

IMPLEMENTASI ALGORITMA BREADTH FIRST SEARCH DAN *STRING* MATCHING PADA SISTEM PAKAR

Fajar J. Ekaputra¹, Windarto Harimurti², Aqsa Adhiperwira³

*Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung*

E-mail : if13079@students.if.itb.ac.id¹, if13089@students.if.itb.ac.id²,
if13090@students.if.itb.ac.id³

Abstrak

Dalam mendapatkan sebuah solusi yang paling tepat untuk memecahkan berbagai permasalahan yang dihadapi, dibutuhkan informasi-informasi yang berhubungan dengan permasalahan tersebut, agar solusi yang dihasilkan merupakan solusi yang terbaik untuk diimplementasikan pada permasalahan tersebut. Dalam kehidupan nyata, kita dapat menggunakan jasa konsultan untuk membantu kita dalam menyelesaikan permasalahan yang dihadapi, yaitu membantu kita dalam memberikan alternatif solusi dari himpunan solusi yang ada, sehingga pilihan solusi tersebut merupakan pilihan solusi yang paling merepresentasikan jawaban dari permasalahan yang sedang dihadapi. Jasa konsultasi seperti itu, dapat diimplementasikan dalam bentuk sebuah aplikasi yang dapat memproses informasi-informasi yang telah diberikan, kemudian dapat menghasilkan sebuah himpunan solusi permasalahan yang paling tepat untuk diimplementasikan. Aplikasi tersebut dapat dikembangkan dengan memanfaatkan berbagai algoritma, yaitu algoritma Bread First Search dan *String* Matching.

Kata kunci: Aplikasi, Breadth First Search, String Matching, Sistem Pakar

1. Pendahuluan

Ketika dihadapkan pada suatu problematika tertentu, seseorang tentu ingin solusi pemecahan masalah tersebut memiliki spesifikasi yang sesuai dengan keinginan dari orang tersebut. Biasanya seseorang berkonsultasi dengan seorang ahli atau Expert di bidang problematika tersebut.

Sesuai dengan kemajuan teknologi komputasi dan algoritma, kini aplikasi komputer mampu memiliki kemampuan untuk berpikir sendiri dan memberikan solusi terbaik dalam domain permasalahan tertentu sesuai dengan spesifikasi yang diberikan oleh pengguna. Aplikasi tersebut dinamakan Expert System atau Sistem Pakar

Kemampuan Sistem pakar tersebut diperoleh dengan implementasi suatu algoritma yang membandingkan antara masukkan spesifikasi dari pengguna dengan karakteristik dari berbagai macam himpunan solusi yang dimiliki oleh sistem ahli. Algoritma yang dipakai dalam sistem ahli tersebut adalah Algoritma Breadth First Search dan *String* Matching

2. Ruang Lingkup

Ruang lingkup tugas yang kami kerjakan mencakup penjelasan dari semua hal yang terdapat didalam pengembangan sistem pakar. Diantaranya termasuk sistem pakar itu sendiri, algoritma BFS dan *String*

Matching. Kemudian setelahnya akan diperbandingkan hasil dari kedua algoritma tersebut untuk memperoleh algoritma terbaik untuk permasalahan ini.

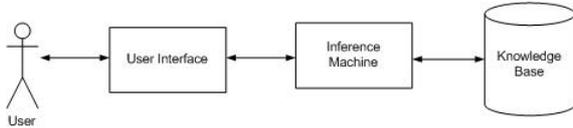
3. Sistem Pakar

Sistem pakar merupakan implementasi program komputer yang terdiri dari kumpulan aturan yang digunakan untuk menganalisis informasi tentang suatu permasalahan yang spesifik. Informasi yang dimiliki pada umumnya diberikan oleh pakar suatu permasalahan yang berperan sebagai pengguna sistem. Tujuan sistem pakar adalah menghasilkan solusi yang tepat untuk menjawab suatu permasalahan. [<http://wikipedia.org>].

Sistem pakar mampu bertindak layaknya seorang manusia, dapat menganalisis permasalahan, mempelajari, kemudian membuat keputusan untuk memecahkan permasalahan yang dihadapi dalam suatu domain permasalahan tertentu. Sistem pakar kedokteran sebagai implementasi sistem pakar dengan domain permasalahan di bidang kedokteran adalah salah satu contoh penerapan sistem pakar.

Sistem pakar merupakan kesatuan sistem yang terdiri dari *knowledge base* (basis pengetahuan), mesin inferensi, dan antarmuka. Basis pengetahuan berfungsi menyimpan data berupa *facts* (fakta-fakta) dan *rules* (aturan-aturan). Mesin inferensi merupakan

implementasi algoritma pemrograman yang bertugas menganalisis *query* (pertanyaan) dari pengguna sistem, memanfaatkan basis pengetahuan yang dimilikinya, kemudian memproses jawaban sehingga didapatkan solusi yang tepat. Antarmuka yang menghubungkan sistem pakar dengan pengguna sistem, bertugas menerima pertanyaan-pertanyaan dari pengguna, kemudian menampilkan solusi dari pertanyaan tersebut.



Fakta-fakta yang terdapat pada basis pengetahuan merupakan informasi-informasi yang diberikan oleh seorang pakar. Aturan-aturan pada basis pengetahuan berfungsi menganalisis pertanyaan-pertanyaan yang diberikan, apakah sesuai dengan informasi-informasi yang dimiliki.

Proses analisis yang dilakukan terhadap pertanyaan-pertanyaan yang diberikan, dapat menggunakan beberapa pilihan algoritma pencarian. *Bread First Search* (BFS) ataupun *String Matching*, dapat digunakan untuk mencari dan mencocokkan pertanyaan yang diberikan oleh pengguna sistem dengan informasi-informasi yang terdapat pada basis pengetahuan.

4. Algoritma Breadth First Search

4.1 Pendahuluan

Algoritma Breadth First Search adalah sebuah algoritma yang digunakan dalam mencari suatu solusi dalam himpunan solusi yang direpresentasikan dalam bentuk Pohon. Pohon disini bisa dibentuk secara dinamis maupun static.

4.2 Sistem Kerja Algoritma

Pada Algoritma ini, Pencarian dilakukan pada Akar Pohon. Akar dibaca dan dimasukkan ke dalam sebuah Antrian dan dibandingkan apakah sudah sesuai dengan solusi. Jika tidak, maka Pencarian dilanjutkan kepada Simpul yang bertetangga dengan Simpul akar namun belum dikunjungi selama pencarian, dimana simpul tersebut akan dimasukkan ke antrian dan akan dibandingkan kembali. Proses dilanjutkan sampai Solusi ditemukan atau semua simpul pada pohon dikunjungi namun solusi belum ditemukan.

4.3 Implementasi

Sistem pakar mampu memberikan solusi dari problematika pengguna, oleh karena itu pada system pakar diimplementasikan prosedur untuk menampilkan pertanyaan, dari pertanyaan tersebut pengguna akan memberikan jawaban. Prosedur pertanyaan itu merupakan representasi dari inference machine pada sistem pakar.

Pada sistem pakar tersebut juga diimplementasikan himpunan solusi yang sudah memiliki karakteristik tertentu. Setiap elemen dari himpunan solusi memiliki himpunan karakteristik lebih dari satu. Himpunan solusi disini merupakan representasi dari basis pengetahuan pada sistem pakar

Aplikasi akan meng-interpretasikan jawaban yang diberikan oleh pengguna sebagai salah satu spesifikasi dari solusi, dan akan menyeleksi Himpunan Solusi dengan sesuai dengan spesifikasi pengguna dengan algoritma BFS

Pada Algoritma ini, yang harus dilakukan adalah membentuk suatu pohon yang setiap simpulnya adalah salah satu karakteristik dari himpunan solusi. Pembentukan pohon pada algoritma ini dengan mengambil salah satu elemen dari himpunan karakteristik dalam Himpunan solusi, kemudian dari Tipe karakteristik tersebut akan dicari jenis-jenis karakteristik, jenis karakteristik inilah yang akan dijadikan simpul anak. Simpul anak tersebut juga akan membentuk simpul anak dibawahnya dengan cara mengambil tipe karakteristik yang masih ada pada himpunan karakteristik, dan kemudian akan menggunakannya untuk mencari jenis karakteristiknya . Cara tersebut dilakukan sampai himpunan karakteristik sudah kosong.

Metode pencarian solusi itu dilakukan dengan cara menyusuri pohon solusi yang terbentuk. Ketika pengguna memberikan spesifikasi, sistem pakar akan memberikan interpretasi jawaban. Interpretasi tersebut akan dibandingkan terhadap Pohon solusi yang sudah terbentuk. Pertama akar akan membandingkan hasil interpretasi dengan semua Simpul anak yang bertetangga, jika sesuai dengan salah satu simpul, maka simpul yang lain akan dimatikan. Jika pengguna memasukkan spesifikasi yang lain, hasil interpretasi sistem pakar akan dibandingkan dengan anak pohon dari simpul yang hidup. Proses tersebut akan berjalan sampai pengguna berhenti memasukkan spesifikasi atau Pohon solusi sudah sampai pada anak pohon yang terakhir. Hasil dari pohon solusi aka disimpam pada suatu variabel dan akan dibandingkan dengan Himpunan solusi dari sistem pakar. Himpunan solusi akan mengeluarkan solusi berdasarkan pada input pengguna terhadap Pohon Solusi sistem pakar.

4.4 Pseudo Code

```

S = {Himpunan Solusi Sistem Pakar}
C = {Himpunan Tipe Karakteristik dari himpunan
solusi}
SC(elemen dari C) = {Himpunan Jenis
karakteristik berdasarkan tipe karakteristik
tertentu}

//Prosedur membuat Pohon
do {
  Ctemp ← {C};

```

```

Delete Ctemp from {C};
do{
    Sctemp ← {SC(Ctemp)};
If (NotElemenPohon(Sctemp)) then
    BuatSimpulAnak(Sctemp);
}
while( Sctemp = {});
}
while(Ctemp = {});

// Prosedur Pencarian Solusi

BeginFromAkar(); // mulai dari akar
Interpretation(Hasil);
// sistem pakar memberika interpretasi dari
jawaban pengguna dan disimpan di hasil

boolean ketemu = false;

while not ketemu do {
    if IsSimpulAnak(Hasil) then
    {
        DeleteAkarLama();
        //menghapus akar lama
        BuatAkarBaru();
        //membuat akar baru pada simpul anak
        HapusSimpulAnakLain();
        //menghapus simpul anak yang lain yang
bertetangga dengan Akar lama
    }
    else
    {
        CariSimpulAnak();
        //mencari simpul anak yang lain
    }
}

```

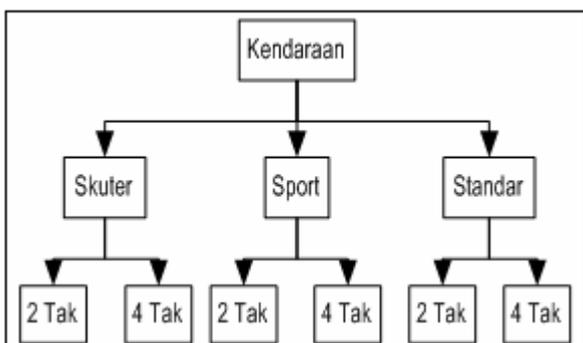
4.5 Contoh Algoritma

Sebuah Sistem Pakar Kendaraan Roda Dua, memiliki Himpunan Solusi :

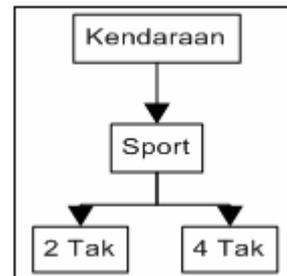
- Kymco Free (Skuter, 2 tak)
- Kymco Jetmatic (Skuter, 4 tak)
- Kawasaki Ninja (Sport, 4 tak)
- Kawasaki Ninja RR (Sport, 2 tak)
- Honda Supra Fit (Standar, 2 tak)
- Honda Supra X125 (Standar, 4 tak)

Tipe karakteristik di atas adalah Jenis Motor dan Teknologi Motor. Jenis Karakteristik dari Tipe Jenis Motor adalah Skuter, Sport, dan Standar. Jenis karakteristik dari Teknologi motor adalah 2 tak dan 4 tak.

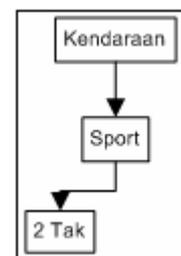
Berdasarkan karakteristik dari himpunan solusi itu maka dapat terbentuk suatu Pohon Solusi seperti :



Jika Pengguna memasukkan spesifikasi dan sistem pakar meng-interpretasikan sebagai Sebuah Motor Sport, maka pohon solusi akan membentuk seperti :



maka pada giliran berikutnya sistem pakar akan mengarahkan pertanyaan pengguna kepada pilihan “2 Tak” atau “4 Tak”. Jika ternyata pengguna memasukkan pilihan 2 Tak maka pohon yang terbentuk.



Sistem Pakar akan mengeluarkan sebuah solusi berupa
 ➤ Kawasaki Ninja RR (Sport, 2 tak)

5. Algoritma String Matching

5.1 Pendahuluan

Algoritma *String Matching* adalah sebuah algoritma yang digunakan dalam pencocokkan suatu pola kata tertentu terhadap suatu kalimat atau teks panjang. Algoritma *String Matching* sendiri dapat dilakukan dengan beberapa cara tertentu, antara lain cara Lempang dan cara Knuth, Morris, Pratt (KMP).

5.2 Sistem Kerja Algoritma

Cara Lempang (Brute Force) dilakukan dengan membandingkan seluruh elemen karakter pada pola dengan kalimat atau teks panjang, dimulai pada elemen karakter pertama pada kalimat tersebut. Jika tidak sesuai maka perbandingan dimulai dengan elemen kedua dari kalimat tersebut.

Cara KMP dilakukan dengan menghitung fungsi pinggiran dari pola terlebih dulu dan kemudian akan dilakukan perbandingan antara pola dan elemen pertama dari kalimat, jika tidak sesuai, maka perbandingan tidak dilakukan pada elemen kedua, namun tergantung dari nilai yang akan dikeluarkan oleh fungsi pinggiran tersebut. pada Jurnal ini, tidak dibahas lebih lanjut mengenai cara ini, karena yang akan kita gunakan adalah cara lempang

5.3 Implementasi

Implementasi hampir sama seperti ketika dengan implementasi dengan Algoritma Breadth First Search. Sistem pakar memiliki prosedur untuk menghasilkan pertanyaan, yang akan dijawab oleh pengguna. Jawaban pengguna ini akan diinterpretasikan oleh sistem pakar sebagai spesifikasi dari solusi yang diinginkan oleh pengguna.

Sistem pakar juga memiliki himpunan solusi yang memiliki karakteristik tertentu. Dimana setiap elemen dari himpunan solusi memiliki karakteristik lebih dari satu.

Pada Algoritma ini, jawaban dari pengguna akan digolongkan menjadi tipe karakteristik oleh sistem pakar. Dari tipe karakteristik tersebut, maka akan dicari jenis karakteristik, umumnya disimpan dalam sebuah format *string*, yang sesuai dengan interpretasi jawaban pengguna oleh sistem pakar. Jika jenis karakteristik cocok dengan interpretasi sistem pakar, maka elemen Solusi tersebut akan disimpan dalam sebuah himpunan solusi sementara 1, proses pencocokan itu akan berjalan sampai semua elemen himpunan solusi sudah dicocokkan. Jika pengguna memasukkan spesifikasi yang lain, maka sistem pakar akan menggolongkannya terlebih dahulu menjadi tipe karakteristik tertentu untuk kemudian mencari jenis karakteristik yang sesuai dengan interpretasi sistem pakar dari input pengguna. Seluruh elemen Solusi yang memiliki kecocokan *string* karakteristik dengan interpretasi sistem pakar akan dimasukkan ke dalam himpunan solusi sementara 2. Sistem pakar akan melakukan pengirisan (\cap) terhadap Himpunan solusi sementara 1 dan himpunan solusi sementara 2. Hasil dari pengirisan akan dimasukkan kepada variabel Solusi final. Jika pengguna memasukkan spesifikasi lagi, maka proses tersebut akan berulang, sampai pengguna berhenti untuk memasukkan spesifikasi. Solusi yang dihasilkan adalah solusi yang terdapat pada variabel Solusi Final.

5.4 Contoh Algoritma

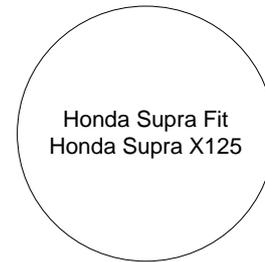
Sebuah Sistem Pakar Kendaraan Roda Dua, memiliki Himpunan Solusi :

- Kymco Free (Skuter, 2 tak)
- Kymco Jetmatic (Skuter, 4 tak)
- Kawasaki Ninja (Sport, 4 tak)
- Kawasaki Ninja RR (Sport, 2 tak)
- Honda Supra Fit (Standar, 2 tak)
- Honda Supra X125 (Standar, 4 tak)

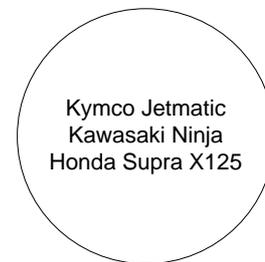
Tipe karakteristik di atas adalah Jenis Motor dan Teknologi Motor. Jenis Karakteristik dari Tipe Jenis Motor adalah Skuter, Sport, dan Standar. Jenis karakteristik dari Teknologi motor adalah 2 tak dan 4 tak.

Ketika pengguna memasukkan sebuah spesifikasi dan sistem pakar meng-interpretasikannya sebagai Motor

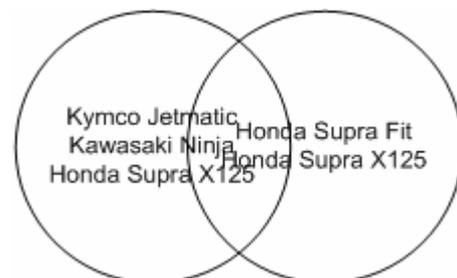
Standar, sehingga akan membentuk sebuah himpunan solusi sementara 1 berupa himpunan motor standar.



ketika pengguna memasukkan spesifikasi berikutnya dan system pakar meng-interpretasikannya sebagai sebuah motor yang memiliki system 4 tak. Maka akan terbentuk sebuah himpunan solusi sementara 2 yaitu himpunan motor 4 Tak.



system pakar kemudian akan mengiris himpunan solusi sementara 1 atau himpunan motor standar dengan himpunan solusi sementara 2 atau himpunan motor 4 tak, sehingga akan menghasilkan himpunan seperti :



dan akan menghasilkan sebuah solusi yaitu

- Honda Supra X125 (Standar, 4 tak)

6. Kesimpulan

Kesimpulan yang didapat dari uraian diatas adalah bahwa masing-masing algoritma penyelesaian ternyata memiliki kelebihan dan kekurangan masing-masing dalam penyelesaiannya.

Dalam sistem pakar menggunakan algoritma BFS, penggunaan basis data yang digunakan sebagai representasi dari knowledge base atau sistem pengetahuan akan diminimalisir, sehingga hanya akan mencakup basis data yang diperlukan saja. Mengapa? Hal itu terjadi karena setiap kali sebuah pertanyaan dijawab, jumlah solusi yang digunakan untuk

pertanyaan berikutnya akan semakin berkurang. Hal ini tidak berarti banyak jika basis data yang digunakan kecil, namun pengaruhnya akan terasa jika database yang digunakan sangat besar.

Penggunaan algoritma BFS juga memiliki kelemahan, yaitu bahwa pertanyaan yang diajukan harus berdasarkan urutan tertentu, dan juga, setiap pertanyaannya haruslah spesifik untuk jalur yang dilalui.

Berbeda dengan pada penggunaan algoritma *String Matching*, pencarian dilakukan dengan selalu mencocokkan setiap pertanyaan dengan semua anggota himpunan. Sehingga keseluruhan anggota himpunan akan diakses ketika setiap kali pertanyaan yang diajukan sistem pakar memperoleh jawaban.

Kebalikan dari algoritma BFS, pada algoritma *String Matching* urutan pertanyaan tidaklah penting, sehingga tidak perlu dilakukan pembuatan jalur-jalur pertanyaan seperti yang dilakukan pada algoritma BFS.

7. Daftar Pustaka

1. Russel, Stuart dan Norvig, Peter (2003), *Artifisial Intelegence, A Modern Aproach, Second Edition*, Prentice-Hall
2. Munir, Rinaldi (2005), *Diktat Strategi Algoritmik*, Informatika ITB
3. www.wikipedia.org