Rendra eka H  
175410028

**Algoritma BFS dan Implementasi dalam permainan Rangkai Kata**

Abstrak

Permainan kata merupakan salah satu jenis permainan yang menarik karena dapat menantang

pemainnya untuk mengeluarkan kreativitasnya dalam mengolah kata. Rangkai kata merupakan salah satucontohnya. Rangkai kata merupakan kata acak yang dibentuk dari kata lain yang telah diubah susunan atau jumlah karakter di dalamnya. Pada pengolahankata ini, pemain akan memasukan kata awal yang ingin dibangkitkan Rangkai kata nya. Kemudian pemain akan menebak kata yang merupakan Rangkai kata dari kata awal. Untuk menemukan semua solusi berupa sekumpulan kata ini, kami mencoba membangkitkan Rangkai katanya dengan algoritma Breadth First Search. Algoritma Breadth First Search yang mengeksplorasi setiap cabang dari setiap node sangat tepat untuk diterapkan pada permasalahan ini. Algoritma ini akan membangkitkan pohon solusi secara dinamis bersamaan dengan dilakukannya proses pencarian solusi.

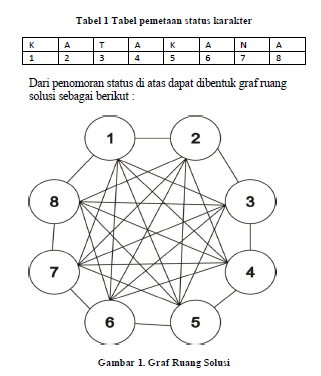
**Kata kunci:** Rangkai kata, Breadth First Search, acak.

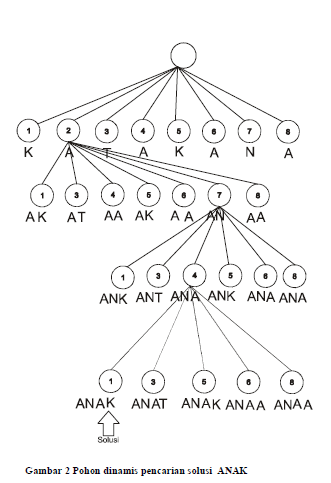
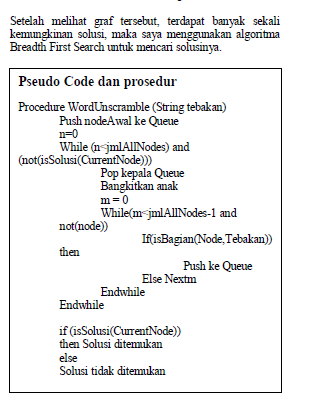
**1. PENDAHULUAN**

Teknologi *game* kini berkembang pesat sejalan dengan perkembangan teknologi informasi. Tidak semua *game* hanya mengandalkan ketangkasan pemain dalam menggerakkan tetikus atau menekan tombol tombol *keyboard.* Ada *game* yang mengandalkan kemampuan logika dan kreativitas pemain untuk menemukan solusinya. Permainan kata Rangkai kata ini menuntut

kemampuan dari pemain dalam mengolah kata. Tujuan dari permainan ini adalah mencari kata yang bisa dibangkitkan dari kata awal. Algoritma Breadth First Search saya implementasi

dalam permainan ini sendiri adalah algoritma penelusuran pohon dengan cara mengunjungi semua anak pada kedalaman tertentu. Kedalaman pohon tidak akan bertambah apabila anaknya belum dikunjungi semua.





Pada kedalaman 1, anak yang memenuhi kriteria akan dimasukan ke dalam queue. Lalu dari anak yang masuk queue tadi dibangkitkan anaknya.

Dalam gambar di atas status 2 akan dimasukkan ke queue karena memenuhi criteria. Namun sebenarnya yang dibangkitkan anaknya bukan cuma status 2, namun juga status 4,6,8.

Namun karena status status itu merepresentasikan hal yang sama,maka tidak saya gambarkan, karena saya anggap hasilnya akan sama dengan anak anak yang dibangkitkan oleh

status ke 2.

Setiap kedalaman akan dilakukan operasi yang sama, sampai kedalaman maksimal atau saat CurrentNode adalah solusi.

Dengan melihat gambar di atas terlihat jelas keuntungan BFS dalam pencarian solusi ini, karena BFS

mengunjungi anak secara melebar. Sehingga untuk kasus dimana tebakan yang lebih sedikit dimasukkan pohon statsu akan berhenti pada node tersebut tanpa harus melangkah lebih dalam.

**2. METODE**

**2.1 Terminologi**

1. Status persoalan (*problem state*)

Simpulsimpul

di dalam pohon dinamis yang

memenuhi kendala (*constraints*).

2. Status solusi (*solusi state*)

Semua simpul telah dikunjungi.

3. Ruang solusi (*solution space*)

Himpunan solusi.

**2.2 Peraturan permainan**

Pertama pemain akan diminta untuk memasukkan

kata yang ingin dibangkitkan Kata Acaknya. Setelah itu

pemain akan memasukkan tebakan kata yang diasumsikan

sebagai kata yang dibangkitkan dari kata awal. Pemain

akan mendapat nilai apabila kata tebakannya merupakan

salah satu kata yang dibangkitkan oleh algoritma BFS.

Nilai yang didapat bergantung dari jumlah karakter kata

tebakan. Semakin banyak karakter tebakan, semakin sulit

dan apabila jawaban benar, pemain akan memperoleh nilai

yang tinggi pula. Dalam implementasinya, diperlukan

struktur data Queue dan pohon.

**2.3 Graf Ruang Solusi**

Agar lebih jelas, akan digunakan contoh untuk

menentukan Graf Ruang Solusinya

Misal :

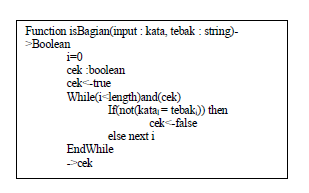
Kata awal yang ingin dibangkitkan : katakana

Tebakan pemain : anak

Setiap karakter dalam kata awal diwakili oleh satu symbol

status. Dalam hal ini setiap karakter diberi penomoran

sebagai berikut :



Penjelasannya adalah sebagai berikut

Fungsi Utama :

Prosedur WordUnscrambler(Tebakan : String, node: int)

Prekondisi :

Dalam prosedur ini kami mengambil asumsi bahwa

setiap node telah merepresentasikan karakter yang

diwakilinya, Perubahan dari status ke status (dari node ke

anaknya) diasumsikan sebagai operasi string untuk

membentuk kata. Kemudian kata tebakan harus lebih kecil

dari kata awal.

1. Masukkan node awal ke Queue

2. Set Kedalaman dengan 0

3. Ambil kepala Queue

4. Bangkitkan semua anak

5. Jika anak memenuhi syarat push ke Queue

6. Jika memenuhi solusi stop, jika tidak ulangi

langkah 3 sampai kedalaman maks

7. Cetak solusi dan perhitungan skor berdasarkan

kedalaman

Fungsi Bantuan :

1. Bangkitkan anak

Prosedur ini membangkitkan anak dari Node

dengan elemen status yang tersisa

2. Fungsi isBagian(input : String,String)

Fungsi ini memeriksa apakah Node tersebut

memenuhi syarat atau merupakan bagian kata

dari tebakan

Sebagai contoh, Saya akan menggunak contoh

persoalan yang telah ada di atas. Secara umum pohon

yang dibentuk adalah sebagai berikut :

