sampai goal ditemukan atau sampai tidak ada simpul di dalam open.
3. Jika tidak ada data atau simpul di open maka gagal, tampilkan rute dan berhenti.
4. BestNode diisi dengan simpul yang ada di open dengan f minimal, pindahkan bestNode tersebut dari open ke closed.
5. Jika bestNode sama dengan goal maka sukses, tampilkan rute dan berhenti.
6. Jika bestNode tidak sama dengan goal maka bangkitkan semua suksesor bestNode tapi jangan buat pointer.
7. Untuk semua suksesor kerjakan: hitung
 $g(\text{suksesor}) = g(\text{bestnode}) + \text{actual cost}(\text{bestnode ke suksesor})$.
8. Jika suksesor ada di open maka: isi old dengan data simpul di open yang sama dengan suksesor tersebut. Tambahkan old sebagai suksesor bestnode. Buat pointer dari suksesor ke bestnode. Bandingkan nilai $g(\text{old})$ dan $g(\text{suksesor})$. Jika $g(\text{suksesor})$ lebih baik maka ubah parent old ke bestnode, ubah nilai g dan f yang ada pada old.
9. Jika suksesor ada di closed maka: isi old dengan data simpul di closed yang sama dengan suksesor tersebut, tambahkan old sebagai suksesor bestnode, bandingkan nilai $g(\text{old})$ dan $g(\text{suksesor})$. Jika $g(\text{suksesor})$ lebih baik maka ubah parent old ke bestnode, ubah nilai g dan f yang ada pada old, lakukan pengecekan seperti langkah 7,8,9,10 untuk

merubah nilai g dan f semua suksesor old dengan dengan aturan, loop sampai simpul tidak punya suksesor.

10. Jika suksesor tidak ada di open dan di closed maka lanjutkan ke langkah 11.
11. Jika banyak simpul di open lebih kecil dari maksimal maka masukkan suksesor ke open, tambahkan suksesor tersebut sebagai suksesornya bestnode, hitung $f=g(\text{suksesor})+h(\text{suksesor})$.
12. Jika banyak simpul di open sama dengan maksimal maka hitung $f(\text{suksesor})=g(\text{suksesor})+h(\text{suksesor})$, cek nilai f terbesar antara suksesor dan simpul di open.
13. Jika nilai f suksesor merupakan nilai f terbesar maka suksesor dihapus.
14. Jika nilai f suksesor bukan merupakan nilai f terbesar maka simpul di open yang memiliki nilai f terbesar dihapus, dan masukkan suksesor ke open, lalu tambahkan suksesor tersebut sebagai suksesornya bestnode.
15. Kembali ke langkah 3

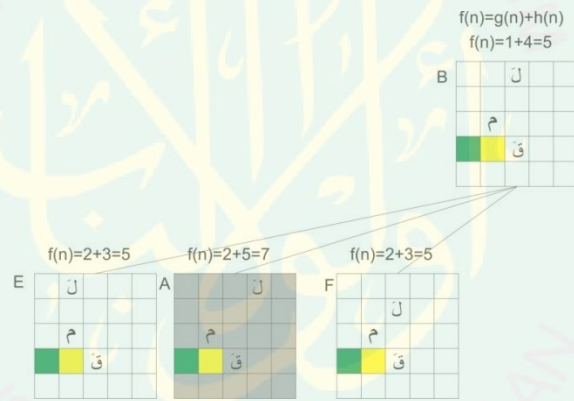
Contoh perhitungan *Beam A**:



Gambar 2.2 Langkah 1 pencarian arah gerak huruf ل

Langkah 1:

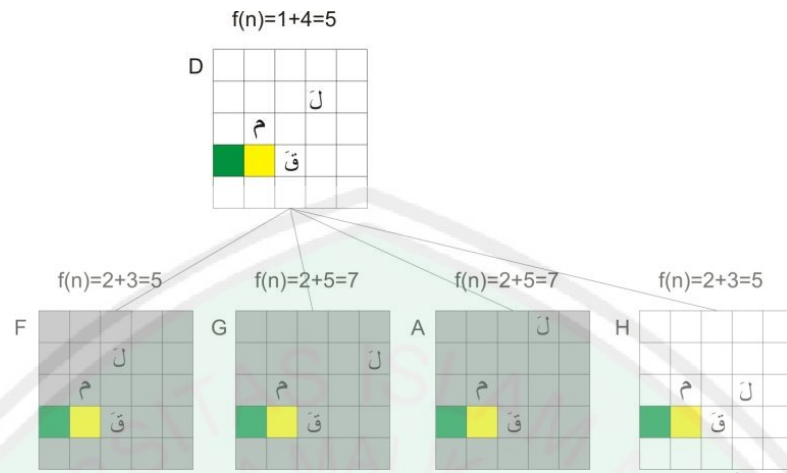
Closed=A, Open=BCD



Gambar 2.3 Langkah 2 pencarian arah gerak huruf ل

Langkah 2:

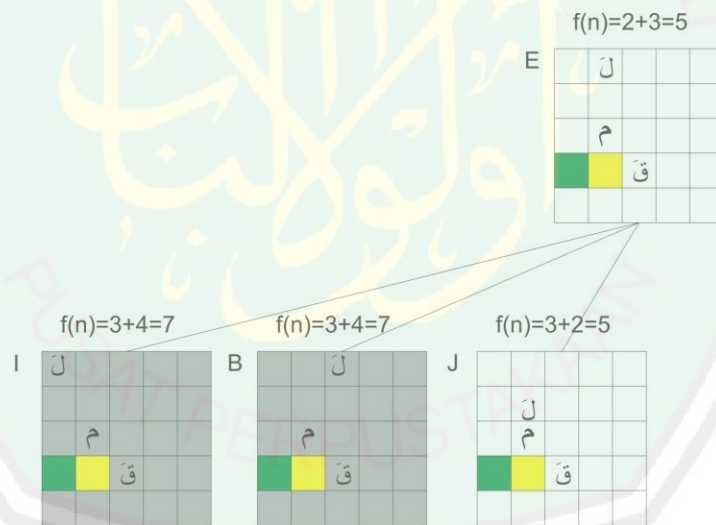
Closed=AB, Open=DEF



Gambar 2.4 Langkah 3 pencarian arah gerak huruf ل

Langkah 3:

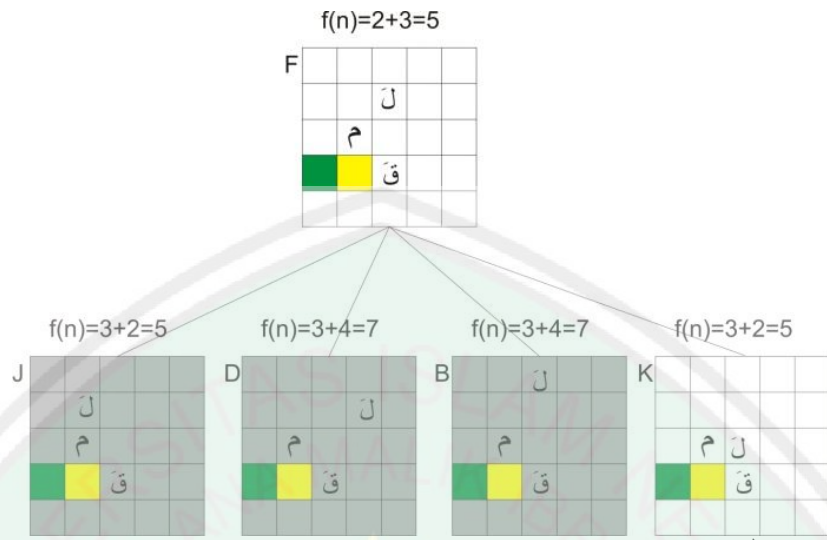
Closed=ABD, Open=EFH



Gambar 2.5 Langkah 4 pencarian arah gerak huruf ل

Langkah 4:

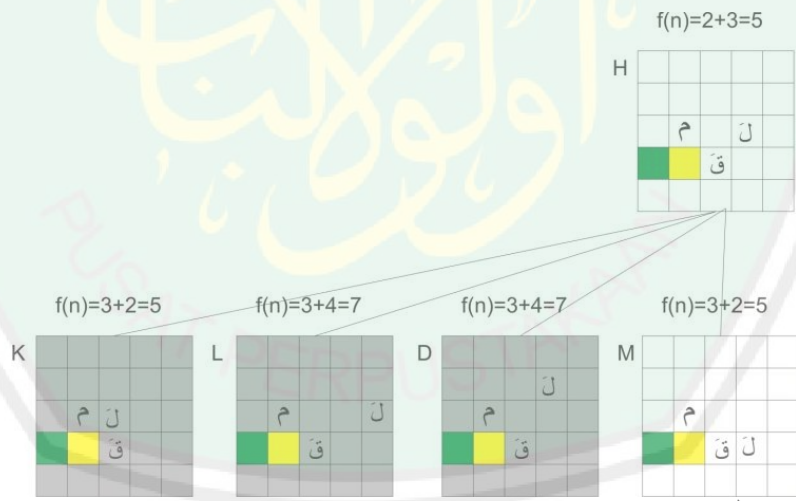
Closed=ABDE, Open=FHJ



Gambar 2.6 Langkah 5 pencarian arah gerak huruf ل

Langkah 5:

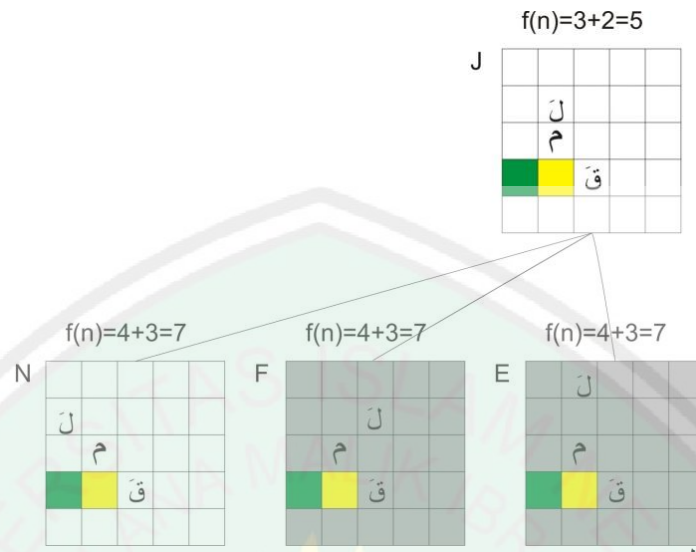
Closed=ABDEF, Open=HJK



Gambar 2.7 Langkah 6 pencarian arah gerak huruf ل

Langkah 6:

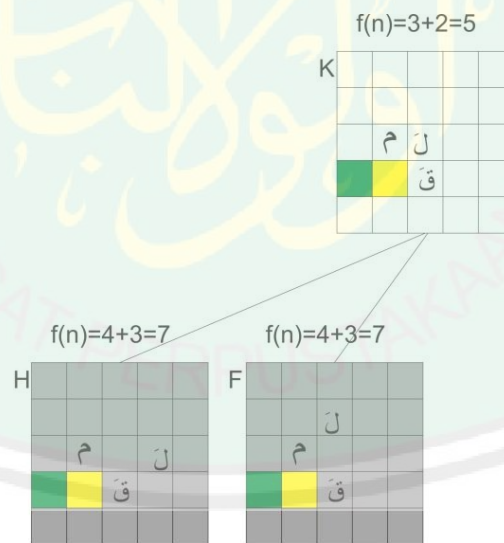
Closed=ABDEFH, Open=JKM



Gambar 2.8 Langkah 7 pencarian arah gerak huruf ل

Langkah 7:

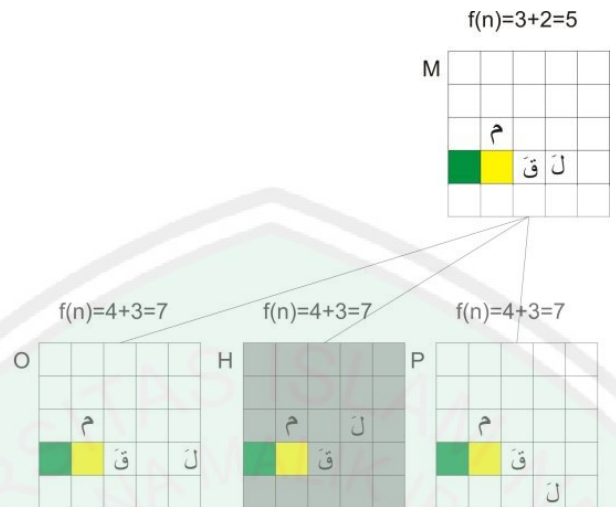
Closed=ABDEFHJ, Open=KMN



Gambar 2.9 Langkah 8 pencarian arah gerak huruf ل

Langkah 8:

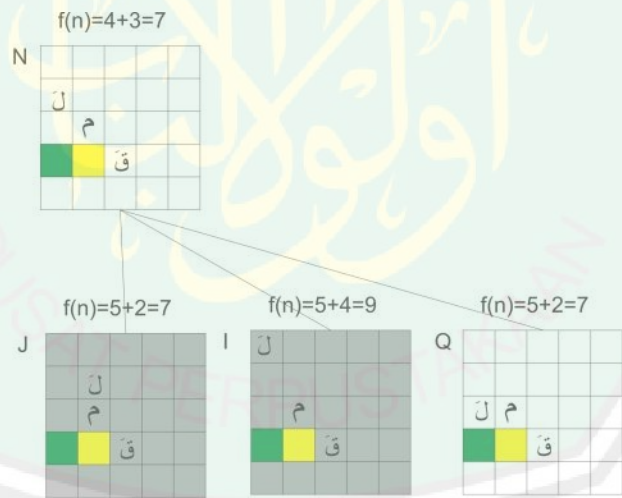
Closed=ABDEFHJK, Open=MN



Gambar 2.10 Langkah 9 pencarian arah gerak huruf ل

Langkah 9:

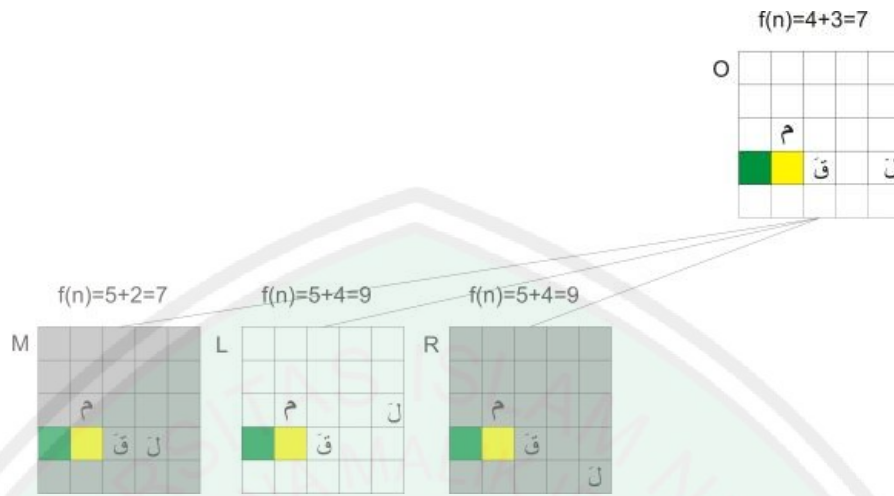
Closed=ABDEFHJKM, Open=NOP



Gambar 2.11 Langkah 10 pencarian arah gerak huruf ل

Langkah 10:

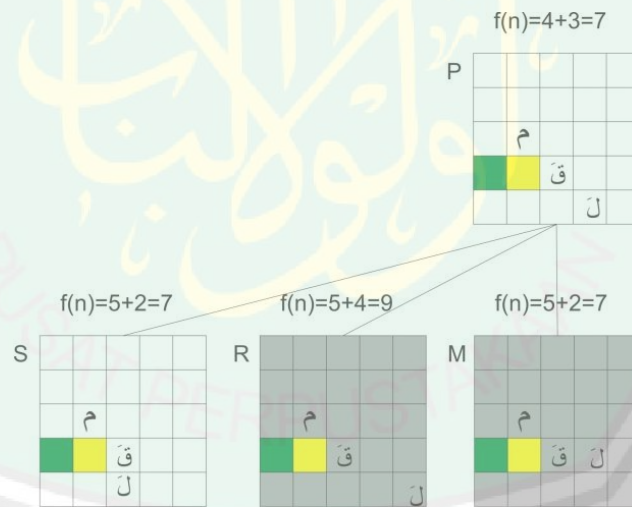
Closed=ABDEFHJKMN, Open=OPQ



Gambar 2.12 Langkah 11 pencarian arah gerak huruf ل

Langkah 11:

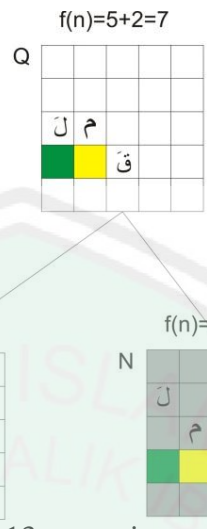
Closed=ABDEFHJKMNO, Open=PQL



Gambar 2.13 Langkah 12 pencarian arah gerak huruf ل

Langkah 12:

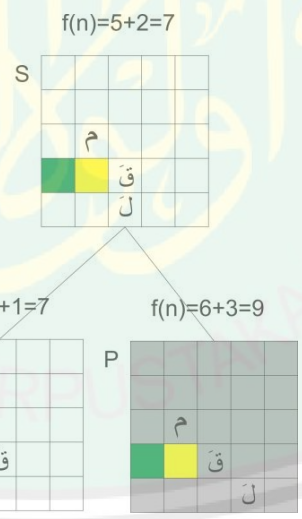
Closed=ABDEFHJKMNOP, Open=QSL



Gambar 2.14 Langkah 13 pencarian arah gerak huruf ل

Langkah 13:

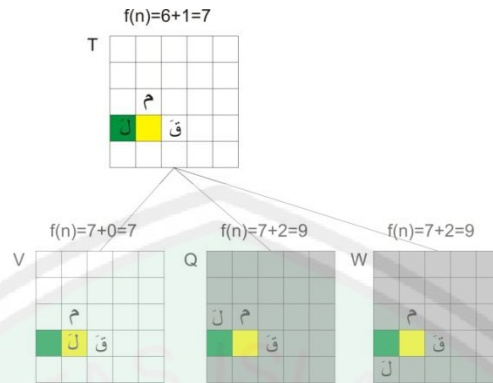
Closed=ABDEFHJKMNOPQ, Open=STL



Gambar 2.15 Langkah 14 pencarian arah gerak huruf ل

Langkah 14:

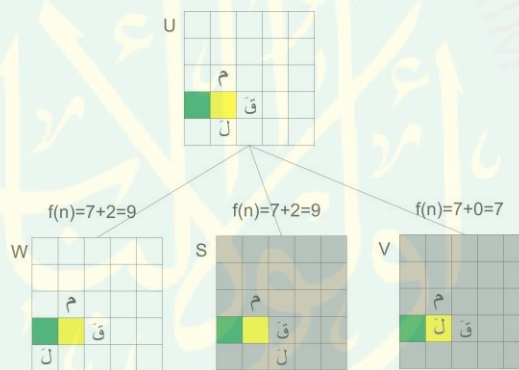
Closed=ABDEFHJKMNOPQS, Open=TUL



Gambar 2.16 Langkah 15 pencarian arah gerak huruf ل

Langkah 15:

Closed=ABDEFHJKLMNQPST, Open=UVL



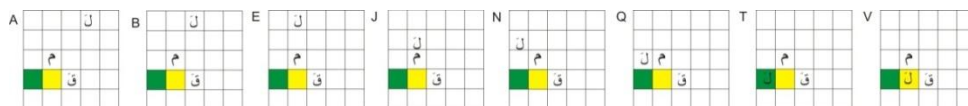
Gambar 2.17 Langkah 16 pencarian arah gerak huruf ل

Langkah 16:

Closed=ABDEFHJKLMNQPSTU, Open=VLW

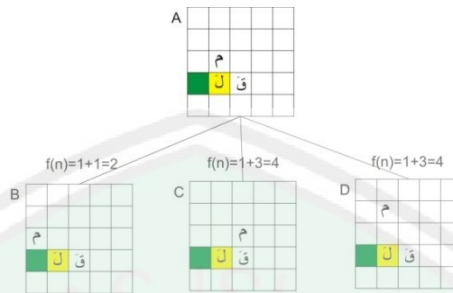
Kemudian V menjadi bestnode dari Open. Karena V merupakan simpul tujuan maka pencarian berhenti dan dihasilkan rute sebagai berikut:

A-B-E-J-N-Q-T-V



Gambar 2.18 Rute arah gerak huruf ل

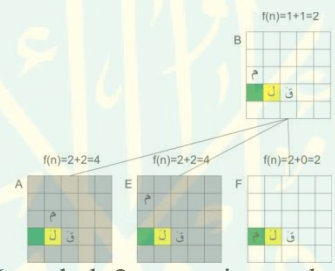
Untuk huruf mim:



Gambar 2.19 Langkah 1 pencarian arah gerak huruf ”م”

Langkah 1:

Close= A, Open= BCD



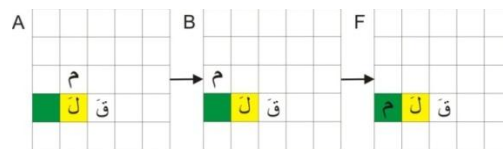
Gambar 2.20 Langkah 2 pencarian arah gerak huruf ”م”

Langkah 2:

Close= AB, Open= FCD

Kemudian F menjadi bestnode dari Open. Karena F merupakan simpul tujuan maka pencarian berhenti dan dihasilkan rute sebagai berikut:

A-B-F



Gambar 2.21 Rute arah gerak huruf ”م”



BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Adapun tahap-tahap yang dilakukan dalam penelitian ini, yaitu:

1. Studi Pustaka

Pada tahap ini dilakukan berbagai pengumpulan informasi terkait dengan penelitian yang akan dibangun sebagai berikut :

- a. Pengumpulan informasi tentang bagaimana cara pembuatan *game puzzle*
- b. Pengumpulan informasi kosa kata bahasa Arab beserta artinya dalam bahasa Indonesia sebagai materi dalam permainan
- c. Pengumpulan informasi tentang algoritma *Beam A** untuk menentukan arah gerak obyek pada *puzzle*.

2. Perancangan dan desain aplikasi

Perancangan algoritma *Beam A** untuk menentukan arah gerak obyek pada *puzzle*. Dan desain aplikasi adalah pembuatan desain menu *game*, karakter-karakter yang terdapat dalam *game*, dan obyek lain yang ada dalam *game*.

3. Pembuatan aplikasi

Mengimplementasikan rancangan dan desain aplikasi dalam bahasa pemrograman Java berbasis android

4. Uji coba dan evaluasi

Tahap pengujian dan evaluasi dilakukan setelah aplikasi selesai dibuat.

5. Penyusunan laporan

Penyusunan laporan terdiri atas dokumentasi dari keseluruhan proses yang telah dilakukan



Gambar 3.1 Tahap-tahap penelitian

3.2 Desain Sistem

3.2.1 Deskripsi Aplikasi

Game ini tergolong dalam genre *edutainment*. *Game* ini bersifat *single player* dan mengharuskan pemain untuk mengatur beberapa obyek sehingga tersusun menjadi sebuah kata bahasa Arab. *Game* ini menggunakan teknologi *mobile* dengan sistem operasi Android sebagai sarana memainkannya dikarenakan semakin banyaknya pengguna teknologi tersebut.

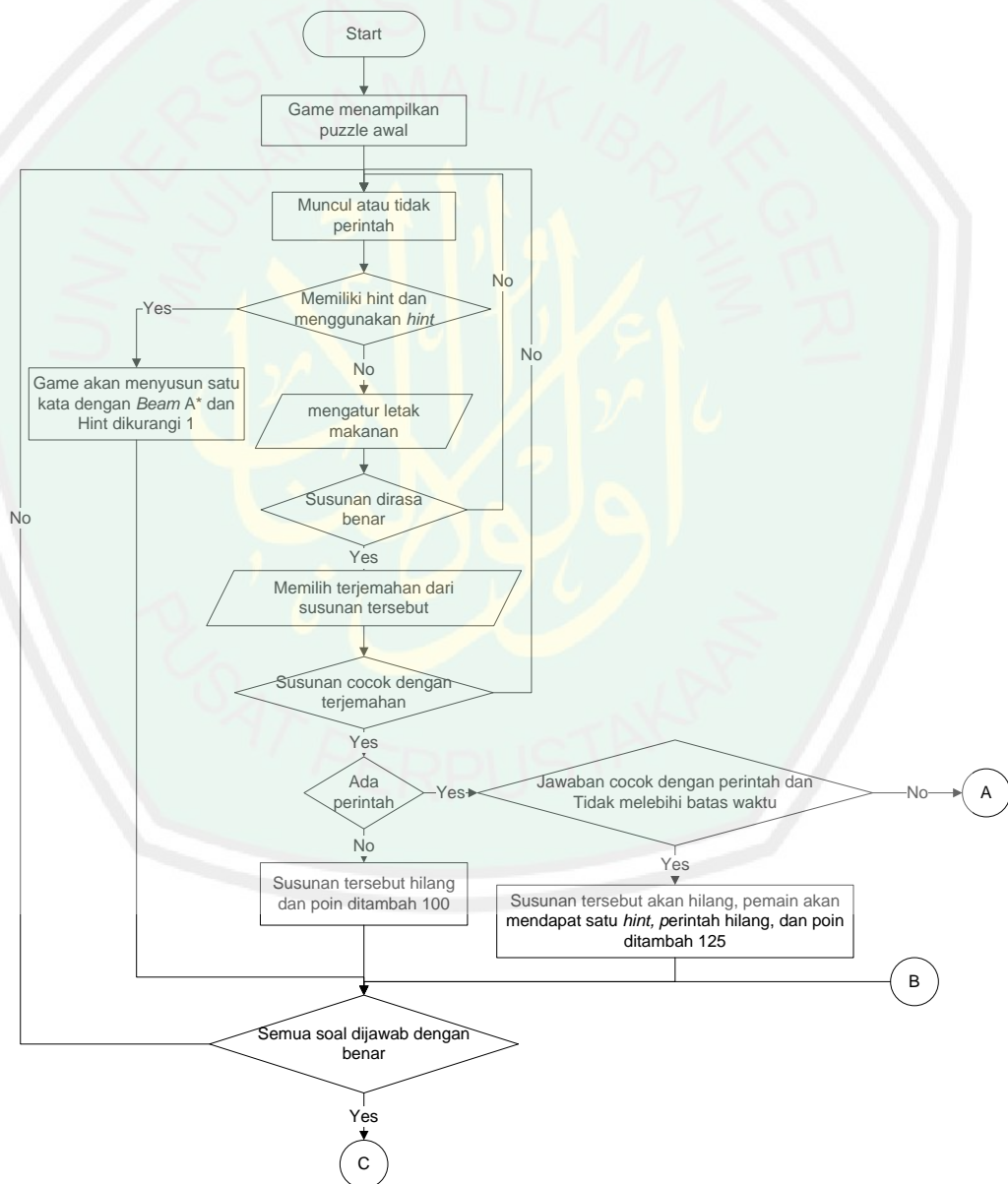
3.2.2 Storyline

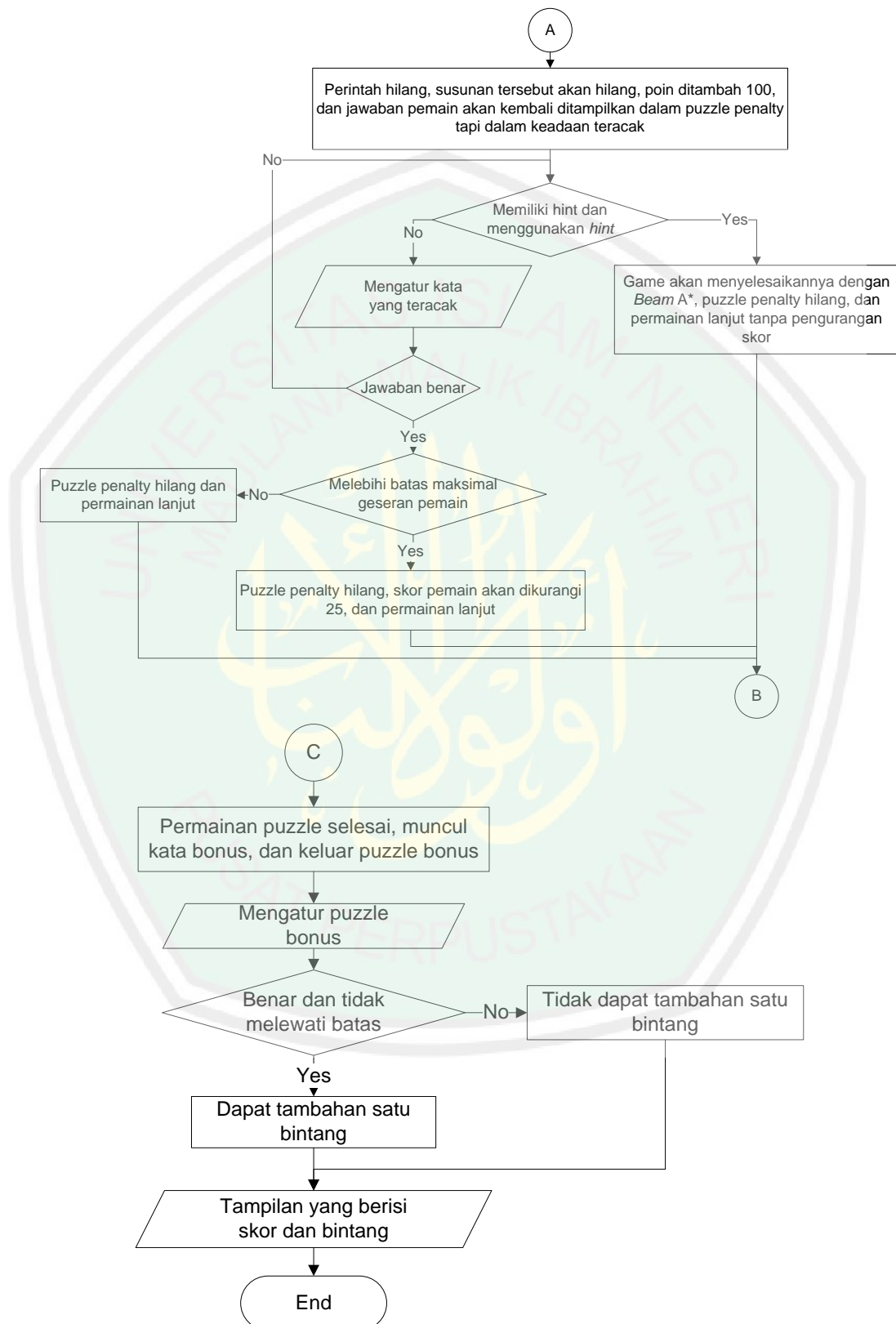
Game ini menceritakan sekelompok semut yang berjuang untuk mengumpulkan makanan, dan mereka bertugas untuk mengumpulkan semua makanan biasa dan makanan bonus. *Game* ini memiliki level. Dan setiap level memiliki makanan bonus yang berbeda-beda.

3.2.3 Gameplay

Dalam *game* terdapat *puzzle* makanan yang di setiap makanan mengandung huruf hijaiyah dan pemain berusaha untuk mengatur letak makanan yang terdapat dalam *puzzle* tersebut agar dapat tersusun menjadi sebuah kata bahasa Arab, setelah makanan-makanan tersebut tersusun pemain harus memilih kata terjemahan bahasa Indonesia dari kata bahasa Arab yang dimaksud. Kemudian sistem akan mencocokkan hasil susunan pemain dengan kata bahasa Arab yang dipilih. Jika susunan tersebut benar maka susunan makanan tersebut akan hilang dari *puzzle* dan akan dikumpulkan oleh semut. Sedangkan untuk

mendapatkan makanan bonus pemain harus menyelesaikan *puzzle* tersebut terlebih dahulu. Setelah semua makanan hilang dari *puzzle* yang menandakan pemain telah menjawab semua soal dengan benar maka kata bonus akan keluar dan pemain bertugas untuk mengatur kata bonus yang teracak. Setelah pemain mengatur kata bonus tersebut maka pemain akan masuk ke level selanjutnya.





Gambar 3.2 Flowchart untuk Game puzzle bahasa Arab

3.2.4 Cara mengatur posisi buah

Pemain bisa mengatur posisi buah dengan cara menggeser gambar buah yang terdapat dalam *puzzle*, baik dalam *puzzle* game inti, maupun *puzzle* penalti dan bonus.

3.2.4.1 *Puzzle* inti

- Pemain bisa menggeser buah ke atas, bawah, kanan, dan kiri.
- Buah tidak bisa digeser ke luar dari *puzzle*.
- Buah tidak bisa digeser ke atas jika di atasnya tidak ada buah.
- Jika buah digeser ke kanan atau ke kiri dan di bawahnya terdapat area kosong maka buah tersebut akan jatuh.

3.2.4.2 *Puzzle* penalti dan *puzzle* bonus

- Pemain bisa menggeser ke kanan dan ke kiri tanpa keluar dari *puzzle*.
- Jumlah geseran pemain tidak boleh melebihi batas geseran yang ditampilkan, jika melebihi batas dan pemain berada di *puzzle* penalti maka poin pemain akan dikurangi 25 tapi jika pemain ada di *puzzle* bonus maka pemain tidak akan mendapat tambahan 1 bintang.

3.2.5 Scoring

- Pemain akan mendapatkan poin 100 jika berhasil menjawab dengan benar.
- Pemain akan mendapatkan poin tambahan 25 jika berhasil menjawab sesuai perintah dan tidak melebihi batas waktu.

- Poin pemain akan dikurangi 25 jika geseran pemain pada *puzzle* penalti melebihi batas geseran.
- Di akhir level pemain akan mendapatkan satu bintang jika hanya berhasil mendapatkan poin kurang dari 50 % dari poin maksimal.
- Di akhir level pemain akan mendapatkan dua bintang jika berhasil mendapatkan poin lebih dari 50 % dari poin maksimal.
- Pemain akan mendapatkan tambahan satu bintang jika berhasil menyelesaikan *puzzle* bonus tanpa melebihi batas geseran di akhir level.

3.2.6 Konten-konten pada aplikasi

Konten-konten aplikasi yang ada dalam *game* adalah :

1. Materi kata bahasa Arab

Tabel 3.1 Tabel materi kata bahasa Arab

No.	Bab I		Bab II		Bab III	
	Indonesia	Arab	Indonesia	Arab	Indonesia	Arab
1	Pena	قَلَم	Gambar	صُورَة	Rumah	بَيْت
2	Kitab	كِتَاب	Keluarga	أُسْرَة	Apartemen	شَقَة
3	Buku	دَفْتَر	Ayah	وَالِد	Kamar-kamar	عُرْف
4	Pensil	مِرْسَم	Ibu	وَالِدَة	Kamar tidur	عُرْفَة نَوْم
5	Buku tulis	كُرْسَة	Bapak ibu	الْأَبْوَاب	Ruang tamu	عُرْفَة جُلُوس

No.	Bab I		Bab II		Bab III	
	Indonesia	Arab	Indonesia	Arab	Indonesia	Arab
6	Kertas	قِرْطَاس	Nenek	جَدَّة	Dapur	مَطْبَخ
7	Kapur	طَبَاسِير	Kakek	جَدَّ	Tempat tidur	سَرِير
8	Papan tulis	سَبُورَة	Anak laki-laki	إِبْن	Sofa	الأَرِيكَة
9	Penghapus	مُمْسَحَة	Anak perempuan	إِبْنَة	Sajadah	سَجَادَة
10	Guru	مُدْرَس	Paman	عَمَّ	Oven	فَرْن
11	Dokter	طَبِيب	Tante	عَمَّة	Almari	خِرَانَة
12	Insinyur	مُهَنْدِس	Anak-anak	أَوْلَاد	Kulkas	ثَلَاجَة
13	Saudara	أَخ	Pohon	شَجَر	Cermin	مِرَاة
14	Saudara (perempuan)	أَخْت	Masjid	مَسْجِد	Tirai	سِتَارَة
15	Teman	صَدِيق	Musholla	مُصَلَّى	Pemanas	سَخَان
16	Saya	أَنَا	Ruangan	عُرْفَة	Lingkungan	حَيَّ
17	Kamu	أَنْت	Al-qur'an	الْقُرْآن	Nomer	رَقْم
18	Kamu (perempuan)	أَنْتِ	Mantel	مِعْطَف	Furniture	أَثَات
19	Ini	هَذَا	Kaca mata	نَظَّارَة	Ruangan	عُرْفَة
20	Ini	هَذِهِ	Toilet	حَمَّام		

No.	Bab I		Bab II		Bab III	
	Indonesia	Arab	Indonesia	Arab	Indonesia	Arab
	(perempuan)					
21	Dia	هُوَ	Berwudlu	يَتَوَضَّأُ		
22	Dia (perempuan)	هِيَ	Membaca	يَقْرَأُ		
23	Murid	تَلْمِيذٌ	Sholat	يُصَلِّي		
24			Enam	سِتَّةٌ		
25			Tujuh	سَبْعَةٌ		
26			Delapan	ثَمَانِيَةٌ		
27			Sembilan	تِسْعَةٌ		
28			Sepuluh	عَشْرَةٌ		

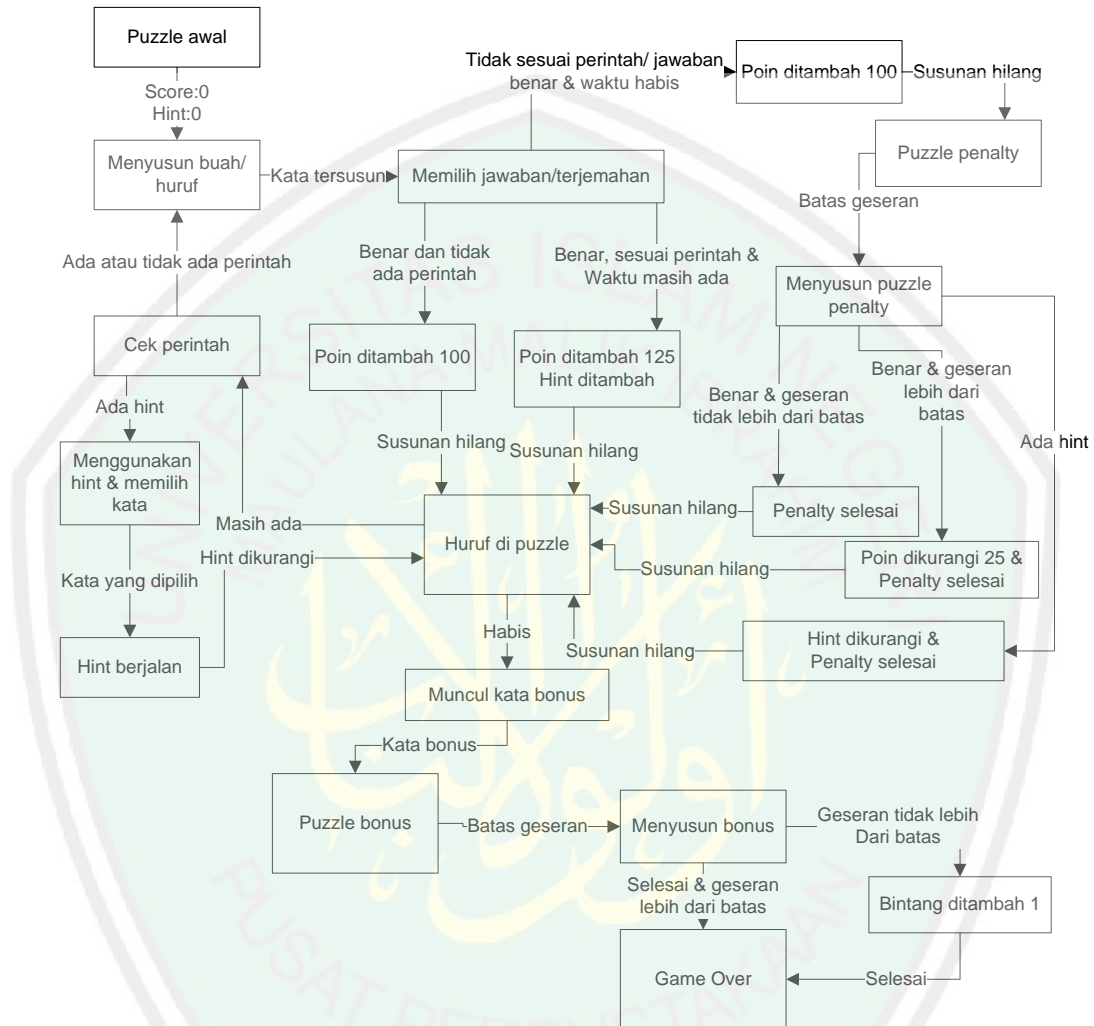
2. Latar

1. Level pertama : pohon apel
2. Level kedua : pohon stroberi
3. Level ketiga : pohon anggur

3. Tokoh utama (pemain)

Sekelompok semut yang berjuang mengumpulkan makanan.

3.2.7 Skenario Game



Gambar 3.3 *Finite state machine* permainan

1. Pemain bertugas untuk menyusun semua makanan dengan benar sehingga semua makanan dapat dikumpulkan oleh semut.
2. Pada saat-saat tertentu akan muncul perintah kepada pemain untuk mengatur obyek menjadi kata bahasa Arab yang dimulai dengan huruf yang ditampilkan oleh sistem.

3. Pemain bisa menggunakan *hint* atau bantuan untuk menyelesaikan soal selama *hint* masih ada.
4. Pemain mengatur posisi makanan baik horizontal dari kanan ke kiri maupun vertical dari atas ke bawah sesuai keinginan pemain, dan susunan tersebut diatur supaya menjadi susunan kata bahasa Arab. Setelah susunan makanan tersebut dianggap benar, kemudian pemain memilih kata bahasa Indonesia yang tersedia yang dianggap sebagai terjemahan dari susunan tersebut.
5. Jika susunan benar maka susunan tersebut akan hilang dan dikumpulkan oleh semut. Jika kata tersusun sebelum selang waktu habis, maka satu kotak kata bonus akan terbuka dan pemain akan mendapat satu *hint*. Dan jika pemain melewati selang waktu atau mengatur obyek kata lain maka jawaban pemain akan kembali ditampilkan tapi dalam keadaan teracak, dan pemain diwajibkan mengatur obyek tersebut dengan tidak melebihi batas maksimal geseran pemain. Jika pemain berhasil mengatur obyek tersebut tanpa melewati batas maka permainan lanjut tanpa pengurangan poin dan satu kotak kata bonus terbuka, tapi jika pemain melewati batas maka skor pemain akan dikurangi 25 tapi satu kotak kata bonus akan terbuka dan permainan lanjut, dan jika pemain tidak berhasil mengatur obyek tersebut, pemain bisa menggunakan satu *hint* untuk mengatur obyek tersebut dan satu kotak kata bonus akan terbuka tanpa pengurangan skor.

6. Setelah semua kata berhasil disusun dengan benar, maka akan ditampilkan kata bonus beserta artinya. Kemudian kata tersebut diacak oleh sistem dan pemain diharuskan mengatur acakan tersebut. Jika aturan benar maka pemain akan mendapat tambahan satu bintang, tapi jika salah dan menyerah maka acakan tersebut akan diselesaikan oleh sistem tapi pemain tidak mendapat tambahan bintang.

3.2.8 Storyboard

1. Misi



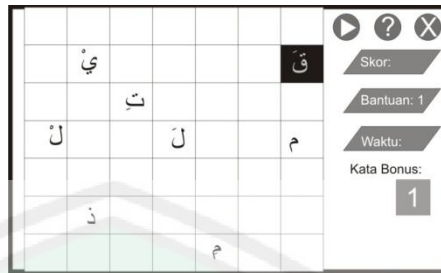
Gambar 3.4 Tampilan misi

2. Puzzle awal



Gambar 3.5 Tampilan *puzzle* awal

3. Pada saat-saat tertentu sistem akan menampilkan perintah untuk menyusun kata yang diawali dengan huruf yang ditampilkan oleh sistem.



Gambar 3.6 Tampilan *game* menampilkan perintah

4. Pemain mengatur susunan puzzle supaya menjadi satu kata bahasa Arab, baik saat ada perintah dari sistem maupun tidak ada.



Gambar 3.7 Tampilan *puzzle* tersusun

5. Jika jawaban dirasa benar pemain kemudian membuka terjemahan bahasa Indonesia untuk memilih terjemahan dari susunan puzzle yang berhasil disusun.



Gambar 3.8 Tampilan terjemahan bahasa Indonesia

6. Jika jawaban benar maka susunan tersebut akan diambil oleh semut dan hilang. Jika jawaban tersebut cocok dengan perintah dan tidak melebihi

batas waktu maka pemain akan mendapatkan tambahan bantuan dan satu kotak kata bonus akan terbuka.



Gambar 3.9 Tampilan semut mengumpulkan *puzzle* yang benar

7. Jika pemain berhasil menjawab dengan benar tapi melebihi batas waktu atau menjawab dengan kata lain, maka akan muncul tampilan seperti di bawah ini dan pemain bertugas untuk mengatur obyek tersebut supaya menjadi satu kata tanpa melebihi batas maksimal gerakan pemain.



Gambar 3.10 Tampilan *puzzle* tambahan

8. Setelah semua soal berhasil terjawab dengan benar, maka permainan selesai dan akan muncul tampilan yang berisi kata bonus yang berhasil didapatkan.



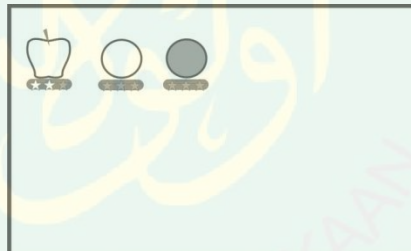
Gambar 3.11 Tampilan kata bonus

9. Pemain bertugas menyusun puzzle menjadi kata bonus yang telah ditampilkan tadi tanpa melewati batas maksimum gerakan pemain. Jika pemain berhasil pemain akan mendapatkan tambahan satu bintang, tapi jika tidak, maka pemain tidak mendapatkan tambahan satu bintang.



Gambar 3.12 Tampilan kata bonus yang diacak

10. Jika game telah selesai maka pemain akan bisa melihat koleksi level yang telah dilewati



Gambar 3.13 Tampilan koleksi level

3.2.9 Penerapan algoritma *Beam A**

1. Algoritma Beam A* digunakan pada proses bantuan atau hint untuk menentukan rute arah gerak potongan kata saat pemain memilih untuk menggunakan fitur tersebut.
2. Algoritma Beam A* juga digunakan untuk mengacak puzzle kata bahasa Arab pada saat tertentu. Yakni pada saat pemain menjawab tapi

melebihi batas waktu dan saat pemain telah menyelesaikan puzzle dan akan ditampilkan puzzle makanan bonus.

3.3 Kebutuhan Sistem

Pada bagian ini akan ditulis kebutuhan perangkat keras(hardware) dan perangkat lunak(software) yang akan digunakan dalam proses pembuatan hingga pengujian.

A. Kebutuhan Perangkat Keras (Hardware)

Laptop dengan processot Core i3-2330M, RAM 4 GB, Hardisk 320 GB, LCD resolusi 1366 x 768, Keyboard, Mouse, Mobile phone Android minimum android versi 3.0.

B. Kebutuhan Perangkat Lunak (*Software*)

1. JDK, *Java Development Kit*
2. Eclipse dan ADT, *Android Development Tools*
3. Inkscape yang merupakan *software* untuk membuat desain grafis dalam *game*.

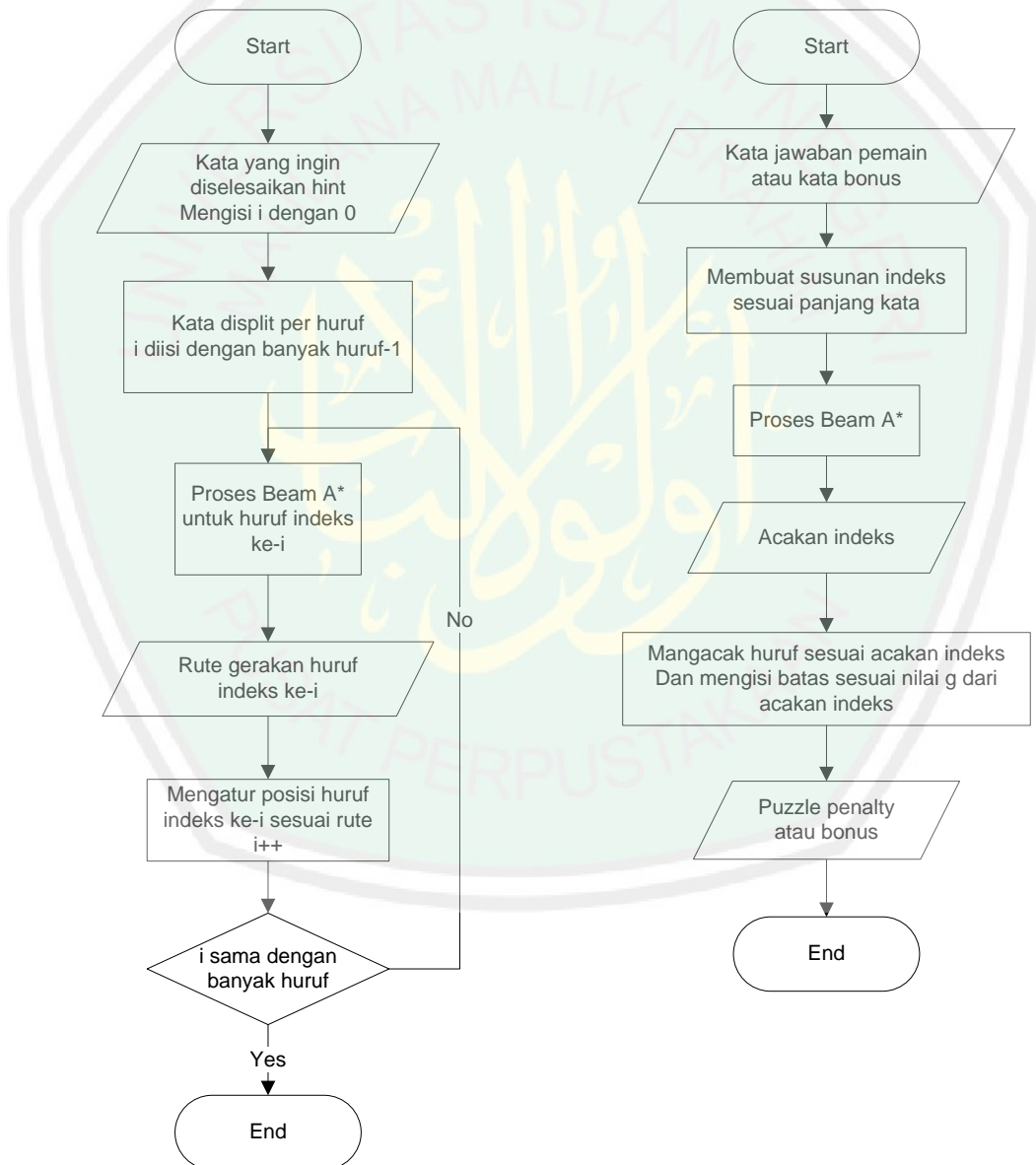


BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Algoritma *Beam A** (BA^*)

Berikut ini flowchart implementasi algoritma *Beam A** dalam game.



Gambar 4.1 flowchart implementasi *Beam A**(menyusun kata dan mengacak kata)

4.2 Pengujian Algoritma *Beam A**

Berikut akan dijelaskan tentang hasil pengujian dari algoritma *Beam A** untuk kata خزانة.

Tabel 4.1 Tabel hasil pengujian algoritma *beam a** huruf خ

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
1	1	[3, 4]	[6, 3] g:1 f:4 [3, 5] g:1 f:5 [4, 4] g:1 f:5	[2, 4]	[3, 4] g:0 f:5	
2	2	[6, 3]	[3, 5] g:1 f:5 [4, 4] g:1 f:5 [6, 4] g:2 f:4	[6, 2]	[3, 4] g:0 f:5 [6, 3] g:1 f:4	
3	3	[6, 4]	[3, 5] g:1 f:5 [4, 4] g:1 f:5 [6, 5] g:3 f:4	[6, 3] [5, 4]	[3, 4] g:0 f:5 [6, 3] g:1 f:4 [6, 4] g:2 f:4	
4	4	[6, 5]	[3, 5] g:1 f:5 [4, 4] g:1 f:5 [6, 6] g:4 f:4	[6, 4] [5, 5]	[3, 4] g:0 f:5 [6, 3] g:1 f:4 [6, 4] g:2 f:4 [6, 5] g:3 f:4	
5	5	[6, 6]	[3, 5] g:1 f:5 [4, 4] g:1 f:5		[3, 4] g:0 f:5 [6, 3] g:1 f:4 [6, 4] g:2 f:4 [6, 5] g:3 f:4 [6, 6] g:4 f:4	Ketemu [6, 6] [6, 5] [6, 4] [6, 3] [3, 4]

Hasil algoritma:

ن	ذ	ن			ف	ن
ا	ة	م			ا	ر
ة	ن	ن			ا	ر
م	ف	ن			ا	ر
م	ي	ف			ا	ر
ن	ب	ا			ا	ر

Gambar 4.2 Matriks awal

ن	ذ	ن			ف	ن
ا	ة	م			ا	ر
ة	ن	ن			ا	ر
م	ف	ن			ا	ر
م	ي	ف			ا	ر
ن	ب	ا			ا	ر

ن	ذ	ن			ف	ن
ا	ة	م			ا	ر
ة	ن	ن			ا	ر
م	ف	ن			ا	ر
م	ي	ف			ا	ر
ن	ب	ا			ا	ر

ن	ذ	ن			ف	ن
ا	ة	م			ا	ر
ة	ن	ن			ا	ر
م	ف	ن			ا	ر
م	ي	ف			ا	ر
ن	ب	ا			ا	ر

Gambar 4.3 Langkah 1, 2, dan 3

ن	ذ	ن			ف	ن
ا	ة	م			ا	ر
ة	ن	ن			ا	ر
م	ف	ن			ا	ر
م	ي	ف			ا	ر
ن	ب	ا			ا	ر

Gambar 4.4 Langkah 4 (selesai)

Tabel 4.2 Tabel hasil pengujian algoritma *beam a** huruf ن

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
1	1	[2, 1]	[2, 0] g:1 f:10 [2, 2] g:1 f:8 [3, 1] g:1 f:8	[1, 1]	[2, 1] g:0 f:8	
2	2	[2, 2]	[3, 1] g:1 f:8 [5, 3] g:2 f:5 [3, 2] g:2 f:8	[2, 1] [1, 2] [2, 0]	[2, 1] g:0 f:8 [2, 2] g:1 f:8	

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
3	3	[5, 3]	[3, 1] g:1 f:8 [5, 4] g:3 f:5 [6, 3] g:3 f:5	[5, 2] [3, 2]	[2, 1] g:0 f:8 [2, 2] g:1 f:8 [5, 3] g:2 f:5	
4	4	[5, 4]	[6, 3] g:3 f:5 [5, 5] g:4 f:5 [6, 4] g:4 f:5	[5, 3] [3, 1] [4, 4]	[2, 1] g:0 f:8 [2, 2] g:1 f:8 [5, 3] g:2 f:5 [5, 4] g:3 f:5	
5	5	[6, 3]	[5, 5] g:4 f:5 [6, 4] g:4 f:5 [6, 2] g:4 f:7	[6, 4]	[2, 1] g:0 f:8 [2, 2] g:1 f:8 [5, 3] g:2 f:5 [5, 4] g:3 f:5 [6, 3] g:3 f:5	
6	6	[5, 5]	[6, 4] g:4 f:5 [6, 2] g:4 f:7 [6, 5] g:5 f:5	[5, 4] [4, 5] [5, 6]	[2, 1] g:0 f:8 [2, 2] g:1 f:8 [5, 3] g:2 f:5 [5, 4] g:3 f:5 [6, 3] g:3 f:5 [5, 5] g:4 f:5	
7	7	[6, 4]	[6, 2] g:4 f:7 [6, 5] g:5 f:5	[6, 3] [6, 5] [5, 4]	[2, 1] g:0 f:8 [2, 2] g:1 f:8 [5, 3] g:2 f:5 [5, 4] g:3 f:5 [6, 3] g:3 f:5 [5, 5] g:4 f:5 [6, 4] g:4 f:5	
8	8	[6, 5]	[6, 2] g:4 f:7		[2, 1] g:0 f:8 [2, 2] g:1 f:8 [5, 3] g:2 f:5 [5, 4] g:3 f:5	Ketemu [6, 5] [5, 5] [5, 4]

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
					[6, 3] g:3 f:5	[5, 3]
					[5, 5] g:4 f:5	[2, 2]
					[6, 4] g:4 f:5	[2, 1]
					[6, 5] g:5 f:5	

Hasil algoritma:

ن	ذ	ن			ق	ن
ا	ة	م			ر	ة
ة	ن	ن		ا	ن	م
م	ف	ن		ن	ا	م
م	ي	ف		ا	ن	ا
ن	ف	ي		ن	م	ن
ن	ب	ة	م	ن	م	م

Gambar 4.5 Matriks awal

ن	ذ	ن			ق	ن
ا	ة	م			ر	ة
ة	ن	ن		ا	ن	م
م	ف	ن		ن	ا	م
م	ي	ف		ا	ن	ا
ن	ف	ي		ن	م	ن
ن	ب	ة	م	ن	م	م

Gambar 4.6 Langkah 1,2, dan 3

ن	ذ				ق	ن
ا	ة	ن			ر	ة
ة	ن	م		ا	م	م
م	ف	ن		ن	ا	م
م	ي	ف		ا	ن	ا
ن	ف	ي		ن	م	ن
ن	ب	ة	م	ن	م	م

Gambar 4.7 Langkah 4 dan 5 (selesai)

Tabel 4.3 Tabel hasil pengujian algoritma *beam a** huruf l

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
1	1	[4, 6]	[4, 5] g:1 f:4 [3, 6] g:1 f:6 [5, 6] g:1 f:4		[4, 6] g:0 f:4	
2	2	[4, 5]	[5, 6] g:1 f:4 [4, 4] g:2 f:4 [5, 5] g:2 f:4	[4, 6] [3, 5] [3, 6]	[4, 6] g:0 f:4 [4, 5] g:1 f:4	
3	3	[5, 6]	[4, 4] g:2 f:4 [5, 5] g:2 f:4	[5, 5] [4, 6]	[4, 6] g:0 f:4 [4, 5] g:1 f:4 [5, 6] g:1 f:4	
4	4	[4, 4]	[5, 5] g:2 f:4 [4, 3] g:3 f:6 [5, 4] g:3 f:4	[4, 5] [3, 4]	[4, 6] g:0 f:4 [4, 5] g:1 f:4 [5, 6] g:1 f:4 [4, 4] g:2 f:4	
5	5	[5, 5]	[4, 3] g:3 f:6 [5, 4] g:3 f:4	[5, 4] [5, 6] [4, 5]	[4, 6] g:0 f:4 [4, 5] g:1 f:4 [5, 6] g:1 f:4 [4, 4] g:2 f:4 [5, 5] g:2 f:4	
6	6	[5, 4]	[4, 3] g:3 f:6 [5, 3] g:4 f:6 [6, 4] g:4 f:4	[5, 5] [4, 4]	[4, 6] g:0 f:4 [4, 5] g:1 f:4 [5, 6] g:1 f:4 [4, 4] g:2 f:4 [5, 5] g:2 f:4 [5, 4] g:3 f:4	
7	7	[6, 4]	[4, 3] g:3 f:6 [5, 3] g:4 f:6		[4, 6] g:0 f:4 [4, 5] g:1 f:4 [5, 6] g:1 f:4 [4, 4] g:2 f:4	Ketemu [6, 4] [5, 4] [4, 4]

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
					[5, 5] g:2 f:4	[4, 5]
					[5, 4] g:3 f:4	[4, 6]
					[6, 4] g:4 f:4	

Hasil algoritma:

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	8
6	5	4	3	2	1	8	7
5	4	3	2	1	8	7	6
4	3	2	1	8	7	6	5
3	2	1	8	7	6	5	4
2	1	8	7	6	5	4	3
1	8	7	6	5	4	3	2

Gambar 4.8 Matriks awal

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	8
6	5	4	3	2	1	8	7
5	4	3	2	1	8	7	6
4	3	2	1	8	7	6	5
3	2	1	8	7	6	5	4
2	1	8	7	6	5	4	3
1	8	7	6	5	4	3	2

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	8
6	5	4	3	2	1	8	7
5	4	3	2	1	8	7	6
4	3	2	1	8	7	6	5
3	2	1	8	7	6	5	4
2	1	8	7	6	5	4	3
1	8	7	6	5	4	3	2

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	8
6	5	4	3	2	1	8	7
5	4	3	2	1	8	7	6
4	3	2	1	8	7	6	5
3	2	1	8	7	6	5	4
2	1	8	7	6	5	4	3
1	8	7	6	5	4	3	2

Gambar 4.9 Langkah 1, 2, dan 3

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	8
6	5	4	3	2	1	8	7
5	4	3	2	1	8	7	6
4	3	2	1	8	7	6	5
3	2	1	8	7	6	5	4
2	1	8	7	6	5	4	3
1	8	7	6	5	4	3	2

Gambar 4.10 Langkah 4 (selesai)

Tabel 4.4 Tabel hasil pengujian algoritma *beam a** huruf \hat{U}

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
1	1	[4, 5]	[4, 4] g:1 f:4 [4, 6] g:1 f:6 [5, 5] g:1 f:4	[3, 5]	[4, 5] g:0 f:4	
2	2	[4, 4]	[5, 5] g:1 f:4 [4, 3] g:2 f:4 [5, 4] g:2 f:4	[4, 5] [3, 4] [4, 6]	[4, 5] g:0 f:4 [4, 4] g:1 f:4	
3	3	[5, 5]	[4, 3] g:2 f:4 [5, 4] g:2 f:4 [5, 6] g:2 f:6	[5, 4] [4, 5]	[4, 5] g:0 f:4 [4, 4] g:1 f:4 [5, 5] g:1 f:4	
4	4	[4, 3]	[5, 4] g:2 f:4 [5, 6] g:2 f:6 [5, 3] g:3 f:4	[4, 4] [4, 2]	[4, 5] g:0 f:4 [4, 4] g:1 f:4 [5, 5] g:1 f:4 [4, 3] g:2 f:4	
5	5	[5, 4]	[5, 6] g:2 f:6 [5, 3] g:3 f:4	[5, 3] [5, 5] [4, 4]	[4, 5] g:0 f:4 [4, 4] g:1 f:4 [5, 5] g:1 f:4 [4, 3] g:2 f:4 [5, 4] g:2 f:4	
6	6	[5, 3]	[5, 6] g:2 f:6 [5, 2] g:4 f:6 [6, 3] g:4 f:4	[5, 4]	[4, 5] g:0 f:4 [4, 4] g:1 f:4 [5, 5] g:1 f:4 [4, 3] g:2 f:4 [5, 4] g:2 f:4 [5, 3] g:3 f:4	
7	7	[6, 3]	[5, 6] g:2 f:6 [5, 2] g:4 f:6		[4, 5] g:0 f:4 [4, 4] g:1 f:4 [5, 5] g:1 f:4 [4, 3] g:2 f:4	Ketemu [6, 3] [5, 3] [4, 3]

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
					[5, 4] g:2 f:4	[4, 4]
					[5, 3] g:3 f:4	[4, 5]
					[6, 3] g:4 f:4	

Hasil algoritma:

ع	د				ف	ب
ا	ة	ن			ي	ة
ة	ن	ن		ا	ن	ا
م	ن	ن		ن	ا	ا
ن	ن	ن		ن	ن	ا
ن	ن	ن	ن	ا	ن	ا

Gambar 4.11 Matriks awal

ع	د				ف	ب
ا	ة	ن			ي	ة
ة	ن	ن		ا	ن	ا
م	ن	ن		ن	ا	ا
ن	ن	ن	ن	ا	ن	ا
ن	ن	ن	ن	ا	ن	ا

Gambar 4.12 Langkah 1, 2, dan 3

ع	د				ف	ب
ا	ة	ن			ي	ة
ة	ن	ن		ا	ن	ا
م	ن	ن		ن	ا	ا
ن	ن	ن	ن	ا	ن	ا
ن	ن	ن	ن	ا	ن	ا

Gambar 4.13 Langkah 4 (selesai)

Tabel 4.5 Tabel hasil pengujian algoritma *beam a** huruf δ

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
1	1	[2, 0]	[2, 1] g:1 f:6 [1, 0] g:1 f:8 [3, 0] g:1 f:6		[2, 0] g:0 f:6	
2	2	[2, 1]	[3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6	[2, 0] [1, 1] [1, 0]	[2, 0] g:0 f:6 [2, 1] g:1 f:6	
3	3	[3, 0]	[2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6	[3, 1] [2, 0]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6	
4	4	[2, 2]	[3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6	[2, 1] [1, 2] [3, 3]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6	
5	5	[3, 1]	[4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6	[3, 0] [3, 2] [2, 1]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6	
6	6	[4, 0]	[3, 2] g:3 f:6 [4, 1] g:3 f:6 [5, 0] g:3 f:6	[4, 1] [3, 0]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6	
7	7	[3, 2]	[4, 1] g:3 f:6 [5, 0] g:3 f:6 [4, 2] g:4 f:6	[3, 1] [2, 2] [3, 3]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6	

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
					[3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6	
8	8	[4, 1]	[5, 0] g:3 f:6 [4, 2] g:4 f:6 [5, 1] g:4 f:6	[4, 0] [4, 2] [3, 1]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6	
9	9	[5, 0]	[4, 2] g:4 f:6 [5, 1] g:4 f:6 [6, 0] g:4 f:6	[5, 1] [4, 0]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6 [5, 0] g:3 f:6	
10	10	[4, 2]	[5, 1] g:4 f:6 [6, 0] g:4 f:6 [5, 2] g:5 f:6	[4, 1] [3, 2] [4, 3]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6 [5, 0] g:3 f:6	

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
					[4, 2] g:4 f:6	
11	11	[5, 1]	[6, 0] g:4 f:6 [5, 2] g:5 f:6 [6, 1] g:5 f:6	[5, 0] [5, 2] [4, 1]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6 [5, 0] g:3 f:6 [4, 2] g:4 f:6 [5, 1] g:4 f:6	
12	12	[6, 0]	[5, 2] g:5 f:6 [6, 1] g:5 f:6	[6, 1] [5, 0]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6 [5, 0] g:3 f:6 [4, 2] g:4 f:6 [5, 1] g:4 f:6 [6, 0] g:4 f:6	
13	13	[5, 2]	[6, 1] g:5 f:6 [5, 3] g:6 f:8 [6, 2] g:6 f:6	[5, 1] [4, 2]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6	

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
					[4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6 [5, 0] g:3 f:6 [4, 2] g:4 f:6 [5, 1] g:4 f:6 [6, 0] g:4 f:6 [5, 2] g:5 f:6	
14	14	[6, 1]	[5, 3] g:6 f:8 [6, 2] g:6 f:6	[6, 0] [6, 2] [5, 1]	[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6 [4, 1] g:3 f:6 [5, 0] g:3 f:6 [4, 2] g:4 f:6 [5, 1] g:4 f:6 [6, 0] g:4 f:6 [5, 2] g:5 f:6 [6, 1] g:5 f:6	
15	15	[6, 2]	[5, 3] g:6 f:8		[2, 0] g:0 f:6 [2, 1] g:1 f:6 [3, 0] g:1 f:6 [2, 2] g:2 f:6 [3, 1] g:2 f:6 [4, 0] g:2 f:6 [3, 2] g:3 f:6	Ketemu [6, 2] [5, 2] [4, 2] [3, 2] [2, 2] [2, 1]

No	Langkah	Bestnode	Open	Dihapus	Closed	Ket.
					[4, 1] g:3 f:6 [5, 0] g:3 f:6 [4, 2] g:4 f:6 [5, 1] g:4 f:6 [6, 0] g:4 f:6 [5, 2] g:5 f:6 [6, 1] g:5 f:6 [6, 2] g:6 f:6	[2, 0]

Hasil Algoritma:

س	ذ				ف	ب
ا	ة	ن			س	ة
ة	ن	م			س	ة
م	ن	ف		ا	ا	ة
م	ي	ن	س	ن	م	ة
ن	ف	ي	م	ن	م	ة
ن	ب	ة	ن	ا	ن	م

Gambar 4.14 Matriks awal

س	ذ				ف	ب
ا	ة	ن			س	ة
ن	ة	م			س	ة
م	ن	ف		ا	ا	ة
م	ي	ن	س	ن	م	ة
ن	ف	ي	م	ن	م	ة
ن	ب	ة	ن	ا	ن	م

س	ذ				ف	ب
ا	ة	ن			س	ة
ن	ة	م			س	ة
م	ن	ف		ا	ا	ة
م	ي	ن	س	ن	م	ة
ن	ف	ي	م	ن	م	ة
ن	ب	ة	ن	ا	ن	م

س	ذ				ف	ب
ا	ة	ن			س	ة
ن	ة	م			س	ة
م	ن	ف		ا	ا	ة
م	ي	ن	س	ن	م	ة
ن	ف	ي	م	ن	م	ة
ن	ب	ة	ن	ا	ن	م

Gambar 4.15 Langkah 1,2, dan 3

The image shows three 6x6 grids representing the steps of the Beam A* algorithm for sorting the word 'KATA'. Each grid has columns labeled with letters: K, A, T, A. The cells contain numbers representing the cost function. In each step, the minimum value in the open list (shaded grey) is selected and moved to the closed list (shaded yellow).

Gambar 4.16 Langkah 4, 5, dan 6 (selesai)

Tabel 4.6 Tabel kecepatan pengujian algoritma *Beam A** untuk menyusun kata

No.	Kecepatan Beam A* (milisecond)		
	Matriks 5x5	Matriks 6x6	Matriks 7x7
1	40,351667	26,23	127,586667
2	119,783333	118,856667	29,67
3	28,903333	37,813333	5,998333
4	30,365	139,751667	167,355
5	77,659999	7,188333	73,633333
6	57,78	20,406667	417,786667
7	22,526667	108,403333	121,053334
8	9,468334	108,765	151,395001
9	67,551667	18,603334	82,295
10	14,555	23,216666	15,283333
11	29,756667	22,438334	71,276667
12	18,181667	58,546667	251,715
13	23,721667	11,921667	138,558333
14	19,381667	45,675	40,758333
15	7,991667	110,323334	16,136667
16	165,276666	41,868334	14,998334
17	12,136667	57,023334	84,613333
18	29,936666	28,81	26,568334
19	48,151667	14,723333	65,476667
20	24,191666	3,520001	20,408333
21	12,096667	129,08	227,540001
22	29,77	17,711666	41,051666
23	39,148333	84,811667	16,621667
24	25,043333	237,245	11,565
25	17,113333	6,645001	41,12
26	85,653333	61,651667	38,003333
27	17,018334	41,593334	29,176667
28	132,803334	12,831667	41,868333

No.	Kecepatan Beam A* (milisecond)		
	Matriks 5x5	Matriks 6x6	Matriks 7x7
29	11,756666	9,19	38,09
30	57,098333	150,05	17,574999
31	65,423333	84,846667	10,566666
32	24,528333	193,658334	25,038333
33	19,96	36,565	23,123333
34	57,44	12,259999	80,771666
35	31,791666	29,831667	89,888333
Total	1474,316665	2112,056673	2654,566666
Rata-rata	42,12333329	60,34447637	75,84476189

Total waktu untuk 105 pengujian adalah 6.240,940004 milisecond. Dan rata-rata algoritma dapat menyelesaikan rute untuk satu huruf adalah **59,43752385 milisecond** dengan **100%** berhasil menghasilkan rute menuju tujuan.

Tabel 4.7 Tabel hasil pengujian algoritma *beam a** untuk mengacak kata yang memiliki panjang 3 huruf

No	Huruf	Bestnode	Open	Dihapus	Closed	Ket.
1	awal: 0#1#2# Maks. Open = 4 Maks. Normal = panjang kata-1 = 3-1 = 2	[0#1#2#]	(1#0#2#) g:1 f:3 (0#2#1#) g:1 f:3		(0#1#2#) g:0 f:5	
		[1#0#2#]	(0#2#1#) g:1 f:3 (1#2#0#) g:2 f:6	[0#1#2#]	(0#1#2#) g:0 f:5 (1#0#2#) g:1 f:3	
		[0#2#1#]	(1#2#0#) g:2 f:6 (2#0#1#) g:2 f:6	[0#1#2#]	(0#1#2#) g:0 f:5 (1#0#2#) g:1 f:3 (0#2#1#) g:1 f:3	
		[1#2#0#]	(2#0#1#) g:2 f:6		(0#1#2#) g:0 f:5 (1#0#2#) g:1 f:3 (0#2#1#) g:1 f:3 (1#2#0#) g:2 f:6	G=Maks. Normal

Dari hasil uji coba pengacakan dengan panjang 3 karakter sampai 7 karakter 100% berhasil, namun untuk kata yang memiliki panjang 7 karakter algoritma ini masih kurang optimal karena acakan yang seharusnya bisa diurutkan dengan jumlah geseran 6 atau lebih ternyata bisa diselesaikan dengan hanya 4 kali geseran.

Tabel 4.8 Tabel hasil pengujian algoritma *Beam A** untuk menyelesaikan kata *penalty* yang diacak

No	Huruf	Bestnode	Open	Dihapus	Closed	Ket.
1	awal: 2#0#1# target: 0#1#2#	2#0#1#	(0#2#1#) g:1 f:3 (2#1#0#) g:1 f:5		(2#0#1#) g:0 f:5	
		0#2#1#	(2#1#0#) g:1 f:5 (0#1#2#) g:2 f:2		(2#0#1#) g:0 f:5 (0#2#1#) g:1 f:3	
		0#1#2#	(2#1#0#) g:1 f:5		(2#0#1#) g:0 f:5 (0#2#1#) g:1 f:3 (0#1#2#) g:2 f:2	Ketemu [0#1#2#] [0#2#1#] [2#0#1#]

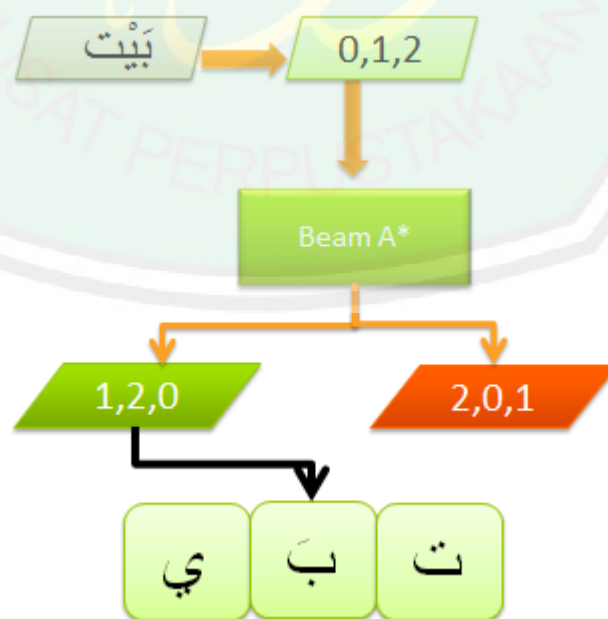
Berikut merupakan contoh atau ilustrasi penggunaan algoritma *Beam A** untuk mengacak kata:

1. Sistem mengambil satu kata yaitu:
 - ✓ Kata jawaban pemain ketika pemain berada di permainan inti dan mendapatkan hukuman *penalti* karena kesalahan yang diperbuat oleh pemain (*puzzle penalty*).
 - ✓ Kata bonus ketika pemain telah menyelesaikan *puzzle* inti dan pemain akan dipindahkan ke permainan untuk *puzzle* bonus.

2. Sistem membuat array indeks berdasarkan panjang huruf dari kata yang diambil, yang kemudian array tersebut akan diproses ke langkah selanjutnya.
3. Array diproses oleh algoritma *Beam A** untuk diperoleh acakannya. Sehingga algoritma tersebut akan berjalan sampai diperoleh acakan yang sesuai dengan nilai batas geseran yang didapat dari:

Batas=panjang kata-1

4. Setelah algoritma selesai dijalankan akan menghasilkan beberapa acakan yang nantinya sistem akan memilih satu acakan tersebut yang digunakan di *puzzle* penalti.
5. *Puzzle* penalti tampil dengan huruf-huruf yang diacak sesuai dengan acakan indeks dari proses algoritma *Beam A**.



Gambar 4.17 proses pengacakan *puzzle* penalti

Tabel 4.9 Tabel kecepatan pengujian algoritma *Beam A** untuk menyelesaikan kata *penalty* yang diacak

No.	Kecepatan (milisecond)
1	97,72667
2	23,67
3	57,725
4	32,22833
5	93,33167
6	21,58667
7	27,66167
8	90,005
9	101,5217
10	97,30333
11	75,88833
12	93,89833
13	53,51833
14	74,71667
15	72,81
16	34,56167
Total	1048,153
Rata-rata	65,50958

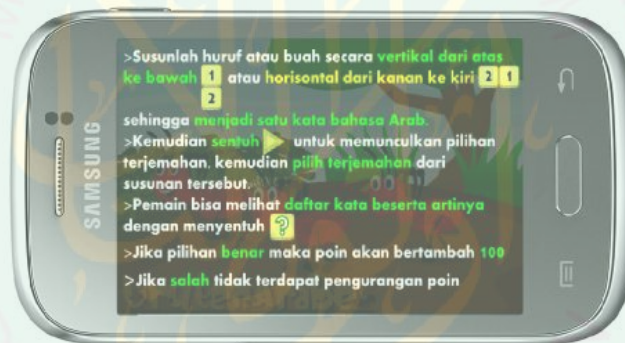
Berdasarkan hasil di atas penggunaan algoritma *Beam A** dapat digunakan juga untuk mengatur huruf-huruf *penalty* dan bonus yang teracak dengan waktu yang cepat yaitu 65,50958 milisecond per geseran dengan keberhasilan mencapai 100%.

4.3 Implementasi Aplikasi *Game*

Berikut adalah tampilan *game* yang telah selesai dibuat



Gambar 4.18 Tampilan menu



Gambar 4.19 Tampilan *help*



Gambar 4.20 Tampilan pilihan level



Gambar 4.21 Tampilan misi



Gambar 4.22 Tampilan daftar kata beserta terjemahan



Gambar 4.23 Tampilan level 1



Gambar 4.24 Tampilan level 2



Gambar 4.25 Tampilan level 3



Gambar 4.26 Tampilan jawaban benar



Gambar 4.27 Tampilan *penalty*



Gambar 4.28 Tampilan menggunakan bantuan atau *hint*



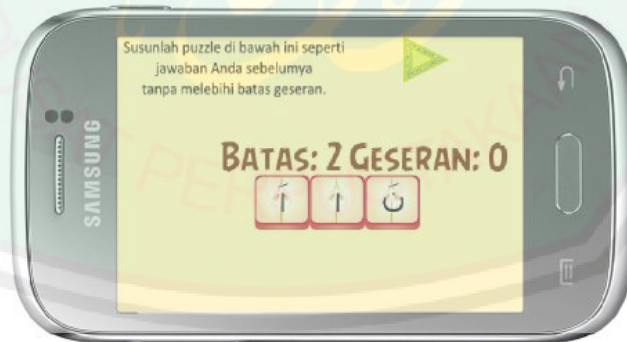
Gambar 4.29 Tampilan ketika *hint* berjalan



Gambar 4.30 Tampilan ketika semua kata telah dijawab



Gambar 4.31 Tampilan kata bonus



Gambar 4.32 Tampilan *puzzle* kata bonus



Gambar 4.33 Tampilan ketika permainan telah selesai

4.4 Pengujian *Game*

Berikut akan dijelaskan tentang hasil pengujian game pada beberapa smartphone.

✓ Sony Xperia SP

Versi Android: v4.3 (Jelly Bean)

GPU: Adreno 320

CPU: Dual-core 1.7 GHz Krait

RAM: 1 GB RAM

Hasil percobaan: **Tampilan dan system berjalan dengan baik**

✓ Evercoss A28M

Versi Android: v4.2 (Jelly Bean)

GPU: Mali 400

CPU: Dual-core 1.3 GHz

RAM: 512 MB

Hasil percobaan: **Tampilan dan system berjalan dengan baik**



Gambar 4.34 Percobaan Hp

✓ Samsung Galaxy Young S6310

Versi Android: v4.1.2 (Jelly Bean)

GPU: Adreno 200

CPU: 1 GHz Cortex-A5

RAM: 768 MB

Hasil percobaan: **Tampilan dan system berjalan dengan baik**



Gambar 4.35 Percobaan Hp

✓ Asus Zenfone 4

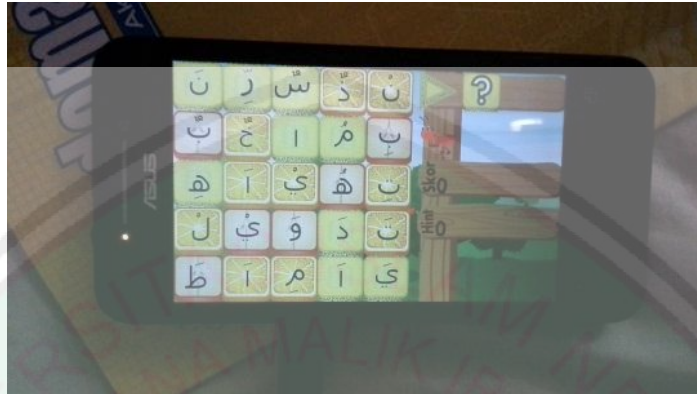
Versi Android: v4.3 (Jelly Bean)

GPU: PowerVR SGX544MP2

CPU: Dual-core 1.2 GHz

RAM: 1 GB

Hasil percobaan: **Tampilan dan system berjalan dengan baik**



Gambar 4.36 Percobaan Hp

4.5 Kuesioner Pengujian *Game*

Berikut pernyataan-pernyataan kuesioner:

1. Saya tahu cara memainkan game puzzle ini
2. Game ini membantuku untuk menghafal kata bahasa Arab beserta artinya
3. Game ini membuat saya lebih semangat untuk belajar bahasa Arab
4. Saya ingin memainkan game ini lagi
5. Tampilan gambar dalam game ini menarik
6. Game ini menarik dan menyenangkan
7. Saya tidak mengalami kesulitan dalam memainkan game

Dan untuk setiap pernyataan kuesioner diberi tiga opsi tanggapan, yaitu:

- Setuju
- Kurang setuju
- Tidak setuju

Nama *

Saya tahu cara memainkan game puzzle ini *

- Setuju
- Kurang Setuju
- Tidak setuju

Game ini membantuku untuk menghafal kata bahasa Arab beserta artinya *

- Setuju
- Kurang Setuju
- Tidak Setuju

Game ini membuat saya lebih semangat untuk belajar bahasa Arab *

- Setuju
- Kurang Setuju
- Tidak Setuju

Saya ingin memainkan game ini lagi *

- Setuju
- Kurang Setuju
- Tidak Setuju

Tampilan gambar dalam game ini menarik *

- Setuju
- Kurang Setuju
- Tidak Setuju

Game ini menarik dan menyenangkan *

- Setuju
- Kurang Setuju
- Tidak Setuju

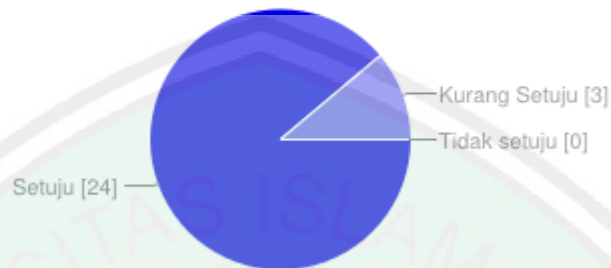
Saya tidak mengalami kesulitan dalam memainkan game *

- Setuju
- Kurang Setuju
- Tidak Setuju

Gambar 4.37 Form kuesioner

Hasil kuesioner:

Saya tahu cara memainkan game puzzle ini

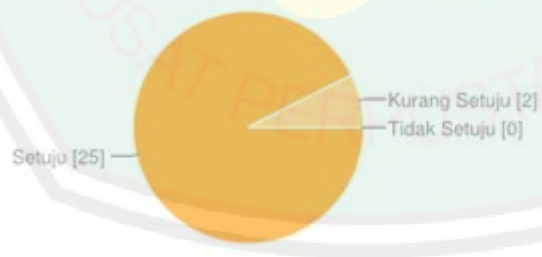


Gambar 4.38 Hasil pernyataan 1

Untuk pernyataan pertama yaitu saya tahu cara memainkan *game puzzle* ini didapatkan hasil seperti di bawah ini:

- **Setuju:** 24 pengguna (89%)
- Kurang setuju: 3 pengguna (11%)
- Kurang setuju: 0 pengguna (0%)

Game ini membantuku untuk menghafal kata bahasa Arab beserta artinya



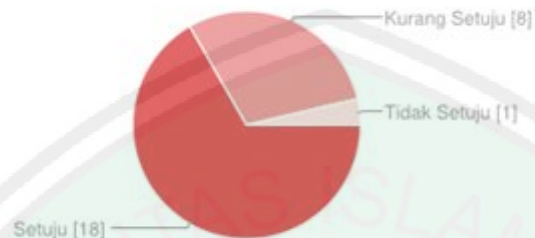
Gambar 4.39 Hasil pernyataan 2

Untuk pernyataan game ini membantuku untuk menghafal kata bahasa Arab beserta artinya didapatkan hasil seperti di bawah ini:

- **Setuju:** 25 pengguna (93%)
- Kurang setuju: 2 pengguna (7%)

- Tidak setuju: 0 pengguna (0%)

Game ini membuat saya lebih semangat untuk belajar bahasa Arab

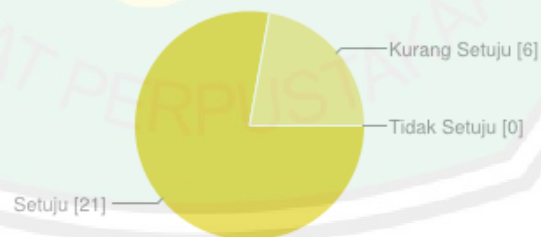


Gambar 4.40 Hasil pernyataan 3

Untuk pernyataan game ini membuat saya lebih semangat untuk belajar bahasa Arab didapatkan hasil seperti di bawah ini:

- **Setuju:** 18 pengguna (67%)
- Kurang setuju: 8 pengguna (30%)
- Kurang setuju: 1 pengguna (4%)

Saya ingin memainkan game ini lagi



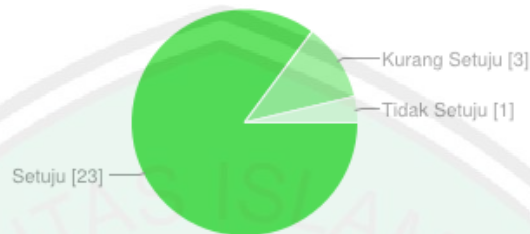
Gambar 4.41 Hasil pernyataan 4

Untuk pernyataan saya ingin memainkan game ini lagi didapatkan hasil seperti di bawah ini:

- **Setuju:** 21 pengguna (78%)
- Kurang setuju: 6 pengguna (22%)

- Tidak setuju: 0 pengguna (0%)

Tampilan gambar dalam game ini menarik

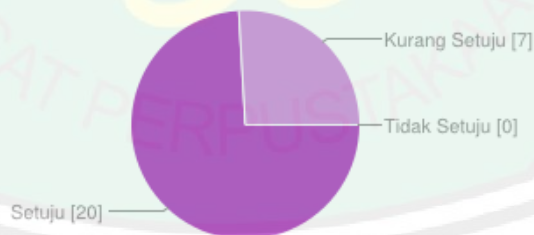


Gambar 4.42 Hasil pernyataan 5

Untuk pernyataan tampilan gambar dalam game ini menarik didapatkan hasil seperti di bawah ini:

- **Setuju:** 23 pengguna (85%)
- Kurang setuju: 3 pengguna (11%)
- Kurang setuju: 1 pengguna (4%)

Game ini menarik dan menyenangkan



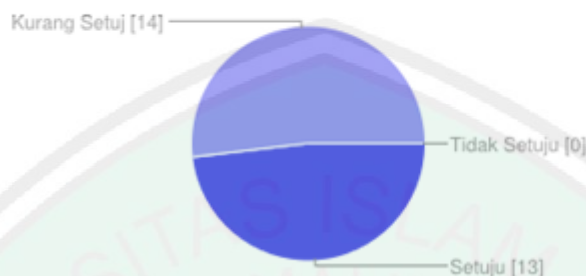
Gambar 4.43 Hasil pernyataan 6

Untuk pernyataan game ini menarik dan menyenangkan didapatkan hasil seperti di bawah ini:

- **Setuju:** 20 pengguna (74%)
- Kurang setuju: 7 pengguna (26%)

- Tidak setuju: 0 pengguna (0%)

Saya tidak mengalami kesulitan dalam memainkan game



Gambar 4.44 Hasil pernyataan 7

Untuk pernyataan saya tidak mengalami kesulitan dalam memainkan game ini didapatkan hasil seperti di bawah ini:

- Setuju: 13 pengguna (48%)
- **Kurang setuju:** 14 pengguna (52%)
- Tidak setuju: 0 pengguna (0%)

Berdasarkan hasil kuesioner di atas dapat disimpulkan bahwa *game* ini bisa menjadi salah satu media belajar alternatif untuk pembelajaran kosa kata bahasa Arab.

4.6 Integrasi Dalam Islam

Bahasa Arab merupakan salah satu bahasa yang digunakan oleh beberapa negara sebagai bahasa resmi atau bahasa nasional negara tersebut, seperti negara Arab, Mesir, dan lain-lain. Bahasa Arab telah dijadikan sebagai salah satu bahasa resmi dalam lingkungan Perserikatan Bangsa-Bangsa (PBB) pada tahun 1973, yang sekaligus meningkatkan kedudukan bahasa Arab itu sendiri. Selain itu bahasa Arab juga digunakan sebagai bahasa al-Qur'an sebagaimana diterangkan dalam al-Qur'an surat Yusuf ayat 2:

إِنَّا أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَعَلَّكُمْ تَعْقِلُونَ

Artinya: “Sesungguhnya Kami menurunkannya berupa Al-Quran dengan berbahasa Arab, agar kamu memahaminya.”

Dalam tafsir Ibnu Katsir dijelaskan bahwa bahasa Arab adalah bahasa yang paling fasih, paling jelas, paling luas dan paling tepat untuk dapat menyampaikan makna (maksud) yang ada di dalam jiwa. Oleh karena itu, Kitab yang paling mulia ini diturunkan dengan bahasa yang paling mulia, kepada Rasul yang paling mulia, dengan utusan Malaikat yang paling mulia, di bumi yang mulia, diturunkan pada bulan yang paling mulia, yaitu bulan Ramadhan. Oleh karena itu bahasa Arab menjadi bahasa yang sangat penting, khususnya bagi umat Islam.

Penelitian ini bertujuan untuk membuat aplikasi yang dapat membantu dan mempermudah pengguna untuk belajar dan menghafalkan kata-kata bahasa Arab beserta artinya. Dalam aplikasi atau *game* ini bahasa Arab digunakan sebagai elemen inti karena *game* ini menggunakan kata bahasa Arab beserta artinya sebagai materi. Sehingga diharapkan pengguna akan lebih mudah untuk belajar dan menghafal kata-kata bahasa Arab yang terdapat dalam *game* ini. Karena sesungguhnya belajar merupakan kewajiban bagi semua umat Islam. Seperti yang dijelaskan dalam al Qur'an surat Shod ayat 29:

كِتَابٌ أَنْزَلْنَاهُ إِلَيْكَ مُبَارَكٌ لِيَدَّبَّرُوا آيَاتِهِ وَلِيَتَذَكَّرَ أُولُوا الْأَلْبَابِ

Artinya: “Ini adalah sebuah kitab yang Kami turunkan kepadamu penuh dengan berkah supaya mereka memperhatikan ayat-ayatnya dan supaya mendapat pelajaran orang-orang yang mempunyai pikiran.”

Dan dalam Al Hadist tentang menuntut ilmu:

طَلَبُ الْعِلْمِ فَرِيضَةٌ عَلَى كُلِّ مُسْلِمٍ وَ مُسْلِمَةٍ - رواه ابن عبد البر

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

BAB V

PENUTUP

5.1 Kesimpulan

Algoritma *Beam A** sangat cocok digunakan untuk menentukan rute arah gerak potongan kata karena algoritma ini dapat menemukan rute dengan waktu yang cukup cepat yakni 59,438 milisecond per huruf dan dengan keberhasilan mencapai 100%. Selain itu penggunaan algoritma *Beam A** untuk mengacak kata dalam *game* ini terbukti mampu untuk mengacak kata yakni 100% berhasil walaupun untuk kasus dengan jumlah 7 huruf masih menghasilkan acakan yang kurang optimal dikarenakan acakan bisa disusun lebih cepat dari batas geseran.

5.2 Saran

Penulis sadar bahwa penelitian dan pembuatan *game* ini masih memiliki banyak kekurangan, sehingga diharapkan di waktu yang akan datang semoga ada perbaikan, diantaranya:

1. Menambah level dan materi bahasa Arab untuk *game* sehingga *game* ini lebih edukatif dan menyenangkan.
2. Diharapkan *game* ini bisa dikembangkan untuk sistem operasi yang lain.
3. Perbaiki algoritma untuk kasus pengacakan kata, karena untuk kasus ini algoritma *Beam A** dinilai masih kurang optimal.

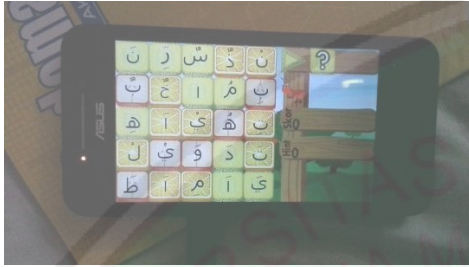
Daftar Pustaka

- Suyanto. 2011. *Artificial Intelligence (Edisi Revisi)*. Bandung: Penerbit Informatika
- Safaat, Nazruddin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika
- Safaat H, Nazruddin. 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika
- Mejia, Gonzalo, & Odrey, Nicholas G. 2006. *An Approach Using Petri Nets And Improved Heuristic Search For Manufacturing System Scheduling*. Journal of Manufacturing System
- Maulana I, Dimas. 2010. *Penerapan Algoritma Breadth First Search dan Depth First Search pada Permainan Slide Puzzle*. Universitas Diponegoro
- Iqbal, Aldy Mereyano. 2013. *Analisis perfomasi algoritma beam pada pencarian rute terpendek di Kota Tasikmalaya*. UNIKOM
- Usman, Chandra & Candra, Kenedy. 2013. *Rancang Bangun Game Slider Puzzle Berbasis Android Menggunakan Metode Heuristik dengan Teknik Best First Search*. STMIK MDP
- Rouse, Richard. 2001. *Game Design: Theory & Practice*. United States: Wordware Publishing
- Xin, C. 2009. *Multiplayer Game in Mobile Phone Serious Game*. International Joint Conference on Artificial Intelligence
- Putra, Yogie Susdyastama, Muslim M. Aziz, & Naba, Agus. 2013. *Game Chicken Roll dengan Menggunakan Metode Forward Chaining*. Jurnal EECCIS Vol. 7, No. 1, 2013
- Zechner, M. 2011. *Beginning Android Games*. New York: Apress
- Kusno, Gustaaf. 2011. *Sepuluh Bahasa Terbanyak Dipakai di Dunia*. <http://bahasa.kompasiana.com/2011/09/16/sepuluh-bahasa-terbanyak-dipakai-di-dunia-395936.html>. Diakses tanggal 05 Februari 2014 pukul 07.00 WIB

- Breed, Henry. 2011. *Frequently Asked Questions (FAQs)*. <http://www.un.org/depts/DGACM/faqs.shtml>. Diakses tanggal 05 Februari 2014 pukul 07.10 WIB
- Didno. 2013. *Macam-macam Sistem Operasi Pada Smartphone*. <http://www.didno76.com/2013/04/macam-macam-sistem-operasi-pada.html>. Diakses tanggal 05 Februari 2014 pukul 07.30 WIB
- Aeni, Wiwik Akhirul. 2009. *Antara Game, Pendidikan dan HP (Game Mobile Learning Sebagai Wacana Pendidikan)*. <http://m-edukasi.kemdikbud.go.id/artikel-mobile-learning-isi.php?kodenya=2009-ac>. Diakses tanggal 05 Februari 2014 pukul 07.53 WIB
- Librianty, Andina. 2013. *Ini Alasan Android Laris Manis di Indonesia*. <http://techno.okezone.com/read/2013/01/31/57/754702/ini-alasan-android-laris-manis-di-indonesia>. Diakses tanggal 05 Februari 2014 pukul 08.03 WIB
- Al-Fauzan, Abdurrahman bin Ibrahim, Husain, Mukhtar At-Thahir & Fadl, Muhammad Abdul Khaliq Muhammad. 2007. *Al 'Arabiyah baina yadaik jilid 1*. Riyadh: Al-Arabiyyah Lil jami'
- Statcounter. 2014. <http://statcounter.com/>. Diakses tanggal 05 Februari 2014 pukul 09.35 WIB
- Al- Mubarakfury, Syaikh Shafiiyur Rahman. *Shahih Tafsir Ibnu Katsir*. Pustaka Ibnu Katsir

Lampiran

Pengujian Game Pada Smartphone



Asus Zenfone 4



Evercoss A28M



Samsung Galaxy Young S6310