Perbandingan Metode Depth First Search (DFS) dan Breadth First Search (BFS) untuk Mengidentifikasi Kerusakan Handphone

Hendry
STMIK IBBI
Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548
Email: hendry@gmail.com

Abstrak

Penelitian ini digunakan untuk membandingkan metode Depth First Search (DFS) dengan Breadth Firsth Search (BFS) yang diimplementasikan untuk menelusuri kerusakan handphone. Dari pengumpulan Role Base dari sistem pakar dapat dengan mudah untuk mencari solusi oleh user. Sistem Pakar itu sendiri diciptakan untuk menggantikan pakar itu sendiri untuk menyelesaikan masalah dalam kecepatan keakuratan dan juga ketelitiannya. Karena tidak semua orang mengetahui atau mempunyai keahlian dalam memperbaiki ponsel yang rusak. Dari permasalahan tersebut maka dibutuhkan sebuah sistem pakar yang dapat membantu mengatasi dan mencari solusi cara memperbaiki atau mengatasinya Perancangan Sistem pakar Untuk Menelusuri Kerusakan pada *Handphone* Menggunakan metode Depth First Search (DFS) dengan Breadth Firsth Search (BFS) ini merupakan suatu sistem berbasis komputer yang digunakan sebagai alat bantu bagi para pengguna/teknisi *handphone* dalam menyelesaikan permasalahan yang ada pada ponsel. Sistem pakar dirancang sedemikian rupa sehingga menjadi suatu sistem yang berbentuk *interface* dalam membantu memecahkan permasalahan yang dialami para pengguna sistem pakar.

Kata Kunci: *Sistem Pakar*, *Identifikasi*, metode *DFS dan BFS*

Abstract

This research method is used to compare Depth First Search (DFS) with the Breadth Firsth Search (BFS) to be implemented to trace cell phone damage. Rule Base from the collection of expert system can easily to find a solution by the user. Expert System itself was created to replace the experts themselves to solve problems in accuracy and speed accuracy. Because not everyone knows or has expertise in repairing a damaged phone. Of the problems it is required an expert system that can help to overcome and find a solution how to fix or cope Design Expert System For Tracing Faults in Mobile Using the method of Depth First Search (DFS) with the Breadth Firsth Search (BFS) is a computer-based system used as a tool for the user / mobile phone technician in solving the existing problems on the phone. Expert systems are designed so that becomes a system that shaped interface in helping solve the problems experienced by users of expert systems.

Keywords: *Expert System, Identification, DFS and BFS Algorithm*

1. Pendahuluan

Telepon selular atau sering disebut dengan *Handphone* merupakan suatu alat komunikasi yang vital saat ini dan tidak dapat dipisahkan dari kehidupan manusia dalam beraktivitas, diantaranya dalam bertransaksi bisnis, penjualan, pembelian, kebutuhan akan informasi dan komunikasi interaktif lainnya.

Melihat fungsi diatas, ponsel akan bermanfaat bila dapat melakukan komunikasi sebagaimana mestinya. Kerusakan atau gangguan yang terjadi pada ponsel, seperti : tiba-tiba mati *(hang),* tidak ada sinyal *(no signal)*, atau terus menerus mencari sinyal *(searching signal)* karena tidak dapat menangkap dan mengunci jaringan, hal ini dapat mengakibatkan ponsel tersebut tidak bermanfaat atau tidak berfungsi sebagaimana mestinya. Oleh sebab itu untuk mengatasinya, pengguna seharusnya mengetahui jenis kerusakan yang terjadi serta cara memperbaikinya paling tidak mengatasinya.

Sistem Pakar merupakan salah satu cabang dari *Artificial Intelligence* (AI) yang dapat digunakan secara luas, *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia pakar. Seorang pakar

disini merupakan orang yang memiliki keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak miliki. Teknologi sistem pakar ini meliputi bahasa sistem pakar, program, perangkat lunak dan perangkat keras yang dirancang untuk membantu pengembangan dan pembuatan sistem pakar *[5]*.

Kemungkinan *Counter* ponsel menerima perbaikan yang tidak sesuai dengan pengguna miliki. Kebanyakan *Counter* ponsel biasanya hanya melakuakan transaksi jual-beli. Atau *Counter* tersebut akan mengirimkan ponsel ke *counter* ponsel lain yang memang menangani masalah tersebut dan tentunya bersifat pribadi, maksudnya hanya orang-orang pada *counter* tertentu saja yang mengetahui tempat memperbaikinya. Melihat fakta tersebut, tidak ada salahnya sebagai pengguna ponsel mengetahui cara mendidentifikasi dan memperbaiki kerusakan ponsel. Resiko memang selalu ada. Namun, bila pengguna berhati-hati dengan cermat, resiko tersebut sedikit banyaknya dapat diminimalisasikan atau bahkan dapat dihilangkan. Namun untuk menjadi seorang ahli dalam perbaikan ponsel, dibutuhkan waktu pendidikan yang cukup lama serta memerlukan biaya dan pengalaman yang cukup memadai.

2. Metodologi Penelitian

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tertentu. Secara umum, sistem pakar (*expert sistem atau* ES) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan pencarian keputusan suatu masalah seperti yang biasa dilakukan oleh para ahli.

Dalam melakukan pengambilan keputusan maka dapat dilakukan dengan menggunakan metode pencarian yaitu *Depth First Search dan Breadth First Search (BFS).* Metode *Depth First Search* dilakukan dengan node awal secara mendalam hingg**a** yang paling akhir *(dead-end)* sampai ditemukan *goal state*. Dengan kata lain, simpul cabang anak yang terlebih dahulu dikunjungi. Andaikan tujuan yang diinginkan belum tercapai maka pencarian dilanjutkan ke cabang berikutnya hingga semua node di kunjungi. Sedangkan Metode *Breadth First Search (BFS)* merupakan pencarian yng dilakukan dengan mengunjungi tiap-tiap node secara sistematis pada setiap level hingga keadaan tujuan (goal state) ditemukan. Dengan kata lain menggunjungi terlebih dahulu semua node yang selevel hingga ditemukan goal statenya.

Depth First Search (DFS)

Depth First Search (DFS) adalah algoritma pencarian simpul dalam graf secara *travelsal* yang dimulai dari simpul akar dan mengecek simpul anaknya yang pertama, setelah itu, algoritma mengecek simpul anak dari simpul anak yang pertama tersebut, hingga mencapai simpul daun atau simpul tujuan. Jika solusi belum ditemukan, algoritma melakukan runut balik *(backtracking)* ke simpul orangtuanya yang paling baru diperiksa lalu dan mengecek simpul anaknya yang belum diperika, sedemikian seterusnya hingga simpul solusi ditemukan. *Depth First Search* jauh lebih efesien untuk ruang pencarian dengan banyak percabangan, karena tidak perlu mengefaluasi semua simpul pada tingkat tertentu pada saat dimulainya pencarian. *(Wikipedia,(2007)), Depth-First Search.*

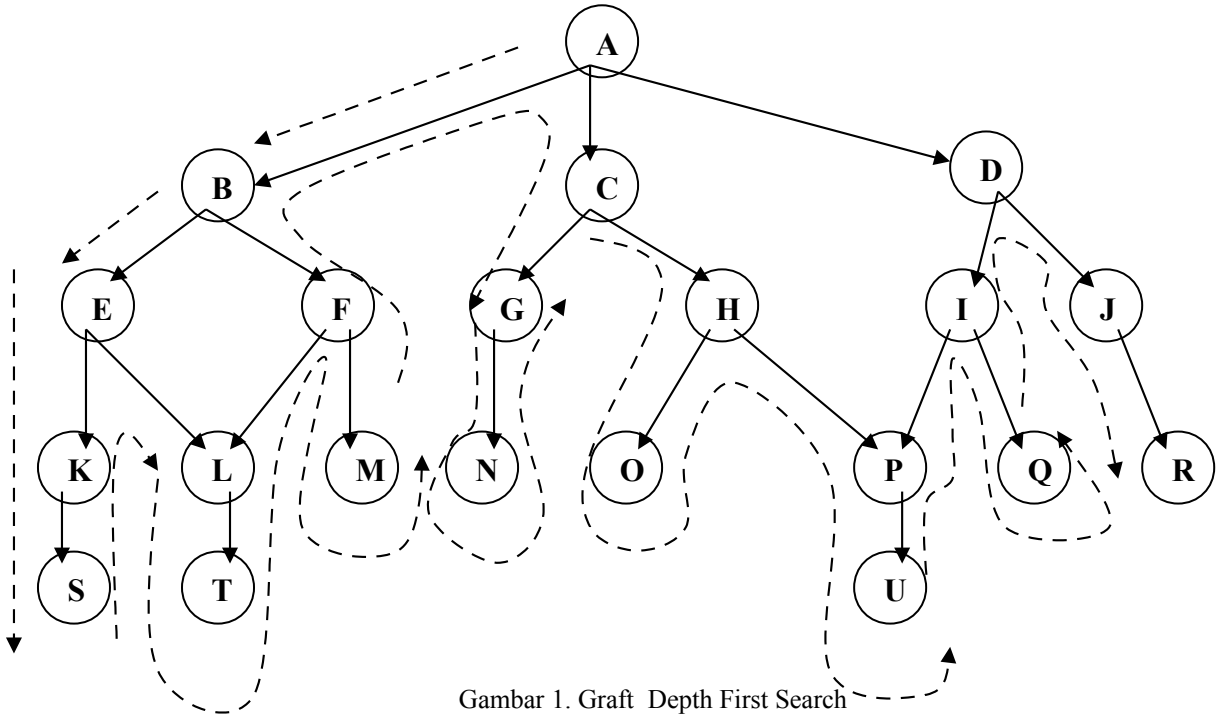
Depth First Search mempunyai keuntungan dan kelemahan :

a. Keuntungan dari *Depth First Search*
    1. Membutuhkan memori yang relatif kecil, karena hanya node-node pada lintasan yang aktif yang di simpan.
    2. Secara kebetulan, metode *Depth First Searh* akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.

b. Kelemahan dari *Depth First Search*.

1. Memungkinkan tidak ditemukannya tujuan yang diharapkan.
2. Hanya akan mendapatkan satu solusi pada setiap pencarian.

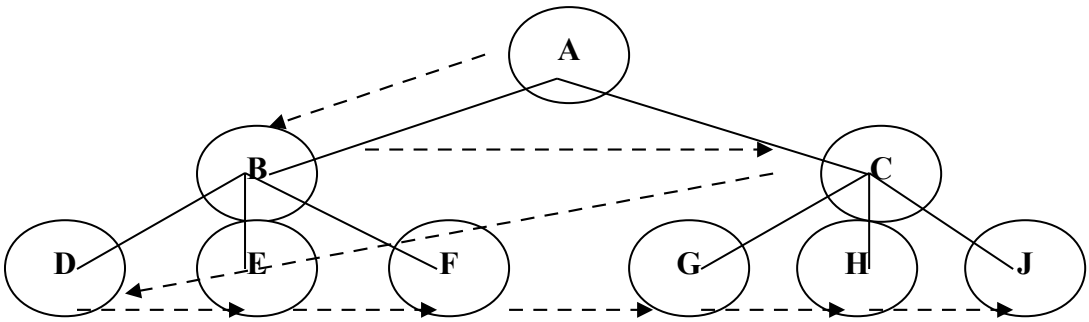
Untuk memeriksa algoritma diatas, bahwa keadaan-keadaan turunan (descendent) ditambahkan atau dihapuskan dari ujung kiri open. Hasil dari penerapan algoritma tersebut dapat dilihat seperti yang terlihat pada Gambar 1.



Gambar 1. Graft Depth First Search
Sumber: Elearning.uin-suka.ac.id/attachment/pencarian_heuristik_bjyr6.pdf

Breadth First Search (BFS)

Breadth First Search adalah algoritma pencarian simpul dalam graf (pohon) secara travelsal yang dimulai dari simpul akar dan mengecek semua simpul-simpul tetangganya. Setelah itu, dari tiap simpul tetangganya, algoritma akan terus mencek semua simpul tetangganya yang belum dicek, sedemikian seterusnya hingga menemukan simpul tujuan *Breadt First Search. Interpreter* kaidah mulai dari fakta yang ada yaitu hipotesa kemudian kaidah bagian *THEN* mulai di uji untuk mendukung hipotesa awal. Jika ditemukan maka kaidah *IF* yang cocok digunakan untuk menghasilkan hipotesa antara yang baru. Kemudian proses berantai terus di ulang, mengumpulkan bukti yang mendukung, sehingga hipotesa terbukti kebenarannya. *(Wikipedia (2007)),Breadth First Search.*
Caranya dapat dibuktikan seperti pada Gambar 2.



Gambar 2 *Graft Breadth First Search*
Sumber : E-Learning.uin-suka.ac.id/attachement/pencarian/heuristic_bjyr6.pdf

Breadth First Search merupakan proses penalaran dengan pendekatan proses *goal-driven.* Memulai titik pendekatanya dari *goal* yang akan dicari nilainya kemudian bergerak mencari informasi yang mendukung *goal* tersebut.
Keuntungan dari metode ini :

1. Tidak menemui jalan buntu.
2. Jika ada suatu solusi, maka Breadth-first search akan menemukannya. Dan jika didapat lebih dari satu solusi, maka solusi minimum akan ditemukan

Kelemahan metode ini :

TABLE I. Membutuhkan memori yang cukup banyak, karena menyimpan semua node dalam suatu pohon.

TABLE II. Membutuhkan waktu yang cukup lama, karena akan menguji n level untuk mendapatkan solusi pada level ke- (n + 1). (Kusumadewi. S, 2002).

Desain Arsitektur Sistem

Knowledge base berisi himpunan aturan atau rule-rule untuk mencari aturan, mencari macam-macam kerusakan, jenis-jenis kerusakan, gejala kerusakan, ciri-ciri kerusakan dan diagnosa kerusakan.

Contoh aturan-aturan adalah sebagai berikut :

IF Audio Rusak
AND IC Rusak
AND Kerusakan Hardware
AND Ciri kerusakan Sekering terbakar
THEN Jenis Kerusakannya: Panaskan dengan solder agar kaki IC menempel dengan baik (Audio Rusak). Ganti dengan yang baru jika tidak bisa (audio rusak). Ganti dengan komponen yang baru (IC Rusak).

Aturan atau rule diatas menunjukkan bahwa *JIKA* macam kerusakan adalah pada bagian Audio *DAN* jenis kerusakan adalah IC PS Rusak *DAN* Kerusakan terjadi juga pada *Hardware DAN* Kerusakan terjadi pula pada software *MAKA* Panaskan dengan solder agar kaki *IC* menempel dengan baik. Ganti dengan yang baru jika tidak bisa (Audio Rusak). Ganti dengan komponen yang baru.

Desain Database

Pengembangan sistem dilakukan untuk mengetahui Informasi secara detail kebutuhan pengguna. Database ini digunakan untuk menyimpan data dan informasi yang diperlukan dalam hal penggunaan sistem.

Database ini berisikan tentang fakta-fakta yang dibutuhkan dalam pemakaian data-data berupa tabel kerusakan , tabel gejala, tabel solusi. Adapun tabel database masing-masing adalah :

Tabel 1. Tabel Gejala

No	Field_Nama	Type	Size	Keterangan
	Kode_Kerusakan	Text	5	Kode Kerusakan
	Kode_Gejala	Text	5	Kode Gejala
	Nama_Gejala	Text	255	Nama Gejala

Pada tabel 1 ini berisikan tiga kategori yaitu Kode_Kerusakan, Kode_Gejala, Nama_Gejala yang diinput sesuai dengan ketentuan pada database.

Tabel 2. Tabel Kerusakan

No	Field_Nama	Type	Size	Keterangan
	Kode_Kerusakan	Text	5	Kode Kerusakan
	Nama_Kerusakan	Text	255	Nama Kerusakan

Pada tabel 2 ini database ini berisi informasi tentang tabel kerusakan yang berisi Kode_kerusakan dan Nama_kerusakan. Semua informasi tentang kerusakan akan disimpan pada database ini yang bernama tbl_kerusakan.dbf.

Tabel 3. Tabel Solusi

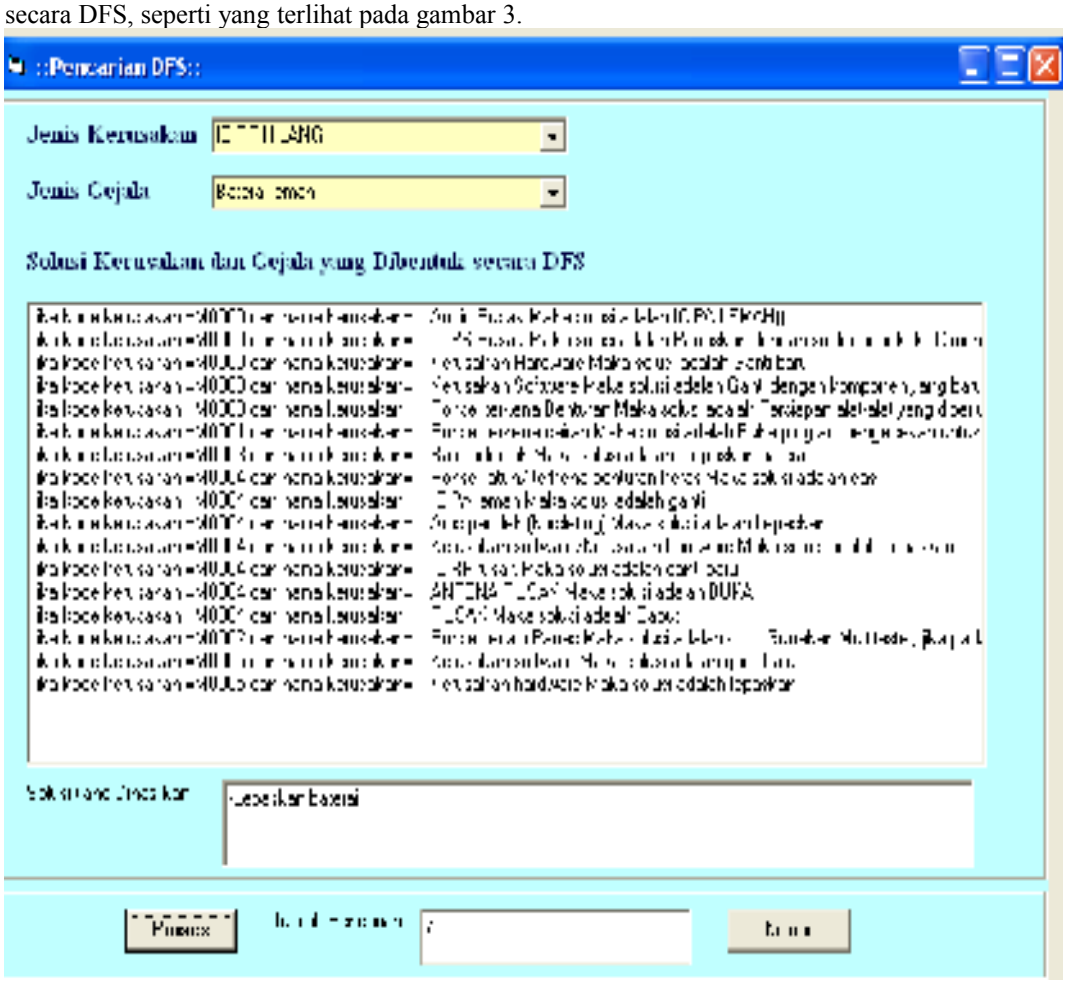
No	Field_Nama	Type	Size	Keterangan

	Kode_Gejala	Text	5	Kode Gejala
	Kode_Solusi	Text	5	Kode Solusi
	Nama_Solusi	Text	255	Nama Solusi

Tabel 3 ini berisi semua informasi tentang solusi dari proses yang terjadi setelah sistem dijalankan. Data yang tersimpan di tabel ini adalah Kode_Gejala, Kode_Solusi, Nama_Solusi.

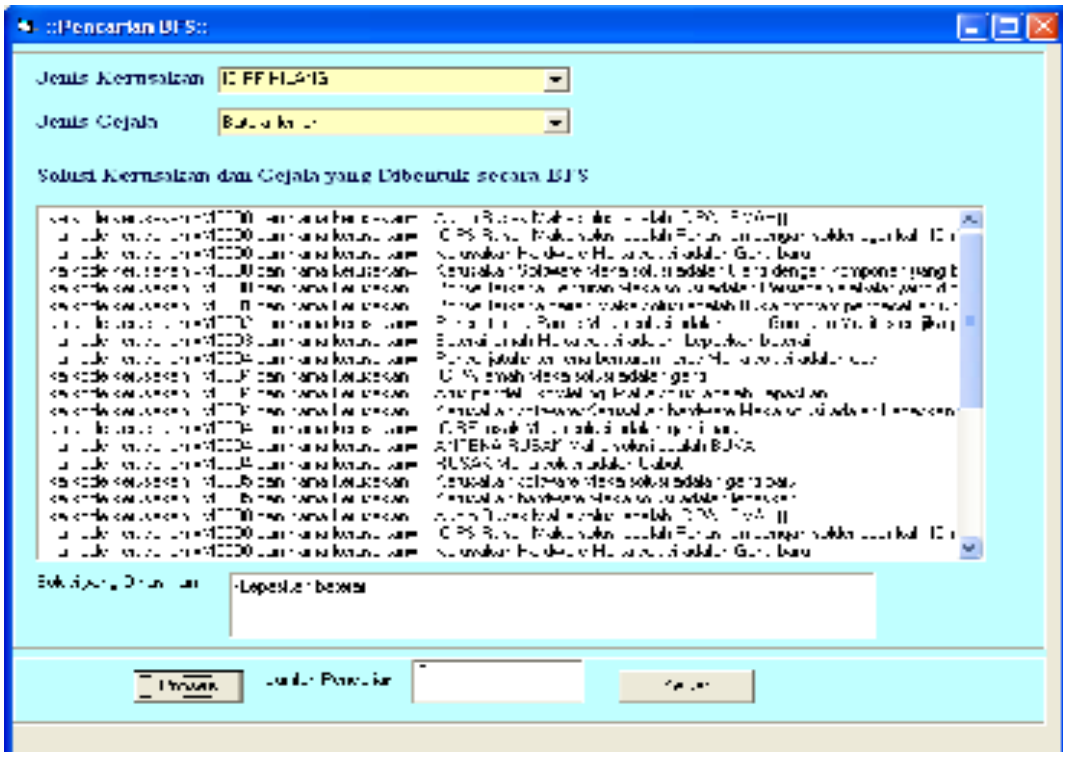
4. Hasil Program

Pertama – tama *user* harus menginputkan jenis kerusakan dan jenis gejala seperti untuk dicari secara DFS, seperti yang terlihat pada gambar 3.



Gambar 3. Solusi Kerusakan dengan DFS

Kemudian cara yang sama dapat digunakan untuk menghasilkan solusi dari metode BFS seperti tampilan gambar berikut ini :



Gambar 4 Hasil Proses BFS

Perhatikan gambar 3 dan 4, kita dapat melihat adanya perbedaan proses pencarian solusi yang dihasilkan antara metode DFS dan BFS. Perbedaan yang nampak adalah pada proses pencarian kerusakan. Di mana pada metode DFS pohon keputusan yang dibentuk adalah M0000, M0002, M0003, M0004, M0002, dan M00005 sedangkan oada metode BFS, pohon keputusan yang terbentuk adalah M0000,M0001, M0002, M0003, M0004, dan M0005. Berdasarkan perbedaan pohon keputusan yang dihasilkan maka untuk mencari kerusakan yang sama, seperti yang ditunjukkan pada gambar 3 metode DFS butuh 7 proses dan metode BFS butuh 8 proses.

Dari hasil eksperimen yang didapat, kedua algoritma baik BFS maupun DFS dapat menemukan solusi. Akan tetapi dalam hal performa, jelas untuk algoritma DFS lebih baik daripada BFS, tetapi untuk algoritma BFS jauh mengungguli algoritma DFS.

Hal ini diakibatkan oleh karena BFS selalu membandingkan simpul-simpul yang bertetanggaan, sehingga untuk rute perjalanan yang memutar BFS terpaksa memeriksa semua kemungkinan yang ada. Di lain persoalan, algoritma BFS memiliki keunggulan jika lokasi simpul solusi berada di dekat simpul akar karena algoritma dapat menemukan solusi sebelum membentuk pohon yang lebar.

BFS dalam kenyataannya memerlukan memori yang lebih besar, karena struktur data antrian yang digunakan harus menyimpan banyak informasi simpul yang bertumbuh secara eksponensial, sehingga untuk peta yang besar, jika posisi tujuan yang ingin dicapai cukup jauh, algoritma ini akan sangat menghabiskan memori. Akan tetapi BFS tidak akan terjebak dalam rute yang salah dan akan selalu mendapatkan solusi terbaik (*shortest path*). Jadi, algoritma BFS cocok diterapkan jika posisi yang dituju berada dalam jarak yang tidak terlalu jauh dari posisi awal.

Di pihak lain, DFS yang langsung memeriksa semua simpul anak pertama (dalam hal ini urutan pengecekan anak adalah selatan-timur-utara-barat) hingga ditemukan simpul solusi mendapat keuntungan karena simpul solusi selalu berada di dalam pohon anak pertama sehingga tidak terjadi *backtracking*. Akan tetapi DFS mengalami jalan buntu setelah melakukan pengecekan hingga simpul yang jauh bahkan menghasilkan solusi yang tidak optimal.

Dari segi pemakaian memori, DFS cenderung memerlukan kapasitas memori yang lebih kecil dibandingkan dengan BFS karena struktur data stack hanya akan menyimpan informasi simpul-simpul yang berada dalam rute yang sedang dicek. Akan tetapi algoritma ini sangat sulit dalam menemukan solusi optimal dan dapat terjebak dalam rute yang salah. Jadi, algoritma DFS cocok digunakan untuk permasalahan yang memiliki banyak solusi (simpul tujuan) tanpa harus memperhatikan solusi optimal,

sehingga untuk kasus pencarian rute perjalanan objek dua dimensi ini, DFS tidak cocok digunakan (simpul tujuan selalu hanya ada satu).

4. Kesimpulan dan Saran

4.1 Kesimpulan

Dari pembahasan sampai pada implementasi dari bab-bab dan sub bab sebelumnya maka diperoleh sebuah kesimpulan sebagai berikut :

[1] Sistem pakar dapat membantu produktifitas kerja dengan memungkinkan para teknisi *handphone* yang kurang berpengalaman dapat menelusuri kerusakan pada telepon selular serta memperbaikinya serta meminimalisir resiko dalam perbaikan.

[2] Sistem pakar ini juga dapat membantu yang bukan ahli, dengan memperoleh informasi serta cara mencegah dan mengatasi gangguan – gangguan yang terjadi dalam menelusuri kerusakan pada telepon selular.

[3] Membantu para teknisi dalam mengambil keputusan yang tepat tanpa merujuk kepada teknisi pakar yang mungkin sulit dihubungi atau dijumpai.

4.2 Saran

Untuk meningkatkan kehandalan perancangan sistem pakar dalam menelusuri kerusakan pada telepon selular ini penulis memberikan beberapa saran sebagai berikut:

1. Diharapkan adanya penelitian lanjutan dari sistem pakar ini sehingga mendapatkan hasil yang lebih baik dan sempurna.
2. Hendaknya hasil dari sistem pakar untuk menelusuri kerusakan pada telepon selular ini dapat dicetak, sehingga memudahkan pengguna dalam mengimplementasikan solusi yang akan diberikan oleh sistem.
3. Sistem pakar ini mempunyai keterbatasan, sehingga harus selalau di-*update* karena pengetahuan dari sang pakar akan dinamis dan berkembang sehingga sistem pakar tidak bisa menggantikan pakar sepenuhnya

Referensi

Buku Teks:

[1] David Steven Wijaya. *Perbandingan Algoritma BFS Dan DFSs Dalam Pembuatan Rute Perjalanan Objek Permainan 2 Dimensi.*.

[2] Darianto. *Pengetahuan dasar ilmu komputer.Bandung:* Yrama Widya. 2003.

[3] M. Arhami. *Konsep dasar Sistem pakar*. Yogyakarta : Penerbit Andi. 2005.

[4] Sandy S. *Artificial Intelligence*. Yogyakarta : Andy offset. 1993.

[5] Sony Daniswara, Ryan. *Mencari dan memperbaiki Handphone*. Depok: kawan pustaka. 2005.

[6] Turban. *Logika Fuzzy, jaringan saraf tiruan dan Robotika*. Yogyakarta : Andy offset. 1995.

[7] Negnevitsky. *Perbandingan Antara Sistem Pakar, Sistem Konvensional Dan Kepakaran Manusia. Jakarta :* Komputindo. 2002.

Buku yang diedit:

[1] Martin, Oxman. Editors. *expert sistem. Yogyakarta :* Andy offset. 1989.