

Lembar Pengesahan

Implementasi Algoritma *Directed Acyclic Word Graph* dalam Perancangan Game *Scrabble*

Implementation Directed Acyclic Word Graph Algorithm On Scrabble Design

**Arie Lasaprima
113088060**

Tugas Akhir ini telah diterima dan disahkan untuk memenuhi sebagian dari syarat
untuk memperoleh gelar Sarjana Teknik Fakultas Teknik Informatika
Institut Teknologi Telkom

Bandung, Juni 2011

Menyetujui

Pembimbing I

Pembimbing II

Agung Toto Wibowo, ST., MT.
NIP : 06810330 – 1

ZK Abdurahman B, S.Si., M.Kom.
NIP : 99750185 – 1

Abstrak

Menerapkan AI pada komputer yang dapat melakukan proses perhitungan lebih cepat dibanding manusia dapat membuat sebuah permainan menjadi lebih menantang dan tidak membosankan. Salah satu platform yang bagus untuk menguji teknik AI pada game yaitu game *scrabble*^[11].

Scrabble adalah permainan kata dimana dua sampai empat pemain membentuk kata dari tujuh huruf yang ada pada mereka pada sebuah papan yang mempunyai luas 15x15 kotak. Permainan ini telah dijual di 121 negara sepanjang dunia dan lebih dari 100 juta set permainan telah terjual^{[12][13]}, di Amerika tiap tiga rumah mempunyai satu set dari permainan ini^[12]. Penerapan AI pada *scrabble* agak sedikit berbeda dengan beberapa permainan lain, teknik AI yang sukses diterapkan pada beberapa permainan yang bersifat deterministik seperti catur, checkers, reversi, awari, hex, go, renju, amazons, pente, dan lainnya tidak cocok diterapkan pada *scrabble*^[11].

AI *scrabble* yang selalu memilih kata dengan nilai tertinggi dari kemungkinan kata yang dapat disusun di setiap giliran akan dengan sangat mudah dikalahkan oleh pemain manusia yang ahli dalam *scrabble*^[11]. Game ini dirancang agar manusia dapat melawan AI yang yang tidak gampang dikalahkan sehingga permainan menjadi lebih menarik dan menantang. Oleh karena itu dirancang strategi permainan sehingga manusia tidak dapat dengan mudah mengalah kan AI system. AI system akan mengevaluasi pemilihan kata yang akan dimainkan di tiap giliran dimana tiap pilihan kata akan mempengaruhi baik papan permainan maupun kemungkinan pilihan kata di giliran berikut. Pada laporan ini untuk membentuk kamus data dalam menyimpan semua kata yang valid digunakan algoritma *DAWG*, algoritma *DAWG* dapat memperkecil ruang penyimpanan dan membantu proses meretrieve kata yang akan dimainkan selain itu dianalisis tiga strategi permainan yang dibentuk dengan membandingkan ketiga strategi tersebut dalam 60 kali percobaan untuk melihat performansi masing-masing strategi.

Berdasarkan uji coba yang telah dilakukan, strategi permainan *scrabble* yang setiap langkahnya hanya berdasarkan score kata yang dibentuk mempunyai persentase kemenangan hanya 40% melawan strategi permainan *scrabble* yang mempunyai evaluasi pemilihan kata.

Kata kunci: Game *Scrabble*, Algoritma *directed acyclic word graph*

Abstract

Applying AI to the computer that can perform the calculations faster than a human can make a game becomes more challenging and not boring. One of the great platform to test the techniques of AI in the game is Scrabble^[11].

Scrabble is a board game in which two to four players form a words from the seven letters that give on them that placed on a board that has a large 15x15 square. These games have been sold in 121 countries throughout the world and more than 100 million sets of the game have been sold^{[12][13]}, in the U.S. every three homes has a set of these games^[12]. Implement of AI in Scrabble is a bit different than some other games, AI techniques are successfully applied to some deterministic games like chess, checkers, reversion, login or register, hex, go, renju, Amazons, pente, and others are not suitable to be applied at Scrabble^[11].

Scrabble AI that always chooses words with the highest value of the possible words that can be arranged at every turn will very easily be defeated by a human expert in the Scrabble^[11]. This game is designed for humans to fight the AI which is not easily defeated, so the game becomes more interesting and challenging. Therefore designed a strategy game so that people can not easily succumb's AI system. AI system will evaluate the selection of words that will be played in each turn in which each word choice will affect both the game board and the possible options next turn. In this report, to establish a data dictionary to store all the valid words using a dawg algorithms, dawg algorithm can reduce the storage space and help to retrieve words that will be played in addition analyzed three strategy game that is formed, by comparing the three strategies in 60 attempts to see the performance of each strategy.

Based on trials that have been done, a strategy game of Scrabble that each step is only based on the score of words that have formed only 40% winning percentage against the strategy game of Scrabble that has the word selection evaluation.

Key words: Scrabble, directed acyclic word graph algorithm

Lembar Persembahan

Dalam kesempatan ini penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah memberikan doa, semangat, dan bantuan dalam penyelesaian tugas akhir ini. Dengan segala kerendahan hati, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala rahmat yang diberikan sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.
2. Kepada orangtuaku tercinta, bapak, ibu, adik-adik dan semua keluarga yang terus memberi semangat dan motivasi. Yang selalu mendoakan dan memberikan kasih sayang dan kesabaran tanpa henti.
3. Bapak Agung Toto Wibowo, S.T., M.T. selaku pembimbing pertama dan Bapak ZK Abdurahman B, S.Si., M.Kom. selaku pembimbing kedua atas semua waktu, motivasi, dan bimbingan yang telah diberikan.
4. Seluruh dosen IT Telkom atas semua ilmu yang telah diberikan dan seluruh staf administrasi Fakultas Teknik Infomatika yang telah banyak membantu selama mengurus administrasi perkuliahan.
5. Teman-teman di IT Telkom baik teman-teman ekstensi 2008 maupun teman-teman angkatan lain yang sempat sekelas, semoga tali silaturahmi kita selalu terjaga.
6. Kepada semua teman-teman yang telah banyak membantu dalam menyelesaikan aplikasi baik yang hanya berniat maupun yang telah dan banyak membantu.
7. Kepada semua teman-teman yang telah menjadi sahabat dan teman baik penulis yang walaupun tidak banyak membantu tapi selalu memberikan semangat, motivasi, dukungan dan doa yang lebih berarti dari pada hanya sekedar bantuan.
8. Semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak bisa disebutkan satu persatu.

Kata Pengantar

Segala puji dan syukur penulis panjatkan kehadiran Allah SWT karena atas kehendak-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “Implementasi Algoritma Directed Acyclic Word Graph pada Perancangan Game Scrabble”. Tugas akhir ini disusun sebagai salah satu syarat kelulusan pada Program Sarjana Teknik Informatika di Institut Teknologi Telkom.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna yang disebabkan karena keterbatasan yang penulis miliki. Untuk itu saran dan kritik yang bersifat membangun dari pembaca sangat penulis harapkan demi perbaikan di masa yang akan datang.

Dengan segala kerendahan hati, penulis berharap semoga tugas akhir ini dapat bermanfaat bagi pembaca dan penulis khususnya, serta bagi dunia pendidikan pada umumnya.

Bandung, Juni 2011

Penulis

Daftar Isi

| | |
|---|-------------|
| ABSTRAK | I |
| ABSTRACT | II |
| LEMBAR PERSEMBAHAN | III |
| KATA PENGANTAR | IV |
| DAFTAR ISI | V |
| DAFTAR GAMBAR | VI |
| DAFTAR TABEL | VII |
| DAFTAR ISTILAH | VIII |
| 1. PENDAHULUAN | VIII |
| 1.1 LATAR BELAKANG..... | 4 |
| 1.2 PERUMUSAN MASALAH | 4 |
| 1.3 TUJUAN | 4 |
| 1.4 METODOLOGI PENYELESAIAN MASALAH | 4 |
| 2. LANDASAN TEORI | VIII |
| 2.1 GAME SCRABBLE | 10 |
| 2.1.1 <i>Sejarah Scrabble</i> | 4 |
| 2.1.2 <i>Aturan permainan</i> | 5 |
| 2.2 ALGORITMA DAWG..... | 10 |
| 2.3 STRUKTUR PENYIMPANAN KAMUS DATA PADA JAVA | 10 |
| 3. ANALISIS DAN PERANCANGAN SISTEM | VIII |
| 3.1 ANALISIS KEBUTUHAN | 10 |
| 3.1.1 <i>Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak</i> | 4 |
| 3.1.2 <i>Analisis Kebutuhan Fungsional</i> | 4 |
| 3.2 PERANCANGAN SISTEM | 10 |
| 3.3 ALGORITMA DIRECTED ACYCLIC WORD GRAPH | 10 |
| 3.4 IMPLEMENTASI ALGORITMA DAWG | 10 |
| 3.5 PERANCANGAN STRATEGI SCRABBLE | 10 |
| 4. PENGUJIAN SISTEM DAN ANALISIS HASIL | VIII |
| 4.1 TUJUAN PENGUJIAN | 10 |
| 4.2 STRATEGI PENGUJIAN | 10 |
| 4.2.1 <i>Pengujian Algoritma Directed Acyclic Word Graph</i> | 4 |
| 4.2.2 <i>Analisa Pengujian Strategi Permainan Scrabble</i> | 4 |
| 4.3 ANALISIS HASIL PENGUJIAN | 30 |
| 4.3.1 <i>Analisa Pengujian Algoritma Pembentuk Kamus Data</i> | 30 |
| 4.3.2 <i>Hasil Analisa Pengujian Strategi Permainan Scrabble</i> | 32 |
| 5. KESIMPULAN DAN SARAN | VIII |
| 5.1 KESIMPULAN | 32 |
| 5.2 SARAN | 32 |
| DAFTAR PUSTAKA | VIII |
| LAMPIRAN A: DATA PENGUJIAN | VIII |

Daftar Gambar

| | |
|---|----|
| GAMBAR 2.1 : PERMAINAN SCRABBLE..... | 6 |
| GAMBAR 2.2 : PAPAN PERMAINAN SCRABBLE | 6 |
| GAMBAR 2.3 : HURUF-HURUF DALAM PERMAINAN SCRABBLE..... | 6 |
| GAMBAR 2.4 : PERMAINAN SCRABBLE DIMULAI..... | 7 |
| GAMBAR 2.5 : POHON HURUF..... | 8 |
| GAMBAR 2.6 : A DIRECTED ACYCLIC WORD GRAPH..... | 9 |
| GAMBAR 2.7 : HASH TABLE | 10 |
| GAMBAR 2.8 : ARRAY LIST BERANTAI | 11 |
| GAMBAR 3.1 : DIAGRAM FLOWCHART GAME SCRABBLE | 6 |
| GAMBAR 3.2 : POHON HURUF ATAU KATA..... | 6 |
| GAMBAR 3.3 : PROSES ALGORITMA DAWG LANGKAH 1..... | 6 |
| GAMBAR 3.4 : PROSES ALGORITMA DAWG LANGKAH 2..... | 6 |
| GAMBAR 3.5 : PROSES ALGORITMA DAWG LANGKAH 3..... | 6 |
| GAMBAR 3.6 : PROSES ALGORITMA DAWG LANGKAH 4..... | 6 |
| GAMBAR 3.7 : PROSES ALGORITMA DAWG LANGKAH 5..... | 6 |
| GAMBAR 3.8 : PROSES ALGORITMA DAWG LANGKAH 6..... | 6 |
| GAMBAR 3.9 : PROSES ALGORITMA DAWG LANGKAH 7..... | 6 |
| GAMBAR 3.10 : PROSES ALGORITMA DAWG LANGKAH 8..... | 19 |
| GAMBAR 3.11 : PROSES ALGORITMA DAWG LANGKAH 9..... | 19 |
| GAMBAR 3.12 : PROSES PENCARIAN KEMUNGKINAN KATA LANGKAH 1 | 6 |
| GAMBAR 3.13 : PROSES PENCARIAN KEMUNGKINAN KATA LANGKAH 2 | 6 |
| GAMBAR 3.14 : PROSES PENCARIAN KEMUNGKINAN KATA LANGKAH 3 | 6 |
| GAMBAR 3.15 : PROSES PENCARIAN KEMUNGKINAN KATA LANGKAH 4. | 6 |
| GAMBAR 3.16 : PROSES PENCARIAN KEMUNGKINAN KATA LANGKAH 5. | 6 |
| GAMBAR 3.17 : PROSES PENCARIAN KEMUNGKINAN KATA LANGKAH 6. | 6 |
| GAMBAR 3.18 : KONDISI PAPAN CONTOH PERTAMA..... | 6 |
| GAMBAR 3.19 : KONDISI PAPAN CONTOH KEDUA..... | 6 |

Daftar Tabel

| | |
|---|----|
| TABEL 3.1 : FORM PERBANDINGAN SAMPEL ACAK KOMBINASI HURUF | 24 |
| TABEL 3.2 : FORM KATA YANG DAPAT DIBENTUK PADA CONTOH KEDUA | 24 |
| TABEL 3.3 : TABEL BASIC RACK EVALUATION..... | 27 |
| TABEL 4.1 : PENGUJIAN PERBANDINGAN DUA HURUF | 24 |
| TABEL 4.2 : TABEL BASIC RACK EVALUATION. | 24 |
| TABEL 4.3 : FORM HASIL UJI KECERDASAN AI ALGORITMA MINIMAX. | 30 |
| TABEL 4.4 : TABEL PENGUJIAN ALGORITMA DAWG | 31 |
| TABEL 4.5 : TABEL HASIL PERBANDINGAN DUA HURUF | 32 |
| TABEL 4.6 : TABEL HASIL PERBANDINGAN STRATEGI 1 VS STRATEGI 2 10 KALI PERCOBAAN | 33 |
| TABEL 4.7 : TABEL HASIL PERBANDINGAN STRATEGI 1 VS STRATEGI 3 10 KALI PERCOBAAN | 34 |
| TABEL 4.8 : TABEL HASIL PERBANDINGAN STRATEGI 2 VS STRATEGI 3 10 KALI PERCOBAAN | 35 |
| TABEL 4.9 : TABEL HASIL PERBANDINGAN BEBERAPA STRATEGI 60 KALI PERCOBAAN..... | 36 |
| TABELA-1 DATA PENGUJIAN BEBERAPA PASANG STRATEGI DARI 60 KALI PERCOBAAN | 24 |

Daftar Istilah

| | |
|-------------|--|
| <i>Node</i> | Simpul pada <i>tree</i> , yang dibangun dari <i>root</i> hingga daun |
| <i>Rack</i> | Tempat menampung tujuh huruf acak yang dibagikan tiap pemain |

1. Pendahuluan

1.1 Latar Belakang

Seiring perkembangan jaman dan teknologi, perilaku manusia dalam berfikir sangat mungkin untuk diterapkan pada komputer khususnya pada game, penerapan perilaku manusia ini disebut artificial intelligence atau kecerdasan buatan, dengan menerapkan AI pada komputer yang dapat melakukan proses perhitungan lebih cepat dibanding manusia dapat membuat sebuah permainan menjadi lebih menantang dan tidak membosankan.

Salah satu platform yang bagus untuk menguji teknik AI pada game yaitu game *scrabble*^[11]. *Scrabble* adalah permainan kata dimana dua sampai empat pemain membentuk kata dari tujuh huruf yang ada pada mereka pada sebuah papan yang mempunyai luas 15x15 kotak. Permainan ini telah dijual di 121 negara sepanjang dunia dan lebih dari 100 juta set permainan telah terjual^{[12][13]}, di Amerika tiap tiga rumah mempunyai satu set dari permainan ini^[12]. Pada akhir permainan yaitu pada saat huruf yang disediakan pada kantong telah habis, pemain yang mempunyai total jumlah nilai tertinggi akan menjadi pemenangnya.

Penerapan AI pada *scrabble* agak sedikit berbeda dengan beberapa permainan lain, teknik AI yang sukses diterapkan pada beberapa permainan yang bersifat deterministik seperti catur, checkers, reversi, awari, hex, go, renju, amazons, pente, dan lainnya tidak cocok diterapkan pada *scrabble*^[11]. Perbedaan yang mendasar antara permainan tersebut dengan *scrabble* yaitu terletak pada informasi-informasi pada permainan ini, banyaknya informasi yang bisa digunakan untuk membangun sebuah strategi permainan pada permainan seperti catur, checkers, reversi, awari, hex, go, renju, amazons, pente, hal ini membuat faktor percabangan state menjadi lebih sedikit sehingga semua permainan yang disebutkan diatas disebut *perfect-information games*, sedangkan pada *scrabble* keping huruf yang ada pada lawan tidak diketahui dan keping yang akan kita dapat pun tidak diketahui sehingga langkah yang akan dipilipun tidak dapat diperkirakan dan informasi yang didapat untuk membangun strategi permainan ini sangat sedikit sehingga *scrabble* disebut *a game of imperfect information* dengan faktor percabangan state yang sangat besar^[11] dimana *scrabble* mempunyai ruang kemungkinan yang lebih besar dibanding catur bahkan lebih besar dibanding permainan GO^[11].

AI *scrabble* yang selalu memilih kata dengan nilai tertinggi dari kemungkinan kata yang dapat disusun di setiap giliran akan dengan sangat mudah dikalahkan oleh pemain manusia yang ahli dalam *scrabble*^[11] karena hal tersebut tidak menjamin pemain dapat mengalahkan lawan atau memenangkan permainan. Saat ini algoritma yang akan diterapkan untuk membantu membangun Artificial Intelligence pada permainan kata ini adalah *directed acyclic word graph*, *directed acyclic word graph (DAWG)* adalah data struktur yang mempunyai efisiensi ruang dalam menangani dan menganalisa pengulangan pada sebuah text^[3] sehingga dalam penggunaannya *DAWG* membutuhkan memory yang sangat kecil^[7]. Penelusuran kata pada *DAWG* seperti pada sebuah trie atau pohon data dimana setiap awalan dan akhiran digunakan bersama-sama untuk memperkecil jumlah ruang yang diperlukan untuk menyimpan kosakata^[2] dan juga memperkecil waktu yang diperlukan untuk meretrieve kata^[7].

1.2 Perumusan Masalah

Dalam perancangan AI game scrabble ini perlu diperhatikan beberapa hal yaitu :

- a) Bagaimana menerapkan algoritma *DAWG* pada permainan scrabble dalam membangkitkan atau meretrieve kata.
- b) Bagaimana memberi penilaian yang benar pada fungsi heuristik yang dibentuk dalam pemilihan kata dimana fungsi heuristik nya :
 - Keping huruf yang tertinggal pada rack
 - Pasangan huruf yang tertinggal di rack
 - Evaluasi pemilihan langkah

Batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Game ini dimainkan oleh dua pemain yaitu manusia melawan komputer.
2. Game ini mempunyai beberapa strategi permainan.
3. Papan game berukuran 15x15 kotak.
4. Algoritma *Dawg* hanya untuk membentuk kamus data yang akan digunakan dalam permainan juga membantu pada proses searching dan matching kata yang valid.
5. Kata-kata yang valid atau dapat dimainkan hanya kata-kata yang terdapat dalam kamus yang dipakai dalam laporan ini.

1.3 Tujuan

Tujuan yang akan dicapai dari tugas akhir ini adalah :

1. Menganalisis dan membangun kamus data dengan menggunakan algoritma *DAWG* yang dapat membangkitkan kata dengan benar dan tepat.
2. Mengimplementasikan algoritma *DAWG* dalam membantu proses meretrieve semua kata yang dapat dibentuk.
3. Membangun strategi permainan yang bagus pada permainan *scrabble* sehingga komputer tidak gampang dikalahkan ataupun memenangkan permainan.

1.4 Metodologi Penyelesaian Masalah

Metodologi yang akan digunakan untuk menyelesaikan tugas akhir ini adalah :

1. Study Literatur
Melakukan pengumpulan data-data dari berbagai sumber yang berhubungan atau berkaitan dengan game *Scrabble* dan algoritma *DAWG* sebagai acuan dan dasar perancangan game.
2. Analisis dan Perancangan
Pada tahap ini dilakukan analisis dan perancangan algoritma dalam membuat game, membangun algoritma *DAWG* yang dapat membangkitkan kata dengan benar dan tepat dimana dalam perancangannya tidak ada informasi yang redundant dan menerapkannya dalam game, menganalisa penggabungan nilai probabilistik huruf pada kantong dengan nilai-nilai pada rack dan posisi penempatan kata pada papan dalam pemilihan langkah atau pemilihan kata oleh komputer.

3. Implementasi
Mengimplementasikan algoritma *DAWG* dan fungsi-fungsi heuristic nya untuk membangkitkan kata dalam membangun *game Scrabble* ini.
4. Pengujian dan Analisis Hasil
Melakukan evaluasi dan pengujian terhadap program yang dibentuk atau hasil dari implementasi algoritma *DAWG* dan fungsi heuristik yang diterapkan dan mengevaluasi performansi system AI dengan cara :
 - Membangun beberapa strategi permainan.
 - Mensimulasikan komputer melawan komputer antara strategi yang dirancang sehingga terlihat perbedaan score yang diperoleh.
5. Penyusunan Laporan
Pada tahap ini dilakukan pengambilan kesimpulan dari hasil analisis yang telah dilakukan pada tahap sebelumnya untuk kemudian disusun laporan terhadap analisis yang telah dilakukan dalam bentuk buku tugas akhir.

2. Landasan Teori

2.1 Game Scrabble

2.1.1 Sejarah Scrabble

Permainan Scrabble mulanya diciptakan di tahun 1938 dengan nama "*Criss-Crosswords*" oleh seorang arsitek bernama Alfred Mosher Butts. Permainan ini merupakan penyempurnaan dari permainan *Lexiko* yang lebih dulu diciptakannya, tapi dilengkapi papan permainan dan cara bermain seperti teka-teki silang. Permainan tetap menggunakan keping huruf seperti *Lexiko* yang dibuat berdasarkan hasil perhitungan distribusi frekuensi penggunaan huruf berbagai tulisan berbahasa Inggris, termasuk artikel surat kabar *The New York Times*. Alfred Butts memproduksi sendiri permainan ini dan menawarkannya ke berbagai perusahaan mainan besar tapi tidak laku.^[12]

Di tahun 1948, pengacara James Brunot, penduduk asal Newton, Connecticut membeli hak memproduksi permainan "*Criss-Crosswords*" dengan janji memberi Butts royalti dari setiap unit yang terjual. Sebagai pemilik yang baru, Brunot mengganti nama permainan menjadi "*Scrabble*", sebuah kata dalam bahasa Inggris yang berarti "berjuang membanting tulang". Brunot hanya sedikit mengubah kotak-kotak "bonus" pada papan permainan, dan menyederhanakan aturan permainan. Permainan Scrabble ternyata laku dijual. Salah satu pembelinya adalah toko serba ada Macy's yang berhasil mengangkat Scrabble menjadi mainan yang ingin dibeli konsumen.^[12]

Di tahun 1953, Brunot menjual hak produksi permainan Scrabble kepada Selchow dan Righter, karena kapasitas produksi tidak mampu lagi memenuhi permintaan. Seperti halnya Parker Brothers dan Milton Bradley, Selchow dan Righter adalah salah satu dari pabrik mainan besar yang dulu pernah menolak untuk membeli permainan Scrabble. Di Britania Raya dan Australia, Scrabble mulai dipasarkan J. W. Spear & Sons tanggal 19 Januari 1955. Perusahaan J. W. Spear & Sons sekarang merupakan anak perusahaan dari Mattel. Di tahun 1986, Selchow and Righter menjual permainan ini kepada Coleco yang kemudian menjualnya lagi kepada Hasbro.^[12]



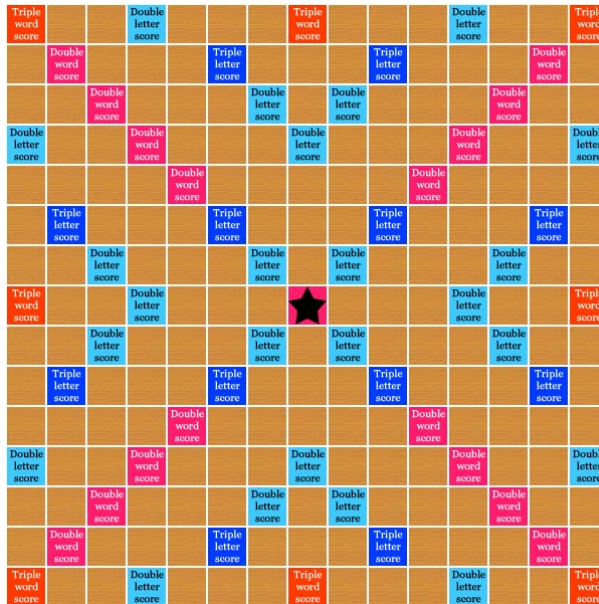
Gambar 2.1 Permainan Scrabble

2.1.2 Aturan Permainan

Permainan dimainkan dua sampai 4 orang, pada TA ini hanya akan ada 2 player, human player vs komputer player dan komputer player vs komputer player, player mengumpulkan nilai dengan meletakkan keping huruf pada papan permainan. Setiap huruf mempunyai point yang berbeda. Sehingga pemenang adalah player yang mendapatkan nilai tertinggi di akhir permainan.

Papan Scrabble

Papan permainan berukuran 15x15 kotak, dimana kotak-kotak ini yang digunakan sebagai tempat meletakkan keping huruf.



Gambar 2.2 Papan Permainan Scrabble

Nilai tambahan

Beberapa kotak dalam papan Scrabble mempunyai nilai tambah tersendiri sesuai dengan nama dan warnanya yaitu :

- **Double Letter Scores** - ketika keeping huruf yang kita letakkan diletakkan pada kotak ini maka nilai nilai keeping huruf dikalikan 2.
- **Triple Letter Score** - ketika keeping huruf yang kita letakkan diletakkan pada kotak ini maka nilai nilai keeping huruf dikalikan 3.
- **Double Word Score** - ketika keeping huruf yang kita letakkan diletakkan pada kotak ini maka nilai kata atau jumlah dari nilai keeping huruf yang membentuk kata dikalikan 2.
- **Triple Word Score** - ketika keeping huruf yang kita letakkan diletakkan pada kotak ini maka nilai kata atau jumlah dari nilai keeping huruf yang membentuk kata dikalikan 3.

- **One Single Use** – setiap kotak bonus hanya bisa dipakai satu kali, artinya nilai akan dikalikan hanya satu kali, jika suatu kata terbentuk dari salah satu huruf dari kata yg sudah terbentuk pada salah satu kotak bonus maka nilai tidak dikalikan.

Keping huruf Scrabble

Permainan dimainkan dengan 100 keping huruf, 98 huruf dari a sampai z dengan jumlah yang ditentukan dan 2 keping kosong adalah bonus atau blank, bisa digunakan untuk mengganti huruf apa saja, tetapi tidak mempunyai score.



Gambar 2.3 Huruf-huruf dalam Permainan Scrabble

Point masing-masing keping huruf

Point-point pada tiap huruf pada permainan Scrabble.

0 Points - Blank tile.

1 Point - A, E, I, L, N, O, R, S, T dan U.

2 Points - D dan G.

3 Points - B, C, M dan P.

4 Points - F, H, V, W dan Y.

5 Points - K.

8 Points - J dan X.

10 Points - Q dan Z

Bonus Bingo

Bonus bingo adalah bonus yang didapat pemain ketika pemain memainkan kata dengan ketujuh huruf yang diberikan, nilai bonus ini ditambahkan pada score kata yang dibentuk dengan nilai 50.

Memulai permainan

Tanpa melihat ke kantong penyimpanan keping huruf, setiap player mengambil 7 keping huruf, artinya keping huruf dibagikan kepada player secara acak.

Player bermain dengan cara membentuk kata dari tujuh huruf yang dibagikan dan meletakkannya dipapan, jika papan telah terdapat kata maka pemain membentuk kata dengan mengkaitkan huruf-huruf yang dipunyai dengan huruf-huruf yang terdapat dipapan untuk membentuk kata.

Permainan berakhir ketika semua atau kedua player tidak dapat lagi membentuk kata.

Kotak tengah yang bergambar bintang merupakan bonus satu kali giliran di awal, kata yang terbentuk mengenai kotak ini nilai nya akan di kalikan dua dan hanya berlaku satu kali.

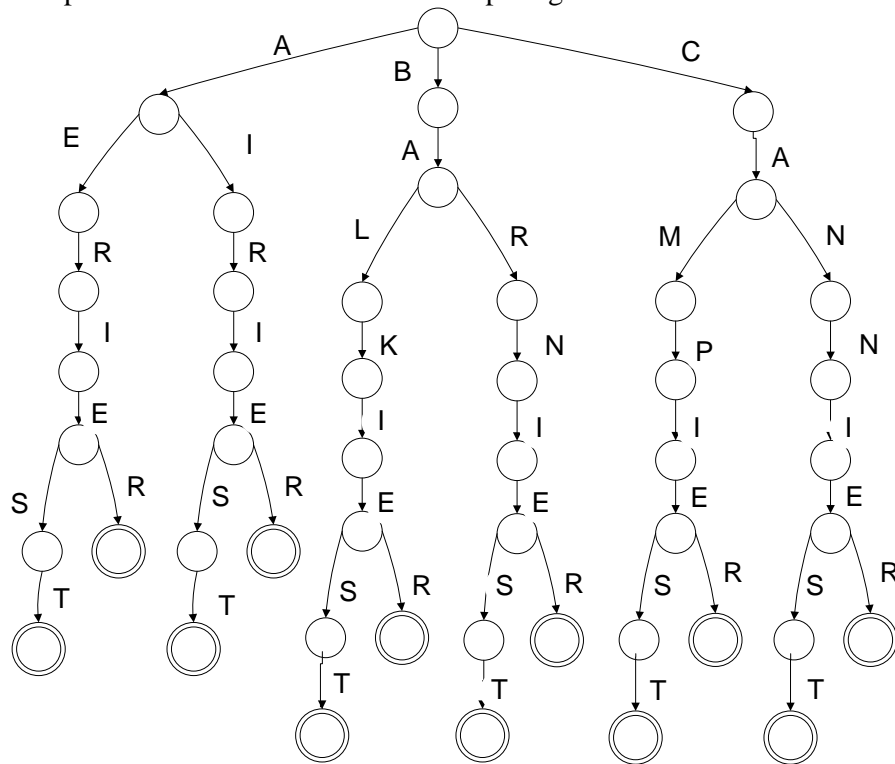


Gambar 2.4 Permainan Scrabble dimulai

2.2 Algoritma DAWG

Kata yang dapat dimainkan dalam permainan scrabble adalah semua kata yang ada didalam kamus. Sehingga untuk mendukung AI dalam program ini perlu dibangun sebuah prosedur untuk mencari kata yang akan dimainkan pada kamus, dengan asumsi bahwa komputer mempunyai semua perbendaharaan kata yang ada didalam kamus. Untuk melakukan pencarian kata dalam kamus perlu dibuat struktur data yang merepresentasikan semua kata yang ada di dalam kamus, secara sederhana dapat direpresentasikan dengan sebuah tree. Tree yang dibangun memiliki node akhir yang khusus ditandai sebagai tanda bahwa node tersebut merupakan akhir dari kata.

Seperti contoh misalkan dalam sebuah kamus terdapat beberapa kata yaitu ; *AERIER*, *AERIEST*, *AIRIER*, *AIRIEST*, *BALKIER*, *BALKIEST*, *BARNIER*, *BARNIEST*, *CAMPIER*, *CAMPIEST*, *CANNIER*, dan *CANNIEST*. Kata-kata ini dapat direpresentasikan dalam sebuah tree seperti gambar dibawah ini :



Gambar 2.5 Pohon huruf (trie)

Seperti yang terlihat pada gambar 2.5 diatas, terdapat banyak sekali huruf-huruf yang seharusnya dapat digunakan bersama. Jika struktur data pada pohon pada gambar 2.5 dapat diperkecil atau node yang sama dipakai bersama maka proses pencarian akan lebih cepat dan memori yang terpakai akan lebih kecil. Ide nya adalah dengan memakai bersama awalan dan akhiran yang sama.

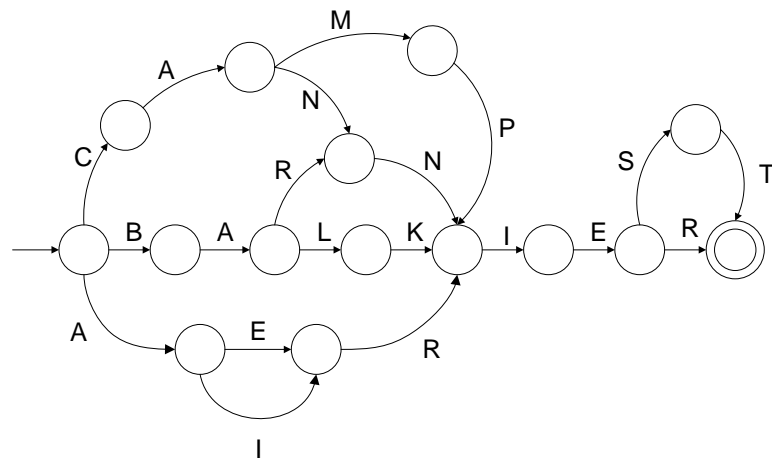
Struktur data yang merepresentasikan kata-kata dengan efisiensi ruang dengan menggunakan awalan dan akhiran yang sama bersama-sama disebut directed acyclic word graph^[2].

Directed acyclic word graph atau DAWG adalah sebuah struktur data yang menyimpan kosakata atau kata dimana dalam penggunaannya DAWG membutuhkan memory yang sangat kecil^[7], dimana setiap awalan dan akhiran digunakan bersama-sama untuk memperkecil jumlah ruang yang diperlukan untuk menyimpan kosakata^[2] sehingga memperkecil waktu yang diperlukan untuk meretrieve kata^[7]

Disebut *directed graph* karena hanya dapat berpindah ke arah tertentu antara dua node, dengan kata lain jika state bergerak dari A ke B maka state tidak dapat bergerak dari B ke A. Disebut *acyclic* karena data struktur ini tidak mempunyai *circle* atau sirkuit atau tidak mempunyai *cycle*, karena hal ini dapat menyebabkan looping tanpa henti pada program pencarian. Misal nya node berpindah dari A ke B kemudian ke C jika kembali lagi ke A akan membuat sebuah *cycle*.

Untuk lebih jelasnya *directed acyclic word graph* merupakan perubahan bentuk dari sebuah pohon huruf (seperti pada gambar 2.4) menjadi sebuah graph yang dibentuk seperti sebuah finite automata yang dapat mengenali kata dengan penggunaan state yang paling minimum^[2].

Dari contoh pada gambar 2.4 *directed acyclic word graph* dapat direpresentasikan seperti gambar berikut :



Gambar 2.6 A directed acyclic word graph

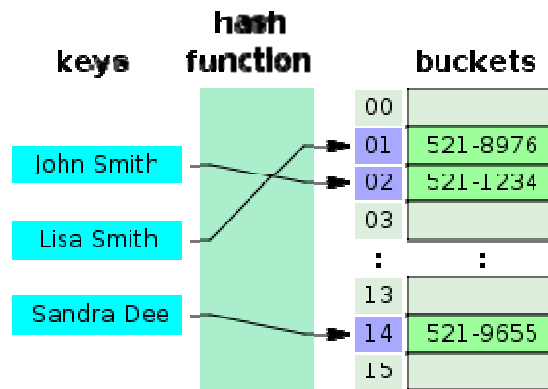
Setelah dilakukan pengurangan state atau node yang sama kemudian dipakai bersama-sama terlihat jumlah node atau state jauh berkurang lebih dari setengah jumlah node awal pada pohon huruf.

Pada contoh, representasi struktur data hanya menggunakan beberapa kata sehingga sangat mudah menggambarkan dan menemukan finite automata yang jumlah statenya paling minimum.

2.3 Struktur Penyimpanan Kamus Data pada Java

Hasil implementasi algoritma *DAWG* adalah berupa graph finite automata yang dapat menerima semua kata bahasa Inggris yang terdapat dalam data pengujian dalam laporan ini, dimana kata-kata yang dipakai hanya sebuah sampel data yang terdiri dari 178691 kata.

Dalam implementasinya sendiri semua node-node yang dibentuk oleh algoritma *DAWG* disimpan menggunakan metode yang telah disediakan oleh Java, dimana semua data yang berkaitan akan dimaping kedalam sebuah metode yaitu hashtable atau hash map. Dimana hash table adalah data struktur yang memanfaatkan fungsi hash dalam mengasosiasikan sebuah key dengan sebuah nilai, sehingga key menjadi index dari data array yang disimpan, dalam hal ini index berbeda dengan kunci tetapi untuk mencari index dihitung dari kunci dengan menggunakan fungsi hash. Karena mirip dengan sebuah array sehingga hashtable juga dikatakan associative array. Misal kan terdapat data nama dan nomor telpon, maka hashtable akan menjadi seperti gambar dibawah ini.

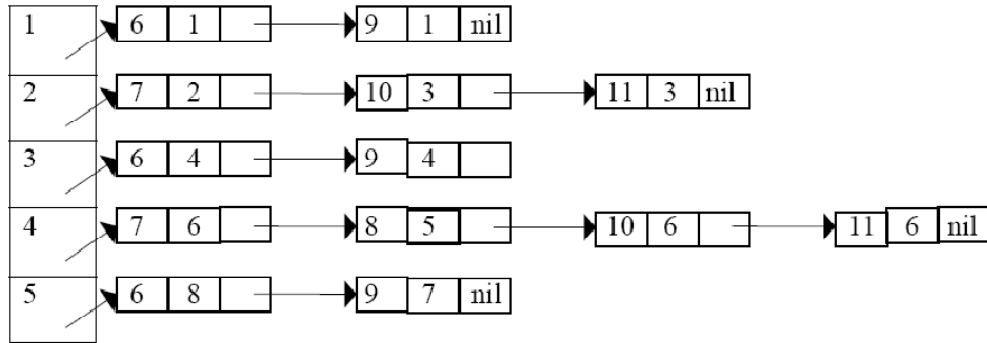


Gambar 2.7 Hashtable

Sedangkan dalam penyimpanan kamus data pada laporan ini yang menjadi index adalah semua node mulai dari root sampai leaf simpul terakhir yang akan diurut mulai dari simpul paling kiri ke simpul paling kanan. Dan huruf-huruf yang terdapat pada edge masing-masing node akan disimpan sebagai nilai yang ditunjuk oleh index yang berupa node itu sendiri. Setiap huruf disimpan dalam bentuk integer dimana a direpresentasikan sebagai 1 dan z sebagai 26 dengan nilai biner.

Nilai array tiap index akan ditempati oleh huruf yang terdapat pada edge yang meninggalkan node tersebut, node pertama yang meninggalkan node tersebut akan disimpan terlebih dahulu diikuti edge selanjutnya berurutan dari simpul paling kiri. Hal ini mungkin terjadi karena setiap lokasi array adalah suatu list berantai yang berisi pasangan kunci dan nilai, jika suatu nilai mempunyai kode hash yang sama maka nilai tersebut akan menempati list yang sama.

Seperti contoh pada gambar dibawah ini.



Gambar 2.8 Array List Berantai

3. Analisis dan Perancangan Sistem

3.1 Analisis Kebutuhan

Untuk membangun game *scrabble* ini perlu dianalisis kebutuhan yang harus dipenuhi untuk merancang dan membangun aplikasi agar dapat dibangun aplikasi game *scrabble* dengan strategi yang baik dan sesuai dengan tujuan yang diharapkan. Pada permainan ini terdapat tiga macam tingkatan level yang dapat dipilih oleh pemain yaitu; easy, medium, dan hard. Tingkatan level adalah tingkatan kesulitan pemain dalam mengalahkan strategi permainan *scrabble* yang dibangun.

3.1.1 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Dalam membangun aplikasi permainan *scrabble* dibutuhkan perangkat lunak seperti software editor dan system operasi untuk menjalankan software tersebut, karena bahasa yang digunakan dalam pengkodean adalah bahasa java maka pembuatan aplikasi permainan *scrabble* ini menggunakan editor Netbean 6 dan system operasi Windows xp.

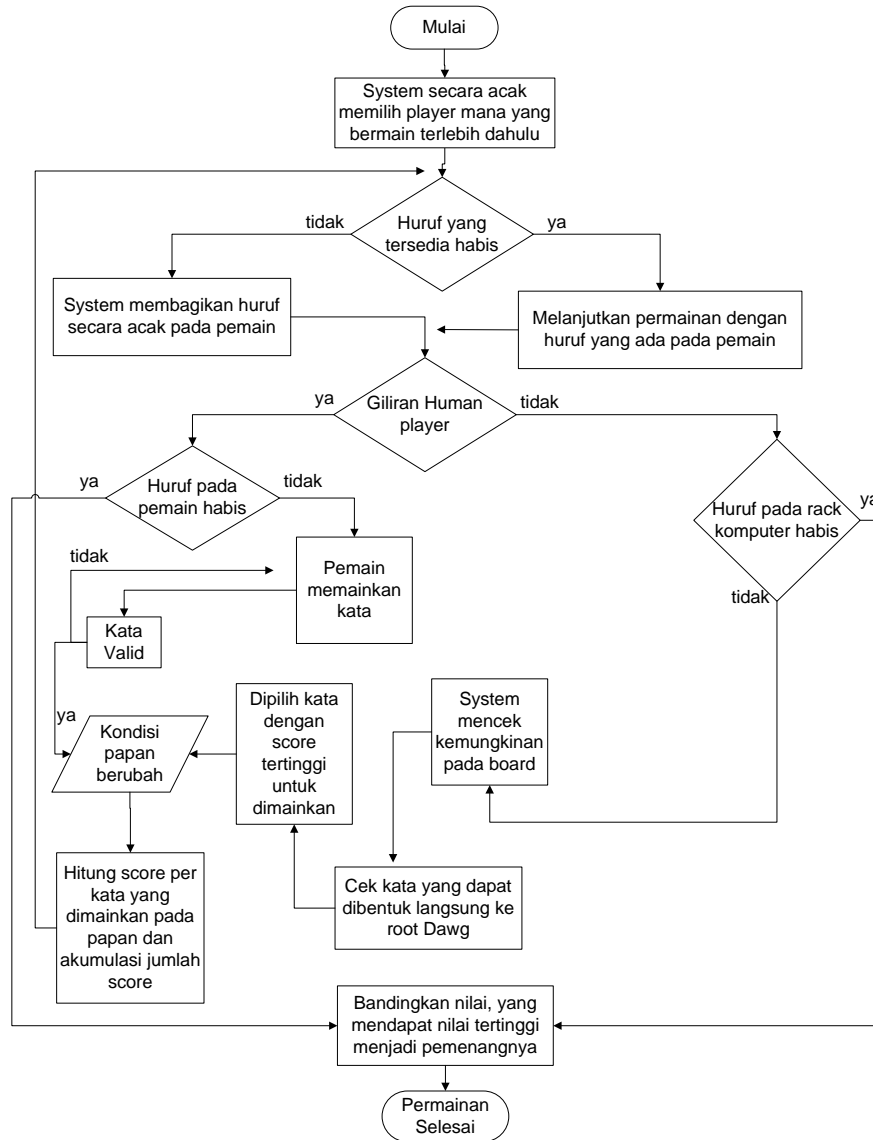
3.1.2 Analisis Kebutuhan Fungsional

Permainan *scrabble* mempunyai beberapa kebutuhan fungsionalitas dan juga antarmuka yaitu form atau tampilan antarmuka sebagai interface dengan pemain yang akan menjadi visualisasi dari permainan. Sedangkan fungsionalitas system meliputi :

1. Papan permainan berukuran 15x15 kotak dan terdapat bonus di kotak-kotak berwarna dengan aturan tertentu.
2. Terdapat beberapa strategi permainan.
3. Ditampilkan score masing-masing yang akan menjadi penentu pemenang dari permainan.
4. Kamus data yang dipakai dalam permainan merupakan kamus data yang dibentuk menggunakan algoritma *directed acyclic word graph* dari file list kata-kata bahasa inggris.

3.2 Perancangan Sistem

Berikut adalah alur permainan scrabble sesuai dengan aturan permainannya yang digambarkan pada diagram flowchart dibawah ini :



Gambar 3.1 Diagram Flowchart Game Scrabble.

Untuk lebih jelasnya berikut rincian penjelasan dari diagram blok diatas beserta masing-masing fungsionalitasnya :

- 1) System secara acak memilih player mana yang akan bermain duluan.
- 2) Pada awal permainan system membagikan 7 huruf pada pemain secara acak, jika huruf yang tersedia habis maka permainan akan dilanjutkan dengan memainkan huruf yang ada ditangan pemain saja, jika pemain tidak memiliki huruf ditangan maka permainan selesai. Permainan akan berakhir jika kedua pemain tidak dapat lagi membentuk kata.

- 3) Pada saat pemain memainkan huruf dan meletakkannya dipapan system akan mengecek apakah huruf yang dimainkan valid atau terdapat didalam kosakata atau tidak. Pada tahap inilah kamus data yang dibentuk berperan dimana pengecekan dilakukan pada kamus data.
- 4) Kata yang valid akan dicek jumlah nilainya sesuai dengan nilai tiap huruf dan nilai bonus dipapan.
- 5) Pemain harus bermain dengan kata yang valid untuk melanjutkan permainan.
- 6) Pemenang ditentukan berdasarkan nilai yang didapat, pada akhir permainan akan dibandingkan dua nilai pemain, pemain dengan nilai tertinggi menjadi pemenang.

Permainan scrabble didukung oleh kamus data yang memuat kata-kata yang akan digunakan dalam menentukan apakah kata yang dimainkan valid atau tidak atautkah kata yang dimainkan pemain terdapat didalam kamus atau tidak, karena kata yang dapat dimainkan dalam permainan hanya kata yang ada dalam kamus. Untuk itu dibentuk kamus data menggunakan algoritma *Directed Acyclic Word Graph* sebelum dibentuk nya aplikasi permainan.

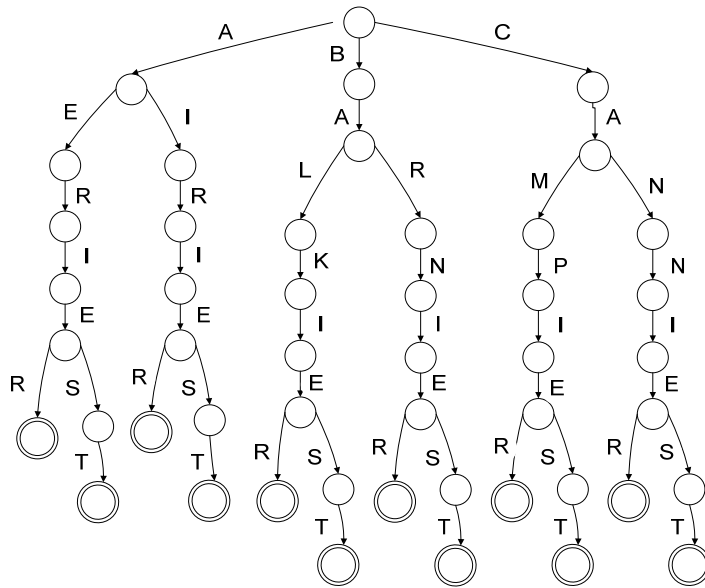
3.3 Algoritma Directed Acyclic Word Graph

Proses pertama yang harus dibangun sebelum membangun aplikasi game scrabble ini adalah membangun kamus data, dimana dalam laporan ini pembangunan kamus data memanfaatkan algoritma *Directed Acyclic Word Graph*. Algoritma *Directed Acyclic Word Graph* merupakan konversi pembentukan pohon data atau pohon huruf menjadi bentuk finite automata yang dapat menerima semua kata dengan jumlah state paling minimum yang merepresentasikan sebuah kosakata atau kata-kata yang terdapat didalam kamus yang dipakai.

Konversi pohon data atau pohon huruf menjadi sebuah finite automata dapat digambarkan oleh proses berikut.

Misalkan diberikan contoh dimana terdapat kata-kata yang terdiri dari *AERIER*, *AERIEST*, *AIRIER*, *AIRIEST*, *BALKIER*, *BALKIEST*, *BARNIER*, *BARNIEST*, *CAMPIER*, *CAMPIEST*, *CANNIER*, dan *CANNIEST* yang disimpan dalam sebuah data dalam file berekstensi TXT. Kemudian isi file dibaca satu persatu tiap huruf tiap kata.

Huruf pertama dari kata menjadi node awal dan huruf kedua menjadi node setelah node awal begitu seterusnya sampai akhir kata menjadi node akhir, node akhir yang menjadi akhir kata ditandai sebagai end of word. Untuk mempermudah proses pembentukan tree. Kemudian setelah semua kata dalam file dibaca dan dibentuk tree akan terbentuk pohon kata seperti gambar dibawah berikut :

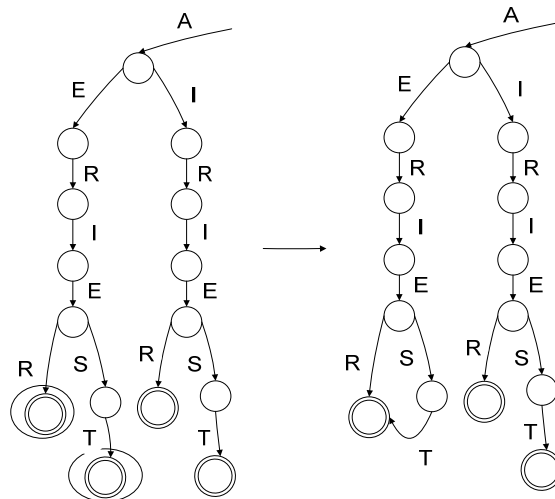


Gambar 3.2 Pohon Huruf atau Kata

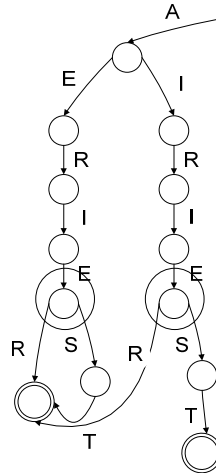
Proses selanjutnya adalah menandai node-node yang identik, dimana node yang dikatakan identik meliputi beberapa kriteria berikut :

- Node memiliki edge huruf yang sama
- Node memiliki jumlah anak yang sama dan node anak mempunyai edge huruf yang sama atau sama-sama end of word atau leaf
- Node mempunyai level yang sama.

Node-node yang identik dijadikan satu dan edge-edge yang menuju node tersebut akan dialihkan hanya pada satu node. Node dicek satu persatu mulai dari simpul paling kiri dan paling akhir atau leaf, leaf simpul pertama dibandingkan dengan simpul kedua. Jika node tidak identik maka node pembanding akan dilanjutkan dibandingkan dengan simpul berikut.

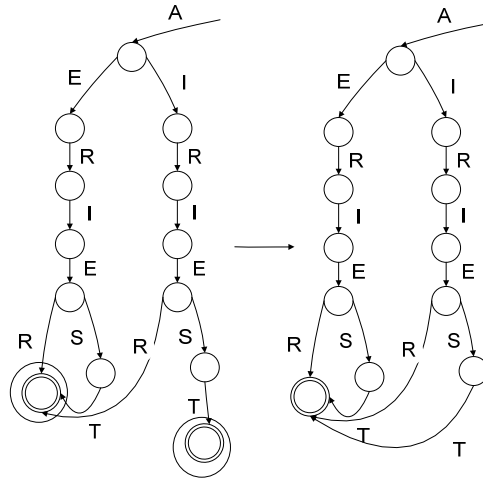


Gambar 3.3 Proses algoritma *dawg* langkah 1



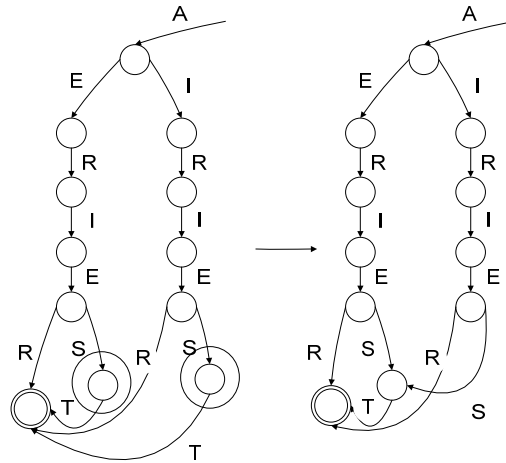
Gambar 3.6 Proses algoritma *dawg* langkah 4

Karena node yang ditunjuk oleh edge E mempunyai node anak yang belum dikunjungi maka untuk mencegah hilangnya node anak sebelum node tersebut dibandingkan terlebih dahulu node anak dibandingkan mulai dari leaf. Node leaf pada simpul awal kemudian dibandingkan dengan node leaf pada simpul node anak yang belum dikunjungi tersebut.



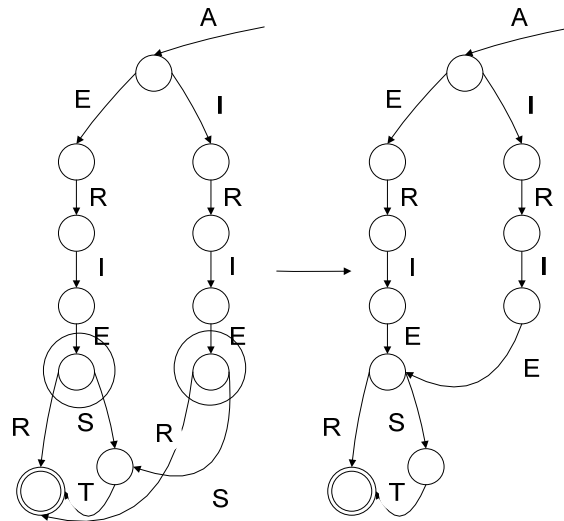
Gambar 3.7 Proses algoritma *dawg* langkah 5

Karena kedua node leaf tersebut identik dimana mereka sama-sama end of word maka node kedua dihapus dan edge T yang menunjuk ke node tersebut dialihkan ke node leaf pertama. Selanjutnya proses matching dilanjutkan ke node dengan kedalaman n-1 pada masing-masing simpul.



Gambar 3.8 Proses algoritma *dawg* langkah 6

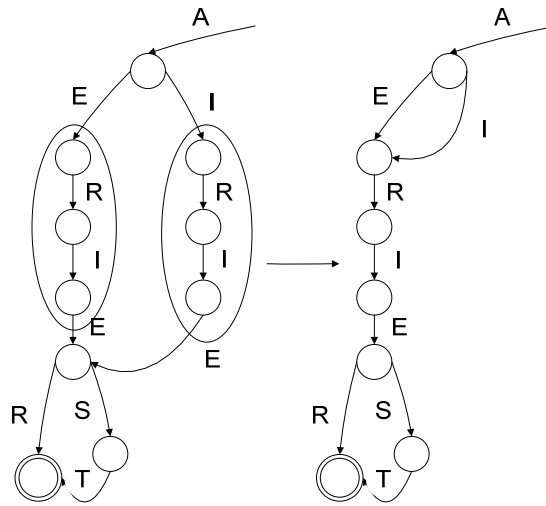
Antara node pembanding dan node yang dibandingkan terlihat bahwa keduanya merupakan node yang identik yaitu sama-sama mempunyai satu anak dan edge T, sehingga kedua node ini dikatakan identik. Selanjutnya dilakukan proses seperti sebelumnya dimana node yang dibandingkan dihapus dan edge pada node sebelumnya dialihkan menuju node pembanding. Proses matching kemudian diteruskan ke node dengan kedalaman level n-1 atau node diatas berikutnya.



Gambar 3.9 Proses algoritma *dawg* langkah 7

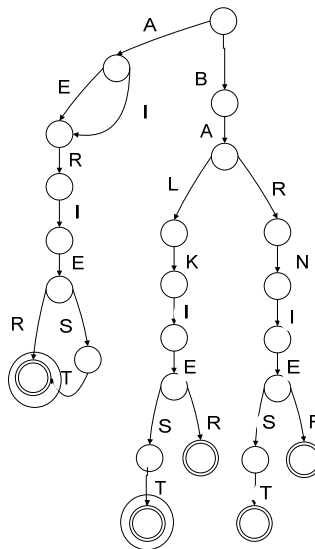
Pada gambar diatas dibandingkan node pertama dan kedua terlihat identik dimana kedua node mempunyai jumlah anak dan edge-edge yang sama sehingga dilakukan proses penghapusan node kedua dan edge E simpul kedua dialihkan ke node simpul pertama. Selanjutnya proses yang sama seperti node-node sebelumnya dilakukan pada node-node dengan edge I dan R dimana bisa terlihat

pada gambar node-node ini identik sehingga node-node ini langsung dapat digabung.



Gambar 3.10 Proses algoritma *dawg* langkah 8

Karena pada kedua simpul ini telah digabung maka proses matching berlanjut ke simpul berikut dimana dibandingkan node leaf simpul awal dengan node leaf simpul berikut yang belum dikunjungi.



Gambar 3.11 Proses algoritma *dawg* langkah 9

Begitu seterusnya sampai kata terakhir, jika proses matching telah dilakukan sampai kata terakhir maka node pembanding akan diteruskan ke simpul berikutnya.

3.4 Implementasi Algoritma *DAWG*

Algoritma *DAWG* berperan dalam membantu system menggenerate langkah dengan cara membangun struktur data dari semua kata yang terdapat dalam kamus berbentuk finite automata dengan jumlah state yang minimal sehingga mempermudah dalam proses searching pada struktur data dan menampilkan kata yang dicari.

Berikut proses system dalam memilih langkah yang dibantu oleh *DAWG* :

- 1) Jika pada papan permainan belum terdapat kata maka system langsung melakukan searching kata mulai dari root pada kamus data dawg dari 7 kata acak yang didapat sehingga semua kata yang didapat merupakan kata yang valid.
- 2) Kemudian dipilih kata dengan nilai tertinggi untuk dimainkan dan diletakkan pada papan permainan.
- 3) Jika pada papan telah terdapat kata, maka system melakukan pencarian pada papan huruf-huruf yang mungkin untuk digabung dengan 7 huruf acak pada pemain dilihat dari posisinya.
- 4) Pencarian pada papan langsung dilakukan searching pada kamus data dawg.
- 5) Pencarian dipisahkan antara kata yang terbentuk secara horizontal dan vertikal.
- 6) Setiap kata yang terbentuk dibandingkan dengan kata yang terbentuk berikutnya dan kata dengan score tertinggi dipilih untuk dimainkan, begitu juga kata yang terbentuk secara vertical dan horizontal dibandingkan score yang dan dipilih score yang tertinggi untuk dimainkan pada papan.
- 7) Kata yang dipilih untuk dimainkan kemudian ditampilkan pada papan.

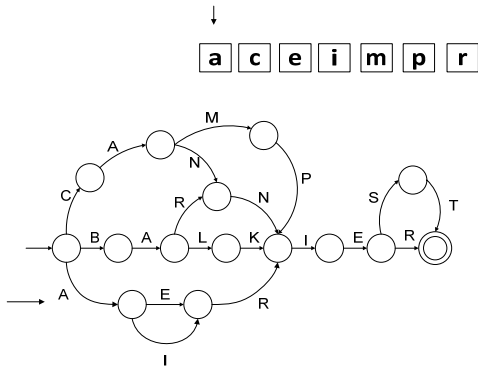
Proses searching dalam meretrieve kata dimulai pada root awal kamus data, dengan cara matching antara huruf yang terdapat pada rack dengan edge-edge yang menyimpan huruf-huruf pada kamus data. Kata diretrieve ketika pencarian berakhir pada leaf yang ditandai dengan end of word, proses search dan matching yang berakhir pada leaf yang bertandakan end of word merupakan kata yang dapat dimainkan. Untuk lebih jelasnya bisa digambarkan oleh ilustrasi berikut :

Misalkan player computer mempunyai rack : campier

Dan kamus data yang dipakai adalah contoh kamus data pada bab 2.

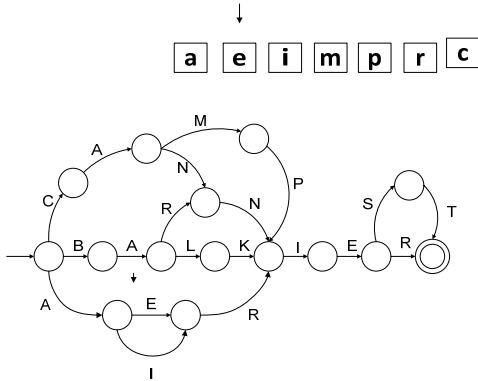
Untuk mempermudah dalam implementasi coding rack akan disusun secara abjad yaitu aceimpr. Kemudian :

- Huruf a akan dicek didalam root apakah terdapat pada kamus data
- Jika huruf a cocok dengan salah satu edge atau huruf yang disimpan dalam kamus data pada root maka a akan menjadi huruf awal dalam membentuk kata.



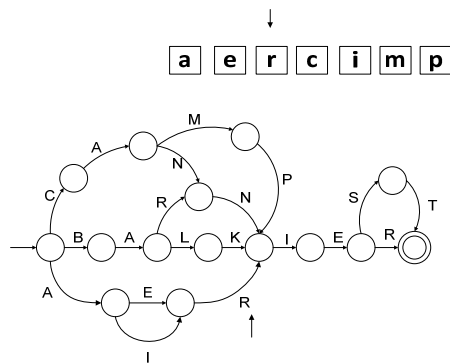
Gambar 3.12 Proses Pencarian Kemungkinan Kata langkah 1

- Setelah itu huruf c dicek apakah huruf a mempunyai node anak yang menyimpan huruf c. Karena setelah node yang ditunjuk oleh edge a tidak terdapat huruf c maka proses matching akan diteruskan ke huruf setelah c yaitu e dan huruf c digeser kebelakang.



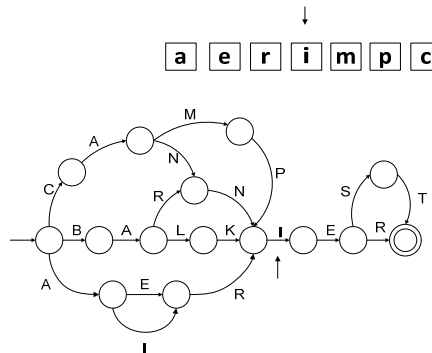
Gambar 3.13 Proses Pencarian Kemungkinan Kata langkah 2

- Karena terdapat edge e setelah node yang ditunjuk oleh edge a maka dicek huruf selanjutnya yaitu i. Begitu seterusnya dimana setelah edge e terlihat pada kamus data hanya ada edge r sehingga akan terlihat seperti ilustrasi dibawah.



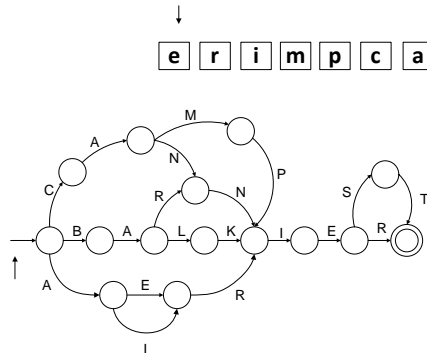
Gambar 3.14 Proses Pencarian Kemungkinan Kata langkah 3

- Setelah itu dicek apakah terdapat edge c setelah node yang ditunjuk edge r, terlihat dari gambar bahwa setelah node yang ditunjuk edge r hanya ada edge i sehingga akan terlihat seperti ilustrasi dibawah ini.



Gambar 3.15 Proses Pencarian Kemungkinan Kata langkah 4

- Setelah itu dicek apakah setelah node yang ditunjuk oleh edge I terdapat edge m, p dan c. Karena tidak terdapat huruf m, p dan c dan pencarian tidak berakhir pada node end of word maka huruf yang sudah tersusun tidak bisa dikatakan sebuah kata atau tidak valid.
- Maka pencarian dilanjutkan kembali ke huruf awal, karena huruf a sudah dicek sampai simpul bawah dan tidak berhenti pada leaf maka pencarian dimulai dari huruf selanjutnya yaitu e.



Gambar 3.16 Proses Pencarian Kemungkinan Kata langkah 5

- Karena tidak terdapat huruf e pada root. Jika dilihat pada gambar diatas huruf yang terdapat pada root selain a yaitu c, matching akan terus bergeser dari huruf e sampai huruf c. Saat huruf c dicek dan terdapat pada root maka pencarian pada kamus data pun dimulai dari huruf c.