**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

1. **IDENTITAS PENGUSUL**

Nama : Yolanda Puspita

NRP : 5108.100.114

Dosen Wali : Imam Kuswardayan, S.Kom, M.T.

1. **JUDUL TUGAS AKHIR**

“Rancang Bangun Perangkat Lunak Permainan Acak Kata dengan Metode *Backtracking* dan Metode *Stemming* Porter pada Android ”

1. **LATAR BELAKANG**

*Game* adalah fenomena global. *Game* merupakan sebuah permainan elektronik yang bertujuan untuk memberikan hiburan kepada pemainnya. Peminat *game* tidak hanya anak kecil tetapi semua umur. Sejarah perkembangan *game* berlangsung selama lima dekade, namun *game* sendiri tidak populer sampai dengan akhir tahun 1970-an dan terus mengalami perkembangan sampai saat ini.

Secara umum *game* memiliki beberapa jenis (*genre*). Beberapa *genre* *game* antara lain : permainan *maze*, permainan papan, permainan teka teki, permainan pertarungan, permainan balap, permainan strategi *real time,* permainan simulasi, permainan edukasi, permainan petualangan, dll.

Pada sebuah *game* bisa terdapat sebuah *genre* murni maupun *genre* campuran yang merupakan gabungan dari beberapa *genre*. *Genre* pada sebuah *game* menentukan tujuan dari *game* itu sendiri yang tentunya berbeda – beda di setiap *genre-*nya.

Pada perkembangannya saat ini, permainan elektronik dibangun menggunakan beberapa *platform*. *Platform* yang digunakan yaitu *Personal Computer* (operasi sistem yang digunakan adalah Windows, Linux, Dos, dan Mac), *Hand-held* (seperti *Gameboy* dan PSP), *Mobile device* (telepon genggam (terutama untuk sistem operasi Android, iOS dan Windows *Phone* 7)), maupun konsol (seperti *PlayStation,* x-*box*, Nintendo Wii, *PlayStation Move*, dan Kinect). Bisnis *game* dewasa ini sudah merambah dimana – mana. Bahkan perusahaan IT di Indonesia pun banyak yang mulai mengarah pada produksi pembuatan *game*. Namun sayangnya masih sedikit sekali *game* yang mengandung *genre* edukasi maupun yang memiliki konten – konten pendidikan di dalamnya. Tanpa disadari sebenarnya *game* edukasi dapat memberikan pengajaran yang bermanfaat terhadap pemainnya.

*Game* edukasi bertujuan untuk menumbuhkan motivasi belajar dan rasa keingintahuan dari pemainnya. *Game* edukasi akan sangat bermanfaat bila konten – konten yang terdapat di dalamnya adalah konten – konten yang mengandung ilmu pengetahuan. Pemain *game* edukasi merasa tidak terbebani ketika dihadapkan dengan suatu pembelajaran yang ada, karena memang lebih menyenangkan belajar sambil bermain.

Dengan latar belakang masalah – masalah di atas maka pada tugas akhir ini akan dibangun sebuah perangkat lunak permainan kecerdasan buatan, yaitu permainan acak kata yang dibangun pada *platform mobile device* untuk sistem operasi Androiddan merupakan permainan dengan *genre* campuran, yaitu permainan edukasi, permainan papan, dan permainan teka teki. Permainan ini dibuat untuk menambah pengetahuan dan memberikan motivasi kepada pemainnya dalam belajar dan memperbanyak kosakata bahasa inggris, sains dalam bahasa inggris, dan sinonim dalam bahasa inggris mengingat masih rendahnya motivasi dan minat rata – rata penduduk di Indonesia khususnya pada usia sekolah dalam belajar bahasa inggris, sedangkan bahasa inggris itu sendiri adalah bahasa internasional yang seharusnya wajib dipelajari dan dikuasai.

Permainan ini dibuat pada *platform mobile device* yang bersistem operasi Android karena seperti yang kita ketahui sistem operasi ini *open source* dan peminat *smart phone* dengan sistem operasi ini jumlahnya paling tinggi dibandingkan dengan peminat *smart phone* dengan sistem operasi lain dikarenakan dari segi harga dan efisiensi. Selain itu *game* pada *mobile device* lebih praktis untuk digunakan.

Dengan semakin berkembangnya *game*, *platform,* dan sistem operasi yang ada maka tugas akhir ini dibangun dengan menggunakan *platform* pada *smart phone* bersistem operasi Android yang sedang banyak diminati. Dengan kemudahan yang disajikan perangkat lunak dan media yang berperan maka perangkat lunak ini diharapkan dapat berguna dalam proses pembelajaran dan dapat meningkatkan motivasi belajar bahasa inggris.

1. **TUJUAN TUGAS AKHIR**

Tujuan dari pembuatan tugas akhir ini adalah :

1. Menciptakan aplikasi permainan gabungan dari permainan papan dan permainan teka teki.
2. Meningkatkan motivasi dan minat pengguna dalam belajar dan memperbanyak kosakata bahasa inggris, sains dalam bahasa inggris, dan sinonim dalam bahasa inggris.
3. Menciptakan sarana belajar kosakata bahasa inggris, sains dalam bahasa inggris, dan sinonim dalam bahasa inggris dengan cara yang praktis karena menggunakan media *smart phone* dengan sistem operasi Android yang sedang diminati sekarang ini.

1. **RUMUSAN PERMASALAHAN**

Permasalahan yang dibahas dalam tugas akhir ini meliputi :

1. Menetapkan aturan permainan, yaitu pemain harus menyusun huruf *random* menjadi sebuah kosakata bahasa inggris dan menjawab sinonim kosakata untuk nilai tambahan.
2. Menetapkan desain level, dimana setiap level mempunyai sistem kecerdasan buatan yang berbeda.
3. Bagaimana mencari solusi untuk menemukan kata yang tepat yang terbentuk dari huruf – huruf *random* pada rak dengan nilai yang paling maksimal dengan menggunakan metode *backtracking*.
4. Bagaimana mencocokkan kata yang dibentuk oleh pemain pada papan dengan kata pada basis data dengan menggunakan metode brute force pencocokan *string*.
5. Bagaimana mencari kata dasar dari kata yang dibentuk oleh pemain pada papan dengan menggunakan metode *stemming* Porter yang kemudian digunakan untuk mencari sinonim dari kata tersebut.
6. **BATASAN MASALAH**

Batasan permasalahan dalam tugas akhir ini adalah sebagai berikut:

1. Kata – kata yang digunakan adalah daftar kata – kata yang terdapat pada kamus Scrabble berbahasa inggris.
2. Kata – kata yang akan dicari kata dasarnya adalah kata – kata yang dapat merepresentasikan metode *Stemming* Porter untuk bahasa inggris.
3. Pertanyaan untuk permainan acak kata adalah pertanyaan seputar sains dalam bahasa inggris.
4. Data *sampling* pertanyaan yang digunakan untuk uji coba adalah pertanyaan seputar sains dalam bahasa inggris.

1. **RINGKASAN TUGAS AKHIR**

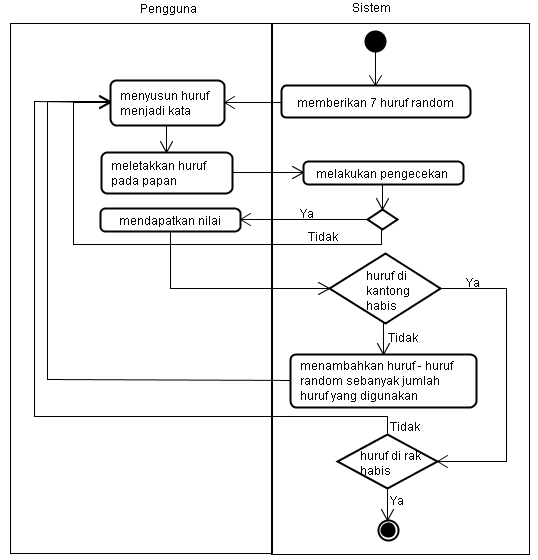
Pada tugas akhir ini akan dibangun sebuah perangkat lunak *game* kecerdasan buatandengan *genre* campuran dari permainan edukasi, permainan papan, dan permainan teka teki.

Pada aplikasi permainan ini, terdapat 2 pilihan permainan, yaitu permainan acak kata, dan permainan teka teki kata. Pada kedua pilihan permainan ini, pemain bermain melawan komputer. Untuk *leveling*, permainan acak kata tidak mempunyai level, sedangkan permainan teka teki kata mempunyai 3 level, yaitu *easy, normal,* dan *hard*. Dimana setiap pilihan permainan mempunyai aturan dan sistem yang berbeda.

Terdapat sebuah kantong yang berisikan sejumlah huruf (di setiap level berbeda jumlahnya), yang merupakan huruf A – Z dengan masing – masing jumlah yang berbeda dan memiliki poin yang berbeda pula yang disesuaikan dengan kecenderungan kemunculan sebuah huruf pada suatu kata dalam bahasa inggris. Huruf akan diberikan kepada pemain dan komputer secara acak. Pemain dan komputer harus menghabiskan huruf – huruf di kantong untuk menyelesaikan permainan dan mendapatkan pemenang.

Terdapat papan permainan untuk meletakkan kata. Papan permainan terdiri dari 15 baris dan 15 kolom yang di tandai dengan angka 1 – 15 untuk kotak yang melintang secara horizontal dan huruf A – O untuk kotak yang melintang secara vertikal. Terdapat kotak - kotak yang memberikan nilai lebih terhadap huruf yang diletakkan diatasnya, antara lain : *double letter* yang menggandakan nilai sebuah huruf yang diletakkan diatasnya, *triple letter* yang membuat nilai sebuah huruf yang diletkkan diatasnya menjadi tiga kali lipat, *double word* yang menggandakan nilai kata yang terbentuk apabila salah satu huruf yang membentuk kata diletakkan diatasnya, dan *triple word* yang mengalikan tiga nilai kata yang terbentuk apabila salah satu huruf yang membentuk kata diletakkan diatasnya.

Untuk permainan acak kata, tugas pemain dan komputer adalah menyusun dua atau lebih huruf dari 7 buah huruf *random* yang diberikan menjadi sebuah kata yang mungkin dalam bahasa inggris dan diletakkan di papan. Karena pemain selalu mendapat giliran pertama, pemain harus meletakkan salah satu huruf yang membentuk kata pertama tepat di tengah papan yang ditandai dengan kotak bergambar bintang. Kata dapat diletakkan secara horizontal maupun vertikal. Dan kemudian dilanjutkan dengan giliran bermain selanjutnya, yaitu komputer. Unutk lebih jelasnya berikut adalah *activity* diagram untuk permainan acak kata.



Gambar 1. *Activity* diagrampermainan acak kata

Setiap kata yang ditambahkan pada papan harus menyentuh kata – kata yang terbentuk sebelumnya. Setiap kata yang terbentuk pada tiap gliran bermain akan diperhitungkan nilainya. Pada setiap kata yang dibentuk akan diakukan pengecekan terhadap huruf – huruf yang bersebelahan baik vertikal maupun horizontal, apakah huruf – huruf tersebut cocok dengan daftar kata di basis data atau tidak. Untuk lebih jelasnya lihat gambar di bawah ini.



Gambar 2. Contoh Peletakan Kata yang Salah

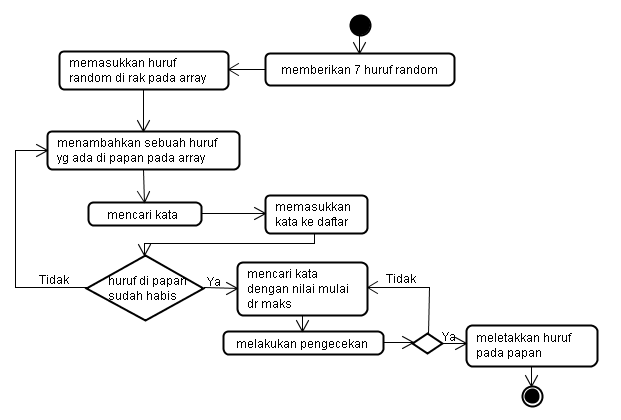
Gambar di atas menunjukkan 2 kata yang terbentuk, yaitu NONE dan AND. Misal kata pertama yang berada di papan adalah NONE. Pada giliran selanjutnya, dibentuk kata AND yang meletakkan huruf N tepat bersebelahan dengan huruf N dari NONE secara horizontal, dan meletakkan huruf D tepat bersebelahan dengan huruf O dari NONE secara horizontal. Maka dilakukan pengecekan apakah AND, NN, dan DO adalah kata yang ada di basis data (kosakata bahasa inggris) atau tidak. AND dan DO adalah kosakata bahasa inggris, sementara NN bukanlah kosakata bahasa inggris. Maka kata AND tidak dibenarkan diletakkan pada papan dengan susunan seperti gambar 1. Untuk peletakan kata yang benar lihat gambar berikut ini.



Gambar 3. Contoh Peletakan Kata yang Benar

Misal pada gambar di atas kata pertama yang berada di papan adalah NONE. Pada giliran selanjutnya, dibentuk kata AND yang meletakkan huruf A tepat bersebelahan dengan huruf N NONE secara horizontal, dan meletakkan huruf D tepat bersebelahan dengan huruf N NONE secara horizontal. Maka kemudian dilakukan pengecekan terhadap AND apakah merupakan kosakata bahasa inggris. AND adalah kosakata bahasa inggris, maka kata AND dibenarkan diletakkan pada papan.

Berikut adalah *activity* diagram sistem untuk kecerdasan buatan pada permainan acak kata ini.



Gambar 4. Kecerdasan buatan pada permainan acak kata

Sistem menambahkan semua huruf yang terdapat dalam papan ke dalam *array* kemudian sistem mencari kata – kata yang mungkin terbentuk. Setelah semua daftar kata terbentuk, dicari kata dengan nilai yang paling tinggi kemudian dilakukan pengecekan. Jika ternyata kata tersebut bukanlah kata yang sesuai maka sistem akan melakukan pengecekan pada kata dengan nilai persis dibawah kata yang dicek sebelumnya, dan seterusnya sampai dengan sistem menemukan kata yang sesuai.

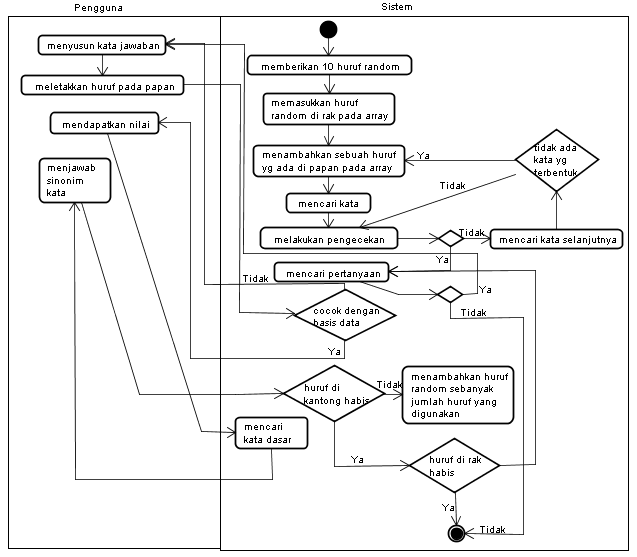
Terdapat tombol *hint* yang berguna untuk memberikan usulan kata untuk dimainkan pemain pada gilirannya. Pemain hanya mempunyai kesempatan 3 kali menggunakan tombol ini. Terdapat pula tombol *exchange* untuk mengganti huruf yang diberikan dengan huruf yang ada di kantong.

Permainan akan selesai pada saat huruf di kantong dan rak sudah habis, dan pemain dengan nilai akhir yang paling tinggi lah yang memenangkan permainan. Atau permainan bisa berhenti apabila kedua pemain (pengguna dan komputer) sudah *pass* sebanyak 5 kali dalam satu kali permainan. Dan apabila masih ada huruf yang tersisa pada saat permainan sudah selesai, maka nilai akhirnya adalah nilai yang sudah dikumpulkan pemain dikurangi dengan jumlah dari nilai huruf – huruf yang tersisa.

Untuk permainan teka teki kata, tugas pemain dan komputer adalah menyusun dua atau lebih huruf dari 10 buah huruf *random* menjadi sebuah kata yang merupakan jawaban dari pertanyaan. Pada permainan teka teki kata, terdapat 100 buah huruf di kantong untuk level *easy*, 130 buah huruf untuk level *normal,* dan 160 buah huruf untuk level *hard*. Isi kantong merupakan huruf A – Z dengan jumlah dan nilai yang berbeda tiap hurufnya. Yang membedakan pelevelannya adalah kecerdasan buatan pada system serta tingkat kesulitan soal.

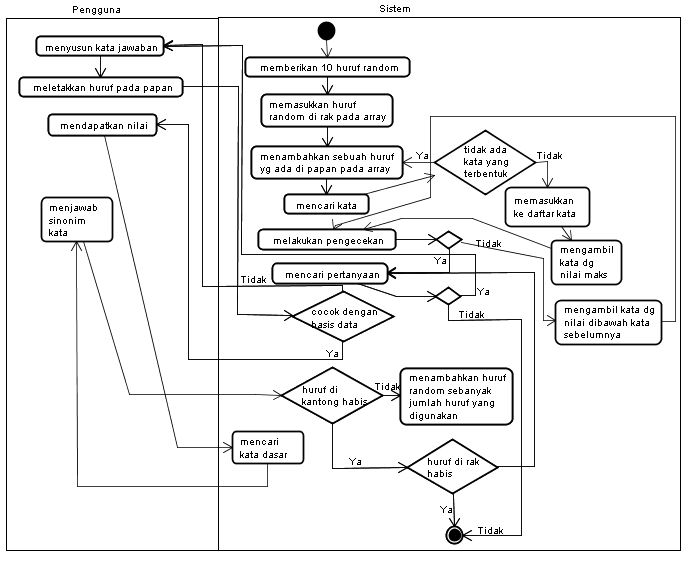
Untuk giliran bermain komputer, sistem melakukan hal yang sama seperti pada giliran pengguna bermain, hanya saja di sini ada sedikit perbedaan, yaitu sistem tidak mengeluarkan pertanyaan, melainkan komputer langsung menyusun kata yang sesuai pada papan.

Sistem permainan pada level *easy* adalah sebagai berikut. Sistem menambahkan huruf yang ada di papan pada *array*. Sistem mencari kata yang mungkin terbentuk, kata pertama yang mungkin terbentuk langsung dilakukan pengecekan. Jika kata merupakan kata yang sesuai maka sistem mencari pertanyaan yang jawabannya merupakan kata tersebut dan mengeluarkan pertanyaan. Jika ternyata kata tersebut bukan merupakan kata yang sesuai maka sistem akan mencari kata berikutnya sampai dengan menemukan kata yang sesuai. Untuk lebih jelasnya bisa dilihat pada gambar berikut.



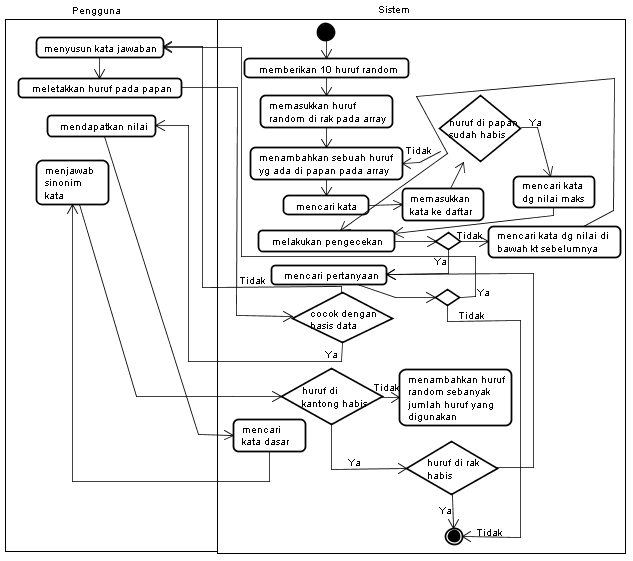
Gambar 5. *Activity* diagram permainan teka teki kata level *easy*

Sistem permainan pada level *normal* adalah sebagai berikut. Sistem menambahkan huruf yang ada di papan pada *array*. Sistem mencari semua kata yang terbentuk, kemudian dicari kata yang nilainya paling tinggi untuk kemudian dicek kemungkinannya. Jika semua kata – kata yang terbentuk adalah kata yang sesuai maka sistem mencari pertanyaan yang jawabannya merupakan kata tersebut dan mengeluarkan pertanyaan. Jika kata dengan nilai yang paling tinggi bukan merupakan kata yang sesuai maka sistem mencari kata dengan nilai dibawahnya sampai dengan menemukan kata yang sesuai. Jika ternyata semua kata tersebut bukan merupakan kata yang terdapat pada basis data maka sistem akan menambahkan huruf lain dari papan sampai dengan menemukan kata yang sesuai. Untuk lebih jelasnya bisa dilihat pada gambar berikut.



Gambar 6. *Activity* diagram permainan teka teki kata level *normal*

Sistem permainan pada level *hard* adalah sebagai berikut. Sistem menambahkan satu per satu huruf yang sudah ada di papan pada *array* sampai semua huruf ditambahkan. Sistem mencari semua kata yang mungkin terbentuk. Setelah semua daftar kata terbentuk, dicari kata dengan nilai yang paling tinggi kemudian dilakukan pengecekan. Jika semua kata yang terbentuk merupakan kata yang sesuai maka sistem mencari pertanyaan yang jawabannya merupakan kata tersebut. Jika ternyata kata tersebut bukanlah kata yang sesuai maka sistem akan melakukan pengecekan pada kata dengan nilai persis dibawah kata yang dicek sebelumnya sampai dengan menemukan kata yang sesuai.



Gambar 7. *Activity* diagram permainan teka teki kata level *hard*

Pada semua level, setelah pemain berhasil menjawab pertanyaan yang diajukan sistem, maka sistem akan mencari kata dasar dari kata tersebut dan mencari sinonim atau persamaan kata dari kata tersebut untuk mendapatkan nilai tambahan.

Pada permainan ini terdapat tombol *hint* (hanya bisa digunakan 5 kali) yang digunakan pada saat pengguna tidak bisa menjawab pertanyaan dari sistem, kemudian sistem akan menyusun kata jawabannya pada papan. Terdapat juga tombol *next* yang berguna untuk mengganti pertanyaan dan sistem otomatis mengganti huruf – huruf *random* sebelumnya. Tombol *pass* mempunyai aturan dan fungsi yang sama dengan permainan acak kata. Permainan ini juga akan selesai sama dengan aturan permainan pertama.

Untuk mencocokkan kata yang telah disusun oleh pemain, sistem menggunakan metode pencocokkan *string*. Algoritma Pencocokan *string* adalah suatu algoritma yang memeriksa kesamaan pettern suatu kata dengan kata yang terdapat dalam kamus kata. Dua buah kata dikatakan cocok apabila jumlah huruf kedua kata itu sama dan huruf-huruf pada indeks yang bersesuaian bernilai sama. Karena pada permainan ini kata yang dicocokkan adalah kata yang disusun oleh pemain pada papan dan kata yang terdapat pada basis data persis dan tidak perlu mencari *substring* dari *string* maka untuk melakukan pencocokkan kata kita menggunakan metode brute force. Diasumsikan bahwa daftar kata berada di dalam *array* T[1..n] dan *pattern* yang akan dicocokkan berada di dalam array P[1..m], maka langkah – langkah pencocokan string nya adalah sebagai berikut :

1. Mula-mula pettern P dicocokkan pada awal teks T.

2. jika cocok, dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam pattern P dengan karakter yang bersesuaian di teks T sampai:

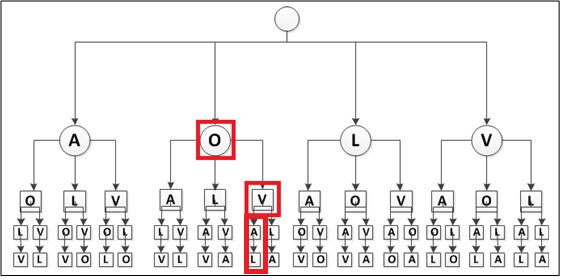
a. semua karakter yang dibandingkan cocok, atau

b. ditemukan ketidakcocokan (belum berhasil).

3. Bila pattern P belum ditemukan kecocokannya dan teks T belum habis, geser pattern P satu karakter ke kanan dan ulangi langkah 2.

Untuk pencarian kata pada basis data yang dibentuk oleh huruf – huruf *random* dilakukan dengan metode *backtracking*. Algoritma *backtracking* (runut balik) merupakan salah satu metode pemecahan masalah yang termasuk dalam strategi yang berbasis pencarian pada ruang status. Algoritma *backtracking* bekerja secara rekursif dan melakukan pencarian solusi persoalan secara sistematis pada semua kemungkinan solusi yang ada. Oleh karena algoritma ini berbasis pada algoritma *Depth-First Search (DFS)*, maka pencarian solusi dilakukan dengan menelusuri suatu struktur berbentuk pohon berakar secara *preorder*. Proses ini dicirikan dengan ekspansi simpul terdalam lebih dahulu sampai tidak ditemukan lagi suksesor dari suatu simpul (Aho, Hopcroft, dan Ullman, 1974, hal:176).

Algoritma *backtracking* mempunyai prinsip dasar kemungkinan solusi. Perbedaan utamanya adalah pada ide dasarnya, semua solusi dibuat dalam bentuk pohon solusi dan algoritma akan menelusuri pohon tersebut secara *Depth First Search* sampai ditemukan solusi yang layak. Nama *backtrack* didapatkan dari sifat algoritma ini yang memanfaat karakteristik himpunan solusinya ang sudah disusun menjadi suatu pohon solusi. Agar lebih jelas bisa dilihat pada gambar di bawah ini



Gambar 8. Contoh pencarian kata menggunakan metode *backtracking*

Gambar di atas adalah contoh pencarian kata yang dilakukan sistem dengan menggunakan metode *backtracking*. Pada gambar di atas terlihat bahwa kata yang diberikan adalah AOLV dan sistem akan mencari kata yang mungkin yang terdapat di dalam database. Dimulai dengan huruf pertama, yaitu A. Maka pencarian langsung dimulai dari indeks huruf A, yaitu 1. Simpul yang dihidupkan pertama kali oleh simpul A adalah O, kemudian terbentuklah string AO, karena masih ada kata yang diawali dengan huruf AO, maka pencarian diteruskan ke simpul berikutnya. Simpul yang dihiduplan pertama kali oleh simpul O adalah L maka terbentuklah string AOL, karena tidak ada kata yang diawali dengan AOL maka pencarian pada simpul O yang pertama dihentikan. Dilanjutkan dengan pencarian simpul kedua yang dihidupkan O, yaitu V dan terbentuklah string AOV. Karena tidak ada kata yang berawalan AOV maka pencarian dihentikan dan dilanjutkan pada simpul dan indeks yang lain sampai dengan menemukan kata yang cocok di basis data. Sampai akhirnya ditemukan kata OVAL.

Sedangkan untuk mencari kata dasar digunakan metode *stemming* porter. *Stemming*  adalah proses untuk memecah setiap varian – varian dari suatu kata menjadi kata dasar. Porter *stemmer* adalah metode *stemming* yang dikembangkan oleh Martin Porter di University of Cambridge pada tahun 1980. *Stemmer* ini dibangun berdasarkan gagasan bahwa akhiran dalam bahasa inggris sebagian besar terdiri dari kombinasi akhiran - akhiran yang kecil dan sederhana. *Stemmer* ini adalah *stemmer* yang menggunakan langkah - langkah linier. *Stemmer* adalah algoritma pembuangan akhiran yang *context sensitive*.

Beberapa definisi yang digunakan dalam algoritma ini :

* Konsonan adalah huruf – huruf selain A, I, U, E, atau O dan Y yang diawali oleh suatu konsonan.
* Vokal adalaha huruf – huruf selain konsonan.

Aturan – aturan yang digunakan untuk melakukan pembuangan akhiran berada dalam bentuk (kondisi) S1 🡪 S2. Kondisi yang diberikan dapat menggunakan ekspresi *and, or,* maupun *not*, dan menggunakan parameter , atau mengandung notasi berikut :

\*S : *stem* berakhir dengan hurud S (dapat juga digunakan untuk huruf – huruf lain)

\*v\*: *stem* mengandung vokal

\*d : *stem* berakhiran *double consonant*

\*o : *stem* berakhiran konsonan, vokal, konsonan dengan konsonan kedua bukan W, X atau Y.

1. **METODOLOGI**
2. **Penyusunan proposal tugas akhir**

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan Proposal Tugas Akhir. Pada proposal ini, penulis mengajukan gagasan pembangunan perangkat lunak permainan acak kata dengan metode *backtracking* dan metode *stemming* Porter pada Android.

1. **Studi literatur**

Pada tahap ini dilakukan pengumpulan informasi yang diperlukan untuk perancangan sistem. Informasi tersebut dapat diperoleh dari literatur seperti paper, maupun buku-buku terkait dengan perancangan maupun pembangunan perangkat lunak. Literatur yang akan digunakan berkaitan dengan :

* 1. Metode brute force pencocokan *string* untuk melakukan pencocokan *pattern* dengan teks
  2. Metode *backtracking* untuk pencarian solusi
  3. Metode *stemming* Porter dalam bahasa inggris untuk pencarian kata dasar

1. **Analisis dan desain perangkat lunak**

Pada tahap analisis kebutuhan dan perancangan sistem untuk merumuskan solusi yang tepat dalam pembuatan aplikasi serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan tersebut. Pada tahap desain ini dilakukan desain class yang digunakan serta desain basis data, dan lain-lain.

1. **Implementasi perangkat lunak**

Pada tahap ini dilakukan implementasi sistem secara menyeluruh. Pengimplementasian dilakukan dengan berbekal pedoman-pedoman yang diperoleh pada tahap sebelumnya.

Tahap implementasi ini diwujudkan dalam bentuk kode program. Untuk tahap implementasi ini penulis menggunakan alat bantu untuk pengkodeannya dengan Eclipse. Dan bahasa pemrograman yang digunakan dalam implementasi ini adalah Java.

1. **Ujicoba dan evaluasi**

Pada tahap ini dilakukan ujicoba terhadap perangkat lunak yang dibuat. Tujuan uji coba perangkat lunak adalah untuk menemukan kesalahan - kesalahan yang mungkin terjadi serta melakukan perbaikan untuk lebih menyempurnakan sistem yang dibuat.

1. **Penyusunan Buku Tugas Akhir**

Pada tahap ini akan dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain :

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Uji Coba dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka
7. **JADWAL PENGERJAAN TUGAS AKHIR**

Tugas akhir ini diharapkan bisa dikerjakan menurut jadwal sebagai berikut:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | Bulan (Tahun 2012) | | | | | | | | | | | | | | | | | | | | |
| Februari | | | | Maret | | | | April | | | | Mei | | | | | Juni | | | |
| Pembuatan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **DAFTAR PUSTAKA**

[1] Wijaya, Riana. 2009. “Pencarian Solusi Permainan Scrabble Menggunakan *Backtracking*”. Universitas Sumatera Utara, Medan.

[2] Izwan, Rahmat. 2007. “Aplikasi Algoritma Pencocokan String Dalam Permainan Scrabble”. Institut Teknologi Bandung, Bandung.

[3] “*The Stemming Porter Algorithm*”. 2005.

<http://snowball.tartarus.org/algorithms/porter/stemmer.html>. Terakhir diakses pada 6 Februari 2012.

**LEMBAR PENGESAHAN**

###### **Surabaya, Februari 2012**

Menyetujui,

Dosen Pembimbing I Dosen Pembimbing II

# **(Imam Kuswardayan, S.Kom., M.T.) (Dwi Sunaryono, S.Kom., M.Kom.)**

# **( NIP. 19761215 200312 1 001) ( NIP. 19720528 199702 1 001)**