

## PERANCANGAN APLIKASI PENCARIAN FILE DENGAN MENGUNAKAN METODE BEST FIRST SEARCH

Oleh :

**Sudi Suryadi**

Dosen Prodi Manajemen Informatika, AMIK Labuhanbatu  
Rantauprapat, Medan; sudi\_suryadi@yahoo.com

### Abstract

*Hal yang sangat menarik dari metode ini adalah bagaimana searching algorithm yang diterapkan berusaha mencari solusi, yang diistilahkan dengan Goal State (GS), paling optimal dan lengkap dengan parameter kompleksitas waktu dan ruang yang dihadapinya dari kondisi awal atau Initial State (IS) yang diberikan. Oleh karena itu untuk lebih mengetahui lebih detail, maka penulis akan membahas analisa pembangkitan generasi penerus ( successor) simpul induk (parent) hingga didapatkan anak (node) terbaik ditinjau dari mekanisme searching algorithm Depth-First search (DFS), Breadth-First Search (BFS) dan Best-First Search (Best FS). Untuk menanggulangi masalah pencarian file searching algorithm dapat diatasi dengan pencarian heuristik.*

*Metode pencarian heuristik adalah sebuah teknik pencarian yang berdasarkan pada suatu panduan tertentu hingga mencapai keadaan yang diinginkan. Pencarian Heuristik merupakan teknik untuk meningkatkan efisiensi dari proses pencarian. Metode pencarian ini menggunakan Wildcard Character untuk mempermudah user dalam melakukan pencarian file. Dan juga menggunakan String Matching yaitu bagaimana menemukan sebuah untaian karakter ( String) didalam satu kata atau kalimat dengan menelusuri karakter per karakter yang pada string dan membandingkannya dengan untaian karakter yang akan dicari.*

**Keyword :** *first search, pencarian,*

### I. PENDAHULUAN

#### 1.1.Latar Belakang

Dalam perkembangan teknologi komputer pada saat ini khususnya dalam penyelesaian suatu permasalahan yang bersifat problem solving yang dihadapi di dunia nyata ditinjau dari perspektif engineering. Engineering dapat dilakukan melalui pendetakan umum dan pendekatan secara intelligence atau “Cerdas”. Dari perspektif intelligence juga terdapat beberapa paradigma yang umum digunakan menyelesaikan suatu permasalahan yakni Artificial intelligence, Computational Intelligence yang diwakili oleh paradigma Artificial Neural Networks (ANN), Fuzzy Logic dan Evolutionary Computation. Dalam konteks permasalahan problem solving, paradigma yang banyak digunakan adalah Artificial Intelligence (AI) atau Kecerdasan Tiruan. Dalam proses mendapatkan solusi permasalahan problem solving yang diberikan, AI mempunyai beberapa kategori metode untuk menyelesaikannya. Salah satu dari kategori tersebut adalah metode Searching yang mencakup beberapa searching

algorithm yang umum digunakan dalam problem solving.

Hal yang sangat menarik dari metode ini adalah bagaimana searching algorithm yang diterapkan berusaha mencari solusi, yang diistilahkan dengan Goal State (GS), paling optimal dan lengkap dengan parameter kompleksitas waktu dan ruang yang dihadapinya dari kondisi awal atau Initial State (IS) yang diberikan. Oleh karena itu untuk lebih mengetahui lebih detail, maka penulis akan membahas analisa pembangkitan generasi penerus ( successor) simpul induk (parent) hingga didapatkan anak (node) terbaik ditinjau dari mekanisme searching algorithm Depth-First search (DFS), Breadth-First Search (BFS) dan Best-First Search (Best FS). Untuk menanggulangi masalah pencarian file searching algorithm dapat diatasi dengan Menggunakan beberapa metode. Adapun metode pencarian yang ditemukan salah satunya adalah pencarian heuristik. Metode pencarian heuristik adalah sebuah teknik pencarian yang berdasarkan pada suatu panduan tertentu hingga mencapai

keadaan yang diinginkan. Pencarian Heuristik merupakan teknik untuk meningkatkan efisiensi dari proses pencarian. Metode pencarian ini menggunakan Wildcard Character untuk mempermudah user dalam melakukan pencarian file. Dan juga menggunakan String Matching yaitu bagaimana menemukan sebuah untaian karakter (String) didalam satu kata atau kalimat dengan menelusuri karakter per karakter yang pada string dan membandingkannya dengan untaian karakter yang ingin dicari. Dalam fungsi Heuristik yang didalamnya terdapat 3 bagian state yaitu: state awal, state akhir, dan tujuan yang ingin dicapai dari state awal dan state akhir. Metode Heuristik ini menggunakan Algoritma Best First Search yang merupakan metode membangkitkan successor dengan mempertimbangkan harga (didapat dari fungsi heuristik tertentu) dari setiap node, bukan aturan baku.

### 1.2. Perumusan Masalah

Permasalahan yang akan diselesaikan adalah bagaimana menemukan suatu file yang terdapat pada media penyimpanan

### 1.3. Batasan Masalah

Agar pembahasan tidak menyimpang dari tujuan, dibuat suatu batasan masalah, yaitu:

1. Perangkat lunak akan dirancang ditujukan untuk melakukan pencarian file.
2. Perancangan perangkat lunak menggunakan Bahasa Pemrograman Visual Basic 6.0.

## II. LANDASAN TEORI

### 2.1. Pengertian File

File merupakan sekumpulan record yang saling berhubungan dan diperlakukan sebagai satuan kesatuan. Beberapa istilah yang penting dalam file yaitu:

1. Character: dasar utama pembentuk data berupa huruf, angka dan simbol-simbol khusus.
2. Field: Sekumpulan karakter yang membentuk satu untaian arti.
3. Record: Kumpulan dari field.

Ketika bekerja dengan sistem aplikasi, program akan menyimpan data yang kita buat kedalam suatu file data. Struktur file data yang dibentuk akan bervariasi sesuai program aplikasi yang sedang dipergunakan.

Komputer akan membedakan file menjadi dua jenis yaitu :

- File yang dapat dijalankan lewat prompt sistem operasi (executable), contohnya:

EXE (Execute), COM (Command), BAT (Batch File).

- File yang tidak dapat dijalankan secara langsung (data file), contohnya: TEXT (Text), PIC (picture), OVL (Overlay), DBF (Database File), WKS (Worksheet), BAK (Backup) dan lain-lain. (Moh. Sjukani, 2008). Pada umumnya jenis file data yang dibentuk oleh program aplikasi pada saat ini secara global digolongkan kedalam 4 bagian yaitu:

1. File data pengelola kata (Word Processor)  
File data pengelola data adalah file yang dibentuk oleh program aplikasi yang berjenis pengelola kata seperti program MSWord, Wordpad, NotePad, WordStart.
2. File data lembar kerja (Spreadsheet)  
File data Spreadsheet adalah file yang dibentuk oleh program aplikasi yang masuk kategori program lembar kerja seperti MSeXcel, Lotus 123, Supercalc, QuantroPro, dan lain sebagainya.
3. File data pengelola data (Database)  
File database adalah file yang dibentuk oleh program aplikasi yang berjenis pengelola data seperti MSAccess, Dbaseiii, Pradox, FoxBase dan lain sebagainya.
4. File data gambar (Image)

File data gambar seperti Photoshop, PCPPaintBrush, Inset, CoreiDrawn, PhotoImpact, Paint dan lain-lain. (Moh.sjukani, 2008)

#### 2.1.1 Sifat-sifat File

Pada waktu file membentuk, file mempunyai sifat, sifat ini lebih dikenal dengan istilah atribut. Atribut-atribut ini dapat diaktifkan (ON) atau dinokatifkan (OFF), keempat atribut itu adalah:

1. Atribut Readinly.  
Atribut ini dipakai untuk mengetahui dapat tidaknya suatu file ditulis. Sifat ini akan melindungi file dari penghapusan. Seandainya atribut suatu file ON maka file tersebut tidak akan dirubah dan dihapus dengan perintah Disk Operating System (DOS) yaitu Erase atau Del ataupun dengan perintah program lainnya dengan fungsi yang sama. Andaikan attribut ini OFF, file tersebut dapat dirubah dan dihapus.
2. Atribut Hiden  
Atribut Hiden dipergunakan untuk menyembunyikan file. Jika atribut hiden ON maka file tidak dapat dilihat dengan perintah DOS (Dir). Bila atribut hiden OFF, file tersebut dapat dilihat dengan biasa.
3. Atribut Sistem

Attribut sistem biasanya dipakai oleh suatu program khusus yang lebih banyak berhubungan dengan pembentukan sistem komputer. Seandainya ON file tersebut dapat dipakai melalui pemanggilan fungsi DOS dengan bahasa pemrograman tertentu.

#### 4. Attribut Archieve

Attribut ini dipakai untuk mengetahui dapat tidaknya suatu file dibackup atau direstore. Seandainya attribut archive suatu file ON maka file tersebut dapat langsung dibackup. Apabila OFF maka file tersebut tidak dapat dibackup. (Moh.Sjukani,2008) Attribut file pada saat dibentuk mempunyai sifat-sifat:

1. Read Only : OFF
2. Hiden : OFF
3. Sistem : OFF
4. Archive : OFF

#### 2.1.2 Penamaan File

File adalah kumpulan data yang diorganisir yang disimpan dalam disk magnetis atau pita magnetis yang mempunyai sebuah nama yang unik. Microsoft Windows dilengkapi fasilitas penggunaan nama-nama panjang untuk file dan direktori. Nama file panjang (long file name, LFN) adalah suatu nama untuk file yang melebihi ukuran nama file standart (8.3 nama file maksimal 8 karakter dan ekstensi file 3 karakter) yang lama dikenal pada sistem operasi DOS. Salah satu file yang dibawa DOS ke memori pada waktu proses booting adalah file COMMAND.COM. File ini adalah sebuah interpreter perintah bagi DOS. Didalam file ini terkandung pula perintah-perintah internal, yaitu perintah yang dapat langsung kita berikan begitu file COMMAND.COM itu dibawa ke memori. Sebelum kita menggunakan sebuah disket, kita harus memformat disket tersebut dengan tujuan membagi-bagi track (lintasan) dan sector, jumlah track dan sector ini tergantung pada jenis disket tersebut. Untuk melakukan perintah ini harus atau direktori yang sedang aktif pada file FORMAT.COM karena FORMAT termasuk perintah eksternal.

Sekarang windows telah dilengkapi dengan pengaksesan nama file panjang disebut dengan protected-mode FAT (File Allocation Table). Protected-Mode FAT mendukung nama file panjang, menyimpan nama ini dan juga tanggal serta jam tentang file tersebut dibuat dan tanggal terakhir file tersebut diakses pada struktur sistem file FAT. Sistem File Protected-Mode FAT memperbolehkan nama file hingga 256 karakter termasuk karakter penutup null.

Windows mempunyai sistem tersendiri dengan dalam penamaan file. Pada prinsipnya windows mempunyai persyaratan sebagai berikut.

1. Nama file terdiri dari huruf, angka dan beberapa tanda-tanda khusus (misalnya " " dan "\$"). Nama file tidak boleh mengandung tanda koma (,) serta ruang kosong dan jumlahnya terdiri dari 8 karakter maksimum.
2. Nama file sebaiknya terdiri dari dua bagian yaitu bagian "namafile" dan bagian "ekstensi". Ekstensi sifatnya tidak wajib tetapi sangat dianjurkan agar memudahkan pengenalan file. Ekstensi dipakai untuk menunjukkan sifat dan jenis dari file tersebut. Ekstensi dapat berupa huruf atau angka dan maksimum hanya tiga karakter. Diantara karakter nama file dan ekstensi diberi tanda titik (.). Dibawah ini merupakan table dari ekstensi yang umum dipakai.

**Tabel 2.1. Tabel Ekstensi Yang Umum Dipakai**

|      | Pemakaian umum                                                                                                                                    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| .EXE | File yang dapat dieksekusi, hasil kompilasi                                                                                                       |
| .DOC | Yang dibuat dengan microsoft word                                                                                                                 |
| .XLS | File dokumen yang dibuat dengan microsoft excel                                                                                                   |
| .TXT | File yang bersifat teks atau ASCII                                                                                                                |
| .SYS | File yang berisikan perintah yang diperlukan untuk sistem operasi                                                                                 |
| .BAT | File yang berisikan serangkaian perintah yang akan dijalankan berturut-turut sesuai dengan urutan file itu, hanya memberikan suatu perintah saja. |
| .PIC | Format tertentu                                                                                                                                   |
| .OVL | Overlaid yang biasanya diperlukan oleh .EXE                                                                                                       |
| .COM | File yang dapat dieksekusi, hasil kompilasi                                                                                                       |

Sumber: Moh.Sjukani, Struktur Data, 2008

Windows menyimpan perbedaan besar kecil huruf yang digunakan untuk nama file panjang. Tetapi sistem file Protected-Mode FAT tidak membedakan besar kecil huruf, tetapi mengizinkan lebih dari satu untuk memiliki nama yang sama kecuali untuk nama direktori mengizinkan lebih dari satu file untuk memiliki nama yang sama

kecuali untuk nama direktori (nama untuk file direktori dapat sama, tetapi masing-masing hanya satu).

Windows menggunakan sistem file yang sama seperti sistem file DOS, tetapi Windows memperluas dukungan pada penggunaan nama file yang panjang. Panjang sebuah file dalam windows biasa sampai 256 karakter termasuk spasi dan karakter-karakter tanda baca untuk memberikan nama suatu file dalam windows, dapat menggunakan simbol-simbol dan karakter tanda baca dalam nama yang panjang. Contoh :\$, %, “, (spasi), @, ~, !, (,), {,}, #. Berikut ini hal-hal penting yang harus diperhatikan ketika menangani file-file dalam windows adalah tidak semua aplikasi mendukung penggunaan nama file panjang. Jika menggunakan sebuah aplikasi

DOS, aplikasi tersebut besar kemungkinan tidak mengenal nama-nama file yang panjang. Jika dengan menggunakan aplikasi tersebut membuka sebuah file dengan nama file yang panjang dan kemudian menyimpan kembali, maka akan kehilangan nama file yang panjang tersebut. Nama-nama file yang panjang dapat menggunakan karakter-karakter yang tidak dapat digunakan dalam nama-nama file dengan format 8.3 yaitu: . + , ; : = []. Aplikasi-aplikasi backup/restore yang tidak didisain untuk mendukung penggunaan nama-nama file yang panjang dapat merusak nama-nama file yang panjang. Jika menyalin file-file dengan nama-nama file panjang kedalam sistem yang tidak mendukung penggunaan nama-nama file yang panjang akan terpotong menjadi 8 karakter.

Nama-nama file yang panjang membedakan antara huruf kecil dan huruf besar. Contoh nama file “Analisa dan Perancangan Perangkat Lunak Pencarian File” berbeda dari nama file “Analisa dan perancangan perangkat lunak pencarian file”. Hal yang penting yang diketahui adalah jangan menggunakan utility-utility disk, misalnya utility backup/restore yang tidak dapat mendukung penggunaan nama file yang panjang. Jika hal ini dilakukan maka akan merusak atau kehilangan nama file yang panjang. (Moh. Sjukani, 2008)

### 2.1.3 Pencarian File (*File Searching*)

Secara umum search diartikan mencari data dengan cara menelusuri tempat penyimpanan data tersebut. Tempat penyimpanan data dalam memori dapat berupa array atau dapat juga dalam bentuk Linked List. Array merupakan kumpulan elemen-elemen data. Kumpulan elemen tersebut mempunyai susunan tertentu yang teratur. Jumlah elemen terbatas, dan semua elemen mempunyai tipe yang sama. Kumpulan array satu dimensi akan

membentuk dua dimensi. Kumpulan array dua dimensi akan membentuk array tiga dimensi dan seterusnya yang secara teori tingkat dimensi tidak ada batasnya. Data juga dapat berada dalam eksternal storage (umumnya hard disk) berupa suatu file. Searching adalah proses mencari atau pencarian. Data yang disearch adalah data yang berada dalam memori dalam bentuk array. Pencarian data yang sering juga disebut dengan table look-up atau storage and retrieval information merupakan suatu proses untuk mengumpulkan sejumlah informasi yang diperlukan didalam memori komputer dan kemudian mencari kembali informasi yang diperlukan secepat mungkin. Sering kali dihadapkan pada suatu situasi dimana sebenarnya kita hanya memerlukan sedikit informasi diantara sekian banyak informasi yang tersedia sehingga sering timbul pemikiran untuk melupakan atau bahkan menghapus informasi-informasi yang tidak digunakan tersebut. Tetapi pada suatu kondisi timbul pemikiran lain untuk mempertahankan informasi-informasi tersebut dan mengorganisasikannya sedemikian rupa sehingga proses pencarian dapat dilaksanakan dengan secepat mungkin. Secara umum akan menggunakan asumsi bahwa sejumlah (N) data telah tersimpan dalam memori komputer dan persoalannya adalah bagaimana mencari suatu data yang diinginkan. Dalam memori komputer terdiri dari sejumlah area data yang disebut dengan medan (field). Untuk membedakan antara satu data dengan data lain, maka setiap data terdapat suatu field yang mempunyai nilai tertentu. Field data bisa membedakan antara satu data dengan data lain yang disebut dengan field kunci (field key), atau disingkat sebagai kunci. Hubungan antara kunci dengan data bisa merupakan hubungan yang sederhana atau kompleks. Dalam bentuknya yang paling sederhana kunci tersebut terletak didalam data yang disebut dengan internal key, sedangkan kunci yang terletak diluar data tersebut dengan eksternal key. Algoritma pencarian (Searching Algorithm) merupakan algoritma yang menerima argument kunci (sebagai kunci) dan dengan langkah-langkah tersebut akan mencari data yang kuncinya bernilai K.

Algoritma tersebut bisa menemukan data secara utuh atau hanya memperoleh pointer yang menunjuk kepada yang dimaksud. Setelah pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan yaitu data yang dicari ditemukan (successful) atau tidak ditemukan (unsuccessful). Pencarian yang berhasil yang menemukan data yang sering disebut dengan

retrieval. Jika pencarian tidak berhasil menemukan data yang dicari, sering kali perlu menambahkan data tersebut kedalam berkas yang sudah ada. (Moh.Sjukani, 2008).

#### 2.1.4. Metode Pencarian File

Berdasarkan lokasi pencarian file, maka metode pencarian file dapat dibagi menjadi:

- a. Pencarian Internal (Internal Searching)  
Dalam pencarian internal, semua data yang diketahui berada pada disk komputer. Jadi komputer akan melakukan pencarian didalam disk.
- b. Pencarian Eksternal (Eksternal Searching)  
Dalam pencarian Eksternal, tidak semua data yang diketahui berada didalam disk komputer, tetapi ada sejumlah data yang tersimpan pada penyimpanan luar misalnya pita, cakram magnetis, tape disk dan sebagainya. Untuk melakukan suatu pencarian baik berupa file terdapat beberapa metode yang dapat digunakan. Berikut adalah beberapa jenis dari sebuah pencarian:

1. Pencarian Berurutan (Sequential Searching)

Secara garis besar, data yang dicari akan dibandingkan satu persatu sampai data tersebut ditemukan atau tidak ditemukan. Pada saat data yang dicari sudah ketemu, maka proses pencarian langsung dihentikan. Tetapi jika data yang dicari belum ketemu, maka pencarian diteruskan sampai seluruh data dibandingkan. Jika jumlah data beribu-ribu, maka pencarian dilakukan juga beribu-ribu kali. Metode pencarian berurutan ini kurang begitu efisien digunakan jika jumlah data yang ada relative banyak.

2. Pencarian Biner

Setelah vektor data yang diketahui diurutkan, vektor tersebut dibagi menjadi dua sub vektor yang mempunyai jumlah elemen yang sama. Kemudian data dibandingkan dengan data yang terakhir dari sub vektor pertama. Jika data yang dicari lebih kecil, pencarian diteruskan pada sub vektor pertama, berarti data yang dicari kemungkinan terletak pada sub vektor kedua. Dengan demikian pencarian diteruskan pada sub vektor kedua. Proses diulang sampai data yang dicari ditemukan atau tidak ditemukan.

3. Pencarian Berindeks

Pada metode ini, setiap elemen pada tabel data berisi suatu kunci (Key). Kunci merupakan suatu nama unik yang dipilih untuk mewakili banyak data. Kegunaan key ialah untuk mempercepat pencarian data karena adanya suatu batasan pencarian pada data key tertentu saja. Dalam pencarian berindeks terdapat beberapa metode

antara lain: Brute Force, Boyer-Moore, Interpolation Searching dan lain-lain.

4. Pencarian Heuristik

Merupakan teknik yang digunakan untuk meningkatkan efisiensi dari proses pencarian. Dalam pencarian state space, heuristik adalah aturan untuk memilih cabang-cabang yang paling mungkin menyebabkan penyelesaian permasalahan. Metode Pencarian Heuristik ini merupakan teknik pencarian yang berdasarkan pada. Suatu panduan tertentu hingga mencapai keadaan yang diinginkan. Dalam melakukan pencarian file dengan metode ini proses pencariannya lebih cepat dibanding dengan metode pencarian lain. Ada 7 metode pencarian yang terdapat dalam teknik pencarian heuristik, yaitu:

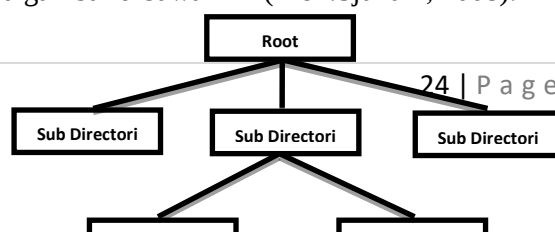
1. Generate and Test (Pembangkitan dan Pengujian)
2. Hill Climbing (Pendakian Bukit)
3. Best First Search (Pencarian Terbaik Pertama)
4. Alpha Beta Pruning
5. Means-End-Analysis
6. Constraint Satisfaction
7. Simulated Annealing

Dalam pencarian ini metode heuristik yang dipakai adalah Best-First Search. (Kusumadewi, S dan H, Purnomo, 2005).

## 2.2. Struktur Direktori

Sebuah direktori merupakan sebuah daerah penyimpanan dalam disk yang mempunyai sebuah nama yang unik yang berisi file-file atau direktori-direktori lainnya. Sejak dahulu pada komputer macintos direktori dinamakan sebagai folder. Sekarang pada sistem operasi Windows menggunakan konsep yang sama yaitu konsep folder, bukan istilah direktori. Sebuah folder dapat berisi aplikasi-aplikasi, icon-icon drive, icon-icon printer dan objek-objek lainnya yang dapat digunakan untuk mengakses sumber daya komputer. Yang dimaksud dengan struktur direktori adalah bagaimana susunan direktori pada suatu media penyimpanan. Direktori utama dan suatu media penyimpanan tersebut dengan root direktori yang memiliki hirarki tertinggi. Root direktori ini terbentuk pada saat penyimpanan tersebut diformat. Direktori-direktori yang berada dibawahnya disebut dengan sub direktori tingkat 1 (Satu), disebut sub direktori tingkat 2 (Dua) dan seterusnya. Struktur direktori demikian merupakan struktur direktori pohon.

Struktur direktori dan sub direktori dapat dilihat pada gambar dibawah ini (Moh.Sjukani, 2008).



Gambar 2.1 Struktur Direktori

Sumber : Moh.Sjukani, Struktur Data, 2008

### 2.2.1 Root Direktori

Root Direktori (Maen Direktori) adalah bagian ketiga dan terakhir dari daerah sistem pada disk. Selain berisi keterangan tentang awal root direktori juga berisi keterangan lain seperti : Nama file, ID, Ukuran file, Tanggal dan waktu pembuatan file Attribut dan tanggal terakhir kali file dibuka.

Setiap direktori yang merupakan cabang dari direktori lain dikatakan sebagai sub direktori karena posisinya lebih rendah. Namun sebuah sub direktori biasanya disebut sebagai direktori saja, kecuali ingin memperjelas hubungannya dengan direktori lain. Setiap direktori dapat berisi masing-masing file, sub direktori atau keduanya. Tiap satu file memiliki suatu direktori entri. Dan pada direktori entri ini ada petunjuk ke FAT entri yang pertama untuk file tersebut. Tiap direktori entri terdiri dari 32 byte. Adapun isi dari direktori entri akan ditunjukkan pada tabel 2.1 berikut ini.

Tabel 2.2 Isi Direktori Entri

| Deskripsi      | Ukuran | Format                                |
|----------------|--------|---------------------------------------|
| Nama File      | 8      | Karakter ASCII                        |
| Ekstensi File  | 3      | Karakter ASCII                        |
| Attribut File  | 1      | Tiap 1 Bit<br>Melambangkan 1 Attribut |
| Cadangan       | 10     | Tidak Dipergunakan                    |
| Waktu          | 2      | Word, dibuat kode                     |
| Tanggal        | 2      | Word, dibuat kode                     |
| Awal FAT Entri | 2      | Word                                  |
| Ukuran File    | 4      | Long Integer                          |

Sumber : Moh.Sjukani,2008

### 2.2.2 Sub Direktori

Ada dua jenis direktori yaitu root dan sub direktori. Pada dasarnya isi dan penggunaan dari kedua tipe direktori adalah sama, yaitu untuk menyimpan informasi file pada disk, tetapi karakteristiknya berbeda.

Adapun perbedaan keduanya adalah:

1. Root direktori disimpan dalam sebuah lokasi yang tetap dalam disk, yaitu pada daerah sistem dan selalu terletak setelah FAT, sedangkan sub direktori disimpan di daerah data halnya file-file yang lain.
2. Root direktori mempunyai ukuran yang tetap, sedangkan sub direktori tidak mempunyai ukuran yang tetap seperti halnya sebuah file dimana ukuran dapat berkurang maupun bertambah tanpa terikat apapun selama ruang disk cukup untuk menyimpannya.
3. Root direktori dibentuk sewaktu pemformatan logis dan tidak dapat lagi dihapus, sedangkan sub direktori dapat dibentuk dan dihapus dengan perintah pada sistem operasi.(Kusumadewi, S dan H Purnomo,2005).

### 2.3.Heuristik

Heuristik adalah sebuah teknik yang meningkatkan efisiensi dari sebuah proses pencarian dengan mengabaikan klaim terhadap kesempurnaan penyelesaian. Metode ini baik karena menunjukkan pada arah atau hasM yang diinginkan, tetapi dilain sisi juga buruk karena ada kemungkinan dilewatinya suatu solusi yang lebih baik. Beberapa teknik heuristik dapat melakukan proses pencarian tanpa mengorbankan klaim terhadap kesempurnaan, sementara teknik-teknik lain dalam heuristik dapat mengakibatkan diabaikannya jalur-jalur terbaik yang mungkin ada. Tetapi rata-rata heuristik dapat meningkatkan kualitas dari rute (path) yang dieksplorasi. Dengan menggunakan heuristik yang baik, dapat diharapkan untuk mendapatkan solusi yang baik (walaupun belum tentu baik ) pada problem-problem berat dengan waktu yang dibutuhkan kurang dari waktu eksponensial. Salah satu teknik yang digunakan untuk mempercepat pencarian solusi adalah teknik heuristik (heuristic). Teknik heuristik digunakan untuk mengeliminasi beberapa kemungkinan solusi tanpa harus mengeksplorasinya secara penuh. Selain itu, teknik heuristik juga membantu memutuskan kemungkinan solusi mana yang pertama kali yang perlu dievaluasi.

Heuristik adalah seni dan ilmu menemukan (art-and-science of discovery). Kata heuristik diturunkan dari bahasa yunani yaitu "eureka" yang berarti "menemukan" (to find atau to discover). Matematikawan yunani bernama Archimedes yang melontarkan kata "eureka" dari sinilah kita menemukan kata "euruka" yang berarti "I have found it". Heuristik berbeda dari algoritma

karena berlaku sebagai panduan (guideline), sedangkan algoritma adalah urutan langkah-langkah penyelesaian. Heuristik mungkin tidak selalu memberikan hasil yang diinginkan, tetapi secara ekstrim ia bernilai pada pemecahan masalah. Heuristik yang baik dapat secara dramatis mengurangi waktu yang dibutuhkan untuk memecahkan masalah dengan cara mengeliminir kebutuhan untuk mempertimbangkan kemungkinan solusi yang tidak perlu. Dalam bidang ilmu komputer, heuristik adalah teknik yang dirancang untuk memecahkan masalah dengan mengabaikan apakah solusi yang dihasilkan dapat dibuktikan secara benar, tetapi biasanya menghasilkan solusi yang baik. Bila diselesaikan secara exhaustive search, harus dicari semua permutasi huruf-huruf pembentuk kata atau kalimat, lalu memeriksa apakah kata atau kalimat yang terbentuk mengandung arti. Teknik heuristik dapat digunakan untuk mengurangi jumlah pencarian solusi. Salah satu teknik heuristik yang digunakan misalnya membuat aturan bahwa dalam bahasa inggris huruf c dan h selalu digunakan berdampingan sebagai ch, sehingga hanya membuat permutasi huruf-huruf dengan c dan h berdampingan. Semua permutasi dengan huruf c dan h tidak berdampingan dan ditolak dari pencarian. (Kusumadewi, S dan H, Purnomo, 2005).

### 2.3.1. Karakteristik Heuristik

Pencarian heuristik memiliki karakteristik. Yang menjadikannya suatu tolak ukur apabila ingin digunakan, karakteristik dari pencarian heuristik adalah:

- Harus mempunyai daerah pengetahuan yang digunakan sebagai panduan pencariannya.
- Umumnya lebih efisien jika dibandingkan dengan pencarian tanpa panduan (blind searches).
- Terkadang disebut juga pencarian yang diberi informasi.
- Pencarian heuristik bekerja dengan memutuskan node selanjutnya yang mana yang terbaik untuk diekspansi (walaupun tidak ada garansi bahwa node tersebut adalah node terbaik).
- Pencarian heuristik memperkirakan nilai untuk mencapai tujuan dari posisi awal. Ini biasa digunakan untuk membuat node baru yang fungsinya dievaluasi dengan  $h(n)$ .
- Node baru dibandingkan dengan node yang memiliki nilai terendah  $g(n)$ .

### 2.3.2. Pencarian Heuristik

Pencarian ini biasanya diistilahkan dengan informet search, pencarian heuristik atau pencarian berdasarkan panduan. George poyla mendefenisikan heuristik sebagai study metode dan kaidah penemuan. Dalam pencarian ruang keadaan, heuristik diyatakan sebagai aturan untuk melakukan pemilihan cabang-cabang dalam ruang keadaan yang paling tepat untuk mencapai solusi permasalahan yang dapat diterima. Dalam melakukan pencarian, perangkat lunak yang dirancang menggunakan String Matching dan Character Wild. String Matching merupakan suatu metode bagaimana melakukan sebuah untai karakter yang lebih panjang (string). String yang dicari disebut pola (pattern), sedangkan Teks yang diperiksa disebut String Teks. Pada Pencarian file ini menggunakan algoritma Best First Search. Sedangkan Wildcard Character Merupakan Karakter-karakter yang digunakan User Dalam melakukan pencarian file. Sistem operasi Windows sampai saat ini menggunakan format Wildcard Charachter untuk mempermudah user dalam melakukan pencarian file. Didalam pencarian Heuristik, Digunakan suatu fungsi Heuristik. Pemilihan fungsi Heuristik adalah agar fungsi tersebut menjadi lebih akurat dan efisien.

Didalam Penggunaan fungsi heuristik hal yang harus penulis perhatikan fungsi tersebut harus mempermudah masalah, menggabungkan beberapa fungsi yang dapat diterima, bersifat statistik, serta memberikan bobot pada faktor-faktor yang diperkirakan mempengaruhi tercapainya suatu tujuan (Learning from Observation). Ini merupakan teknik yang digunakan untuk meningkatkan efisiensi dari proses pencarian. Dalam pencarian State space, Heuristik adalah aturan untuk memilih cabang-cabang yang paling mungkin menyebabkan penyelesaian permasalahan dapat diterima. (Kusumadewi, S dan H, Purnomo, 2005).

### 2.3.3. Teknik pencarian Heuristik

Teknik pencarian heuristik (Heuristic Searching) Merupakan suatu strategi untuk melakukan proses pencarian ruang keadaan (State Space) suatu problema secara selektif, yang memandu proses pencarian yang kita lakukan di sepanjang jalur yang memiliki kemungkinan sukses paling besar, dan mengesampingkan usaha yang bodoh dan memboroskan waktu. Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (Completeness). Untuk dapat menerapkan heuristik tersebut dengan baik dalam suatu Domain tertentu, diperlukan suatu

Fungsi Heuristik. Fungsi Heuristik ini digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

Pada umumnya manusia mempertimbangkan sejumlah alternatif strategi dalam menyelesaikan suatu problema. Dalam permainan catur misalnya, seorang pemain mempertimbangkan sejumlah kemungkinan tentang langkah-langkah berikutnya, memilih yang terbaik menurut kriteria tertentu seperti kemungkinan respon lawannya. Aspek tingkah laku cerdas yang mendasari teknik penyelesaian problema seperti dalam permainan catur tersebut dinamakan proses pencarian ruang keadaan {Space state search}. Exhaustive Search adalah proses pencarian terhadap seluruh ruang keadaan serangkain langkah yang paling dimungkinkan untuk menghasilkan kemungkinan. Walaupun metode ini dapat diterapkan pada setiap ruang keadaan, namun ukuran ruang keadaan yang sangat besar membuat pendekatan ini secara praktis tidak dimungkinkan (dalam permainan catur terdapat 10 keadaan). Bila kasus ini diimplementasikan kedalam sistem komputer, maka akan membutuhkan memori yang sangat besar, dan waktu pencarian yang sangat lama. Dengan kata lain metode Exhaustive search ini tidak efisien dan tidak efektif, sehingga tidak praktis untuk diimplementasikan. Teknik pencarian heuristik tergolong atas 2 bagian antara lain:

1. pencarian Buta (Blind search), disimpulkan bahwa tidak ada informasi awal yang digunakan dalam proses pencarian. Pencarian Buta (Blind Search) terdiri atas 2 bagian, antara lain:
  1. Pencarian Melebar Pertama (Breadth First Search).
  2. Pencarian Mendalam Pertama (Depth First Search).
- Pencarian Terbimbing (Heuristic Search), disimpulkan bahwa adanya informasi awal yang digunakan dalam proses pencarian. Pencarian Terbimbing (Heuristic Search) terdiri atas 2 bagian, antara lain:
  1. Pendakian Bukit (Hill Climbing)
  2. Pencarian Terbaik Pertama (Best-First-Search)

#### 2.3.4. Metode Pencarian Heuristik

Hal penting dalam menentukan keberhasilan sistem cerdas adalah kesuksesan dalam pencarian. Pencarian merupakan suatu proses mencari solusi

dari suatu permasalahan melalui sekumpulan kemungkinan ruang keadaan (state space). Ruang keadaan merupakan suatu ruang yang berisi semua keadaan yang pasti. Untuk mengukur performansi metode pencarian, terdapat empat kriteria yang dapat digunakan antara lain:

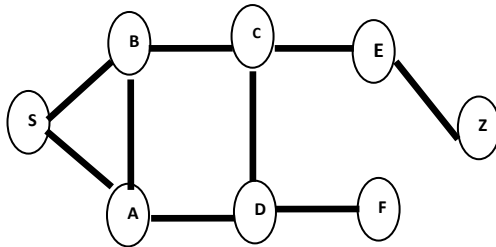
1. Completeness yaitu apakah metode tersebut menjamin penemuan solusi jika solusinya memang ada?
2. Time Complexity yaitu berapa lama waktu yang diperlukan?
3. Space Complexity yaitu berapa banyak memori yang diperlukan?
4. Optimality yaitu apakah metode tersebut menjamin menemukan solusi yang terbaik jika terdapat beberapa solusi berbeda?

Metode pencarian dikatakan penting untuk menyelesaikan permasalahan, karena setiap state (keadaan) menggambarkan langkah-langkah untuk menyelesaikan permasalahan. Metode pencarian dikatakan penting untuk perencanaan, karena dalam sebuah pencarian akan menentukan apa yang harus dilakukan, dimana setiap state menggambarkan kemungkinan posisi pada suatu permasalahan. Metode pencarian file adalah bagian dari kesimpulan, dimana setiap state menggambarkan hipotesis dalam sebuah rangkaian deduktif. (Munir, Rinaldi, 2005).

#### 2.3.5. Pohon Pelacakan

Untuk menghindari kemungkinan adanya proses pelacakan suatu node secara berulang, maka digunakan struktur pohon. Struktur pohon digunakan untuk menggambarkan keadaan secara hirarkis. Struktur pohon juga terdiri dari beberapa node. Node yang terletak pada level-0 disebut juga "akar". Node akar menunjukkan keadaan awal yang biasanya merupakan topik atau objek. Node akar terletak pada level ke-0. Node akar mempunyai beberapa percabangan yang terdiri atas beberapa node successor yang sering disebut dengan nama "anak" dan merupakan node-node perantara. Namun jika dilakukan pencarian mundur, maka dapat dikatakan bahwa node tersebut memiliki predecessor. Node-node yang tidak mempunyai anak sering disebut dengan nama node "daun" yang menunjukkan akhir dan suatu pencarian, dapat berupa tujuan yang diharapkan (goal) atau jalan buntu (dead end). Dalam gambar 2.2 ditampilkan sebuah contoh permasalahan mendasar untuk digunakan dalam penggunaan beberapa metode pencarian. Simpul S merupakan simpul awal dimulainya penelusuran, simpul Z

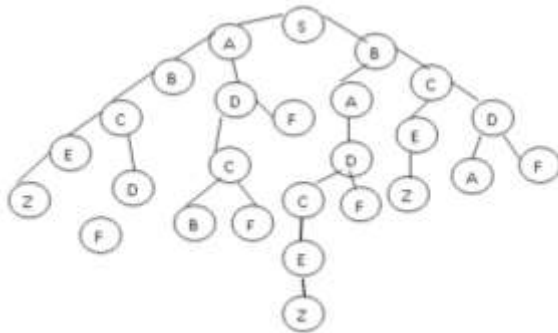
adalah simpul yang akan menjadi tujuan (Kusumadewi,S dan H, Purnomo,2005).



Gambar 2.2 Contoh Graph Yang Berisi Path Antara Kota

Sumber: Moh.Sjukani,Struktur Data,2008.

Dan Graph di atas, dibuat struktur tree-nya. Pada gambar 2.4 menggambarkan tree yang didapat dari graph gambar 2.3.



Gambar 2.3 Struktur Tree dan Graph

Sumber: Moh.Sjukani,Struktur Data,2008

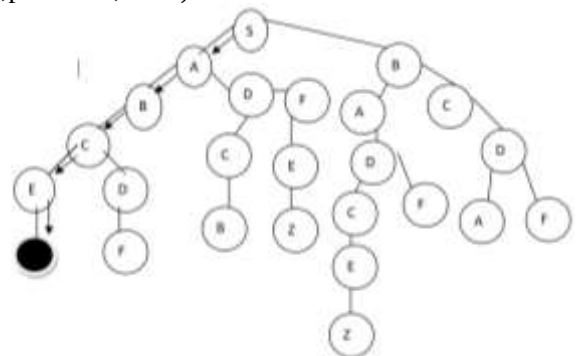
### 2.3.6 Metode Depth First Search(Pencarian Mendalam Pertama)

Pada Metode Depth First Search,Proses pencarian akan dilaksanakan pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel. Pencarian dimulai dari node akar ke level yang lebih tinggi.proses ini diulangi terus hingga ditemukannya solusi.pencarian dilakukan pada suatu simpul dalam setiap level dan yang paling kin. Jika level yang paling dalam tidak ditemukan solusi,maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kin dapat dihapus dari memori. Jika pada level yang paling dalam tidak ditemukan solusi, maka pencarian dilanjutkan pada level sebelumnya. Demikian seterusnya sampai ditemukan solusi. Bila simpul yang dibentuk tidak mengarah ke pencarian solusi maka

pencarian solusi akan diteruskan dengan membangun simpul berikutnya, begitu seterusnya sampai solusi ditemukan. Karena pencarian solusi berarti pencarian traversal terhadap pohon, Maka algoritma traversal seperti Depth First Search (DFS) dapat digunakan untuk memecahkan permasalahan. Untuk mendapatkan hasil analisa dan permasalahan ini, maka penulis melakukan kajian literatur dan pengujian algoritma.

Ada dua metode umum pembentukan pohon ruang status yang dibentuk selama pencarian solusi, yaitu metode Breadth First Search (BFS) dan metode Depth First Search (DFS). Metode DFS(Depth First Search) diterapkan untuk permasalahan guna mengatasi kelemahan metode BFS(Breadth First Search) yaitu karena lamanya waktu yang diperlukan untuk menuju simpul solusi.Pencarian solusi dalam Depth First Search(DFS) dicirikan dengan ekspansi simpul-simpul terdalam terlebih dahulu. Mula-mula simpul akar dibangkitkan pertama kali, kemudian sebuah simpul pada aras kedua, lalu sebuah simpul pada aras ketiga dan begitu seterusnya. Jadi, pencarian mengikuti sebuah lintasan tunggal mulai dari simpul akar terus menurun ke bawah ke simpul-simpul pada aras di bawahnya. Jika pencarian mencapai simpul yang tidak memiliki suksesor, pencarian dilakukan pada lintasan lainnya atau yang dikenal dengan backtrak.

Pada Gambar di bawah ini adalah lintasan pohon(tree) ruang status yang akan digunakan dalam implementasi .(Kusumadewi,S dan H,purnomo,2005).



Hasil solusi:S-A-B-C-E-Z

Gambar 2.4 tree untuk Depth First Search  
Sumber : Moh.Sjukani,Struktur Data,2008

Kelebihan Depth First Search Adalah:

- Pemakaian memori hanya sedikit, berbeda jauh dengan Best First Search(BFS) yang harus menyimpan semua node yang pernah dibangkitkan.



yang lebih rendah meskipun node pada level yang lebih rendah tersebut memiliki nilai heuristik yang lebih baik. Hal ini berbeda dengan metode Best-First Search, dimana pencarian dilakukan dengan melihat satu lintasan, dan memungkinkan untuk berpindah kelintasan lain.

Pada setiap langkah proses pencarian terbaik pertama, node-node dapat dipilih dengan menerapkan fungsi heuristik yang memadai pada setiap node/simpul yang telah dipilih dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya.

Untuk mengimplementasikan algoritma pencarian ini diperlukan dua buah senarai, yaitu: OPEN untuk mengelolah node-node yang pernah dibangkitkan tetapi belum dievaluasi dan CLOSE untuk mengelolah node-node yang pernah dibangkitkan dan sudah dievaluasi. Penggunaan Best First Search sebagai fungsi untuk memilih node terbaik untuk dieksplorasikan pertama kali (walapun tidak pasti itu adalah node terbaik), dengan menggunakan pengetahuan awal. Algoritma selengkapnya adalah sebagai berikut:

Algoritma Best-First Search:

1. Mulai dengan yang berisi state awal
2. Hingga tujuan ditemukan atau tidak ada node yang tersisa lakukan:
  - Pilih node terbaik pada OPEN
  - Bangkitkan suksesornya

Pada setiap succesor lakukan :

- Jika tidak ada yang dibangkitkan, evaluasi node tersebut dan tambahkan kedalam OPEN dan rekam parent-nya.
- Jika telah dibangkitkan, ganti jika merupakan rute baru yang lebih baik dan rute lama. Pada kasus ini perbaharui nilai yang didapat node dan setiap succesor yang dimiliki node.

Pada gambar dibawah ini adalah contoh pohon (Tree) ruang status yang akan digunakan dalam implementasi. (Munir,Rinaldi,2005)

Diasumsikan node dengan nilai yang lebih besar,yang memiliki nilai evaluasi yang lebih baik. Pada keadaan awal, antrian berisi A. Pengujian dilakukan dilevel pertama, node D memiliki nilai terbaik, sehingga menempati antrian pertama, disusul dengan C dan E. Node D memiliki cabang E dan F yang masing-masing bernilai 2 dan 4. Dengan demikian C merupakan pilihan terbaik dengan menempati antrian pertama. Demikian seterusnya. (Munir,Rinaldi,2005).

### 2.3.9 Algoritma A\*

Algoritma A merupakan algoritma Best-First Search pemodifikasian heuristik.

Algoritma ini akan meminimumkan total biaya lintasan, dan pada kondisi yang tepat akan memberikan solusi yang terbaik dalam waktu yang optimal.

Algoritma A juga membutuhkan dua antrian, yaitu OPEN dan CLOSED. Selain

Antrian tersebut, ada juga fungsi heuristik yang memprediksikan keuntungan

Setiap node yang dibuat. Hal ini akan memungkinkan algoritma untuk mela-

Kukan pencarian-pencarian lintasan yang lebih dapat diharapkan.  $F(n)$  sebagai

Pendekatan dari fungsi  $f(n)$  yang merupakan fungsi evaluasi yang sebenarnya

Terhadap node (n).

Dalam banyak penerapan, akan lebih baik jika fungsi ini didefinisikan sebagai kombinasi atau jumlah dua komponen yaitu  $g(n)$  dan  $h(n)$  merupakan ukuran biaya yang dikeluarkan dari keadaan awal sampai (n) ke node (n). Nilai yang diperoleh  $g(n)$  merupakan jumlah biaya penerapan setiap aturan yang dilakukan pada sepanjang lintasan terbaik menuju suatu simpul dan bukan merupakan hasil estimasi. Adapun fungsi  $h(n)$  merupakan pengukur biaya tambahan yang harus dikeluarkan

Paling mendekati solusi terbaik. Kemudian ,simpul pertama pada list diambil,dibangkitkan suksesornya dan kemudian succesor ini disimpan ke dalam list sesuai dengan urutan yang terbaik untuk solusi. Dengan heuristik,maka A\* pasti akan mendapatkan solusi (jika memang ada solusinya). Algoritma A\*search menyelesaikan sebuah permasalahan lintasan terpendek menggunakan penyelidikan sendiri untuk mempercepat pencarian. Pada bidang networking dan telekomunikasi, masalah lintasan terpendek ini sering disebut the min-delay path problem dan biasanya tergantung pada luas masalah lintasan. Seperti : Shortest (min-delay) widest path or Widest shortest (min-delay) path.

### 2.4. String Matching

String Matching merupakan suara metode bagaimana menemukan suatu untai karakter (String) didalam satu kata kalimat (Sting). Hal ini dilakukan dengan menelusuri karakter per karakter yang ada pada string dan membandingkannya dengan untai karakter yang ingin dicari. Ada banyak algoritma String Matching antara lain:

Algoritma Generate and Test, Hill Climbing, Best-First Search, Simulated Annealing dan lain-lain.

Pada metode pencarian ini digunakan Algoritma Best First Search.  
(Moh.Sjukani,2008)

### 2.5. Wildcard Character

Wildcard Character merupakan karakter-karakter yang digunakan user dalam melakukan pencarian file. Sistem operasi windows sampai saat ini menggunakan format wildcard character untuk mempermudah user dalam melakukan pencarian file. Wildcard Character terdiri dari 2 jenis yaitu Asterisk dan Question Mark. Asterisk merupakan karakter yang disimbolkan dengan tanda "\*". Tanda merupakan tanda yang mewakili banyak karakter file yang dicari file. Question Mark merupakan karakter yang disimbolkan dengan tanda "?". Tanda merupakan tanda yang mewakili satu karakter file yang ingin dicari user. (Moh. Sjukani,2008).

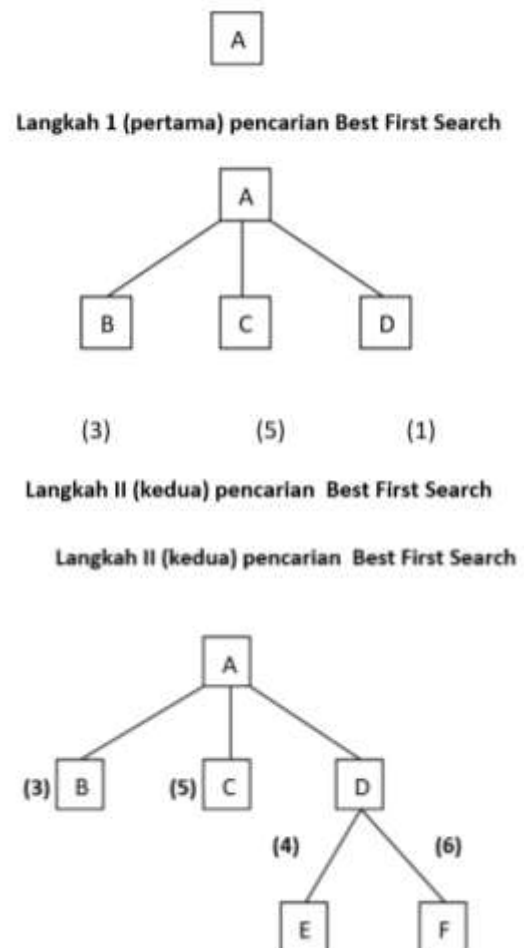
## III. ANALISA DAN PERANCANGAN

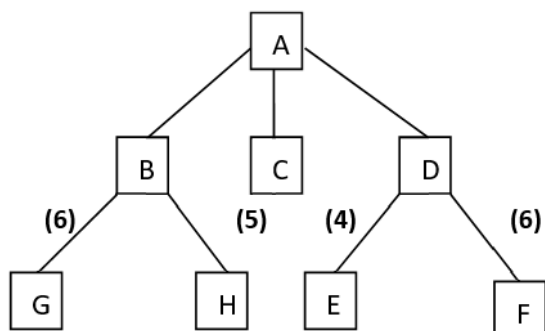
### 3.1 Analisa

Pada analisa pembahasan ini, penulis akan menguraikan bagaimana proses pencarian file dengan menggunakan metode Heuristik Best-First Search dilakukan, dan bagaimana penggunaan String Matching dan Wildcard Character pada proses pencarian. Selain itu, penulis juga akan menjelaskan operasi pengolahan file yang dilakukan pada perangkat lunak ini. Permasalahan pencarian merupakan hal yang sering dijumpai oleh peneliti dibidang kecerdasan buatan. Permasalahan ini merupakan hal penting dalam menentukan keberhasilan sistem keerdasan buatan. Dalam metode pencarian, yang pertama adalah metode yang sederhana yang hanya berusaha mencari kemungkinan penyelesaian. Metode yang termasuk pada bagian ini adalah Depth-First Search, Breadth First Search dan Best-First Search. Metode ini merupakan kombinasi dari Metode Depth First Search dan Breadth-First Search. Pada metode Best-First Search, pencarian diperbolehkan mengunjungi node yang ada di level yang lebih rendah. Pada setiap langkah proses pencarian terbaik pertama, kita memilih node-node dengan menerapkan fungsi heuristik yang memadai pada setiap simpul/node yang kita pilih dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya. Pencarian dilakukan dengan melihat satu lintasan, dan memungkinkan untuk berpindah kelintasan lain. Langkah-langkah pencarian Best First Search dapat dilihat pada contoh berikut ini :

1. Mulai dengan open yang membangkitkan keadaan awal sebuah node yaitu node A.

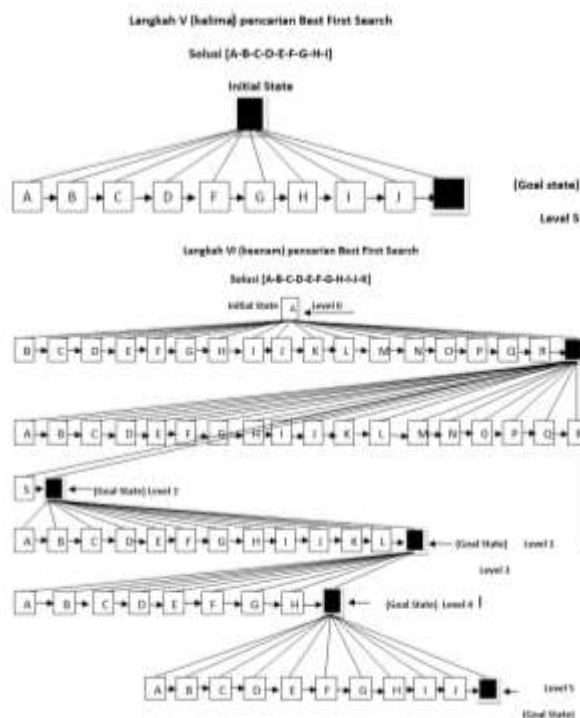
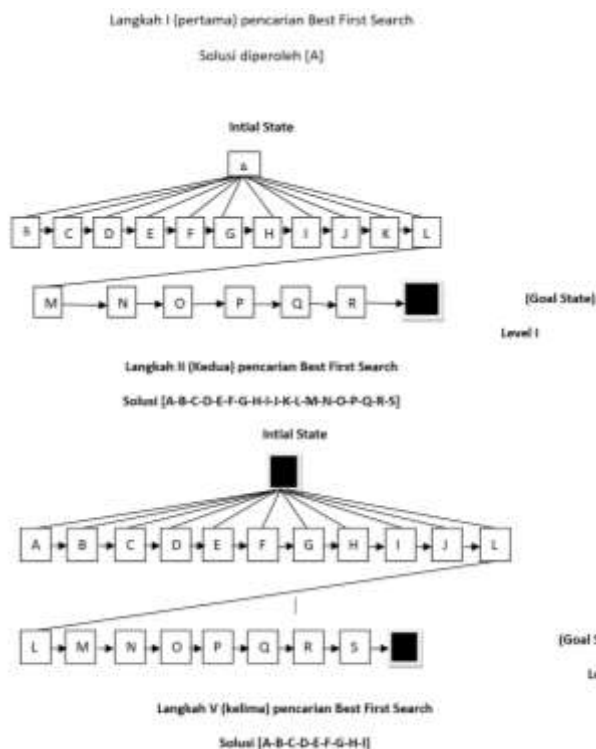
2. Bangkitkan semua successor yang ada dalam node A dan harga node yang paling minimal dibangkitkan kembali.
3. Ulangi dari langkah 1(pertama) sampai semua node telah dibangkitkan dan dipilih harga node yang paling minimal untuk setiap successor dan node.
4. Demikian seterusnya sampai semua node selesai/ditemukan node tujuan.



**Langkah III (ketiga) pencarian Best First Search****Langkah IV (keempat) pencarian Best First Search****Gambar 3.1 Ilustrasi pencarian Best First Search**

**Sumber: Munir, Rinaldi, Strategi Algoritmik, 2005.**

Pada contoh pencarian file pada gambar berikut ini merupakan salah satu contoh bentuk langkah pencarian terhadap file yaitu dengan menggunakan karakter kata “STMIK” dengan menggunakan Metode Heuristik Best-First Search pada sebuah direktori dalam sebuah media penyimpanan komputer (Harddisk), misalkan pencarian tersebut dimulai dari direktori C :\....



Maka dapat disimpulkan bahwa keunggulan dan Metode Best First Search adalah :

1. Membutuhkan memori yang cukup relative kecil, karena hanya node-node pada lintasan yang aktif saja yang disimpan.
2. Metode Best First Search akan menemukan solusi tanpa harus menguji lebih banyak lebih banyak lagi dalam ruang keadaan.

Metode Best First Search menggunakan nilai heuristik tiap simpul yang dibuka.

Bila goal state/tujuan belum ditemukan akan dilakukan pemeriksaan pada simpul berikutnya dengan nilai heuristik terkecil pada kedalaman yang sama. Simpul tersebut kemudian dibuka dan diperiksa apakah terdapat goal state pada cabang-cabangnya. Bila goal state belum ditemukan, akan dilakukan proses yang sama pada simpul berikutnya.

**3.2.Perancangan**

Perancangan perangkat lunak pencarian file dengan menggunakan metode pencarian heuristik, diawali dan merancang antar muka perangkat lunak, kemudian dilanjutkan dengan penulisan kode-kode programnya. Untuk memberikan suatu gambaran dalam pembuatan aplikasi pencarian file yang sebenarnya, maka penulis membuat perancangan from-form dan perancangan output yang akan dipakai pada aplikasi pencarian file dengan metode pencarian heuristik. Rancangan antar muka dan perangkat

lunak pencarian file ini menggunakan bahasa pemrograman Visual Basic 6.0. Penggunaan pemrograman Bahasa Visual ini sangat mendukung untuk tampilan antar muka terhadap perancangan perangkat lunak yang akan dirancang, dan pertama sekali dilakukan perancangan form, karena form merupakan bentuk tampilan program. Perancangan output merupakan hasil keluaran dan perangkat lunak yang akan dirancang.

Sistem yang dirancang adalah suatu perangkat lunak pencarian file, yang dapat mencari file didalam harddisk berdasarkan panduan yang diberikan yaitu paten dan kata yang dikandung untuk arah pencarian. Dimana melalui masukan dan user, berupa directori mencari, pola file, dan kata, maka file akan mulai mencari pola tersebut directori dengan mencari file tersebut.

### 3.2.1. Perancangan Output

From-form yang akan dirancang pada perangkat lunak pencarian file ini adalah : from utama, from cari file, from about me dan from keluar

#### 3.2.1.1. From Utama (frmMaster.frm)

From utama adalah form yang akan ditampilkan saat user mengoperasikan perangkat lunak ini. Proses pencarian ini terdiri dari tombol File, About Me dan keluar. Jika hendak melakukan proses pencarian, maka user menekan tombol file lalu user memilih Can File. Apabila user hendak keluar dan aplikasi, maka user menekan tombol keluar dan akan muncul pesan boks yang bertuliskan “Apakah anda yakin ingin keluar “ dan tombol ini memiliki pilihan antara lain Yes atau No.

From pencarian file adalah form yang akan ditampilkan saat user menekan tombol File pada From menu utama dan memilih tombol Can File.. Pada from inilah proses pencarian dilakukan. Jika hendak melakukan

proses pencarian, maka user menulis nama file yang hendak dicari pada label Nama File yang tertera pada desain perangkat lunak. Form pencarian file menampilkan tombol-tombol control, sederetan toolbar yang akan ditampilkan pada form ini antara lain tombol Can, berhenti dan keluar, serta untuk memilih drive dan untuk memasukkan pola tombol pattern atau nama file yang ingin dicari dengan menggunakan Wildcard Karakter, lalu anda dapat memasukkan kata sebagai panduan pencarian file.

From Cari (frmCari.frm)

Gambar 3.4 Rancangan Form Pencarian File

#### 3.2.1.2 Form About Me (frmabout.frm)

From about me adalah yang akan ditampilkan jika menekan tombol about, yang merupakan keterangan penulis dan nama perangkat lunak. Bentuk rancangan yang diinginkan adalah:

Gambar 3.5 Rancangan Form About

#### 3.2.1.3. From Keluar (Exit)

From keluar (exit) adalah form yang berada pada tampilan menu utama..Jika pemakai ('user, ) hendak keluar dan aplikasi perangkat lunak maka akan muncul kotak pesan (Messages Box,) yang bertuliskan “Apakah anda yakin ingin keluar”. Jika “ Yes” maka program aplikasi perangkat lunak akan

menutup semua fungsi yang ada, sebaliknya jika “No” maka aplikasi program perangkat lunak tetap beroperasi. Bentuk rancangannya adalah sebagai berikut:

Gambar 3.6 Rancangan Form Keluar

#### IV. KESIMPULAN DAN SARAN

##### 4.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan mengenai perangkat lunak pencarian file dengan menggunakan Algoritma Pencarian Best- First Search , maka penulis dapat mengambil beberapa kesimpulan sebagai berikut:

1. Perangkat lunak yang dirancang dapat melakukan proses pencarian file dengan menggunakan Algoritma Best-First Search, dimana file-file yang akan dicari terlebih dahulu memberikan sebuah kata kunci (key) dalam bentuk karakter yang berupa nama file tersebut.
2. Untuk mempermudah dalam melakukan pencarian file, maka pemakai (user) dapat menggunakan string matching dan wild character.

##### 4.2 Saran

Adapun saran yang dapat diberikan setelah melakukan pembahasan mengenai perangkat lunak pencarian file dengan menggunakan Algoritma Breadth-First Search ini yaitu:

1. Melengkapi kriteria pencarian pada perangkat lunak ini yaitu pencarian berdasarkan isi dan perangkat lunak ini yaitu pencarian berdasarkan nama file dan pada file yang akan dicari seperti pada Utility search pada windows.
2. Menambah kemampuan dan perangkat lunak ini agar dapat digunakan untuk mencari file-file yang ada pada sebuah jaringan komputer.
3. Pencarian dengan perangkat lunak ini sudah dapat menyerupai Utility search

pada windows walaupun beberapa proses pencarian belum sempurna dilakukan.

#### DAFTAR PUSTAKA

- Kusumadewi, S. Dan Purnomo,H. 2005. Penyelesaian Masalah Optimal dengan Teknik-teknik Heuristik. Graha Ilmu, Yogyakarta.
- Kusumadewi,S.,2003. ArtUlcial Intellgent (Teknik dan Aplikasinya). Yogyakarta: Graha Ilmu.
- Mooh.Sjukani, 2008. Algoritma dan Struktur Data dengan C,C++.dan Java, Mitra Wacana Media.
- Munir, Rinald. 2005. Strategi Algoritmik. Teknik Informatika ITB, Bandung.
- Panjaitan, W. L.2007. Dasar-dasar Komputasi Cerdas. Andi, Yogyakarta.
- Suyanto. 2007. Artificial Intellegent Searching Reasoning Planning and Learning. Informatik, Bandung