**METODE PENCARIAN *GENERATE AND TEST***

syarat untuk memenuhi tugas karya tulis ilmiah Kecerdasan Buatan



oleh

Rizki Wahyudi (55201120039)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN INFORMATIKA**

**UNIVERSITAS NURTANIO BANDUNG**

**KOTA BANDUNG**

**2022**

**BAB I**

**LANDASAN TEORI**

1. **Pengertian**

Beberapa dasar teori yang menjadi landasan penulisan, yaitu search, heuristik, dan algoritma *Generate and Test*.

1. *Search*

Menurut Luger (2005), *search* adalah sebuah teknik menyelesaikan masalah *(problem solving)* yang mengembangkan sebuah ruang permasalahan secara sistematik dalam sebuah proses. Terdapat 4 kriteria untuk menentukan performa sebuah metode pencarian, yaitu *Completeness, Time Complexity, Space Complexity, dan Optimality*. *Completeness* adalah apakah metode tersebut menjamin ditemukannya solusi jika solusi tersebut ada. *Time Complexity* adalah lama waktu yang dibutuhkan untuk menemukan solusi tersebut. *Space Complexity* adalah jumlah memori yang diperlukan dan yang dimaksud *Optimality* adalah apakah metode tersebut menjamin menemukan solusi yang terbaik jika terdapat beberapa solusi yang lain.

Metode pencarian dibagi menjadi dua strategi, yaitu *uninformed search* dan *informed search. Uninformed search* merupakan suatu strategi pencarian tanpa ada informasi mengenai *cost* (bobot) atau informasi tertentu sedangkan *Informed search* merupakan suatu strategi pencarian yang membutuhkan informasi mengenai *cost* (bobot) atau informasi tertentu.

1. Heuristik

Heuristik berfungsi untuk meningkatkan efisiensi dari sebuah proses pencarian. Tidak semua teknik heuristik memenuhi empat kriteria performa suatu metode pencarian. Fungsi heuristik digunakan untuk menghitung *path cost* dari suatu node awal menuju node tujuan. Fungsi heuristik yang digunakan dalam penelitian ini adalah *Straight-line distance* atau jarak secara garis lurus. Jarak yang digunakan dalam penelitian ini adalah jarak dan waktu sebenarnya yang diperoleh penulis dari Dinas Perhubungan Komunikasi dan Informatika.

1. Algoritma *Generate and Test*

Algoritma *Generate and Test* merupakan algoritma paling sederhana dalam teknik pencarian heuristik. Dalam *Generate and Test*, terdapat dua prosedur penting yaitu *generate* (membangkitkan) yaitu membangkitkan semua solusi yang mungkin dan *test* (pengetesan) yaitu menguji solusi yang dibangkitkan tersebut. Algoritma Generate and Test menggabungkan algoritma DFS dengan pelacakan mundur *(backtracking)*, yaitu bergerak ke belakang menuju state awal.

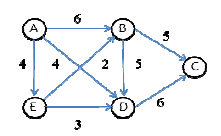
Algoritma *Generate and Test* adalah sebagai berikut :

1. Bangkitkan suatu kemungkinan solusi (membangkitkan suatu titik tertentu atau lintasan tertentu dari keadaan awal).
2. Menguji tiap-tiap node yang merupakan solusi dengan cara membandingkan node tersebut dengan node akhir dari suatu lintasan yang dipilih dengan kumpulan tujuan yang diharapkan.
3. Jika solusi telah ditemukan, maka keluar dari sistem. Jika belum menemukan solusi, maka kembali ke langkah pertama.

**BAB II**

**CONTOH PENERAPAN**

METODE *GENERATE AND TEST*

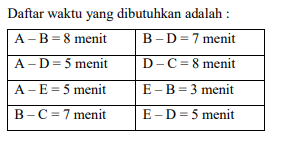


Pada gambar diatas terdapat contoh penyelesaian kasus algoritma *Generate and Test* dengan mengadaptasi situasi dan kondisi bus Trans Jogja.

Pencarian dilakukan dari titik A menuju titik C dengan daftar trayek :

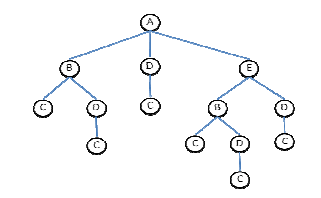
1. 1A melewati halte: A , D, dan C
2. 2A melewati halte : A, E, D, dan C
3. 1B melewati halte : B dan D
4. 2B melewati halte : A dan B
5. 3A melewati halte : B, C dan E

Waktu jeda 1A = 5 menit, 1B = 5 menit, 2A = 7 menit, 2B = 10 menit, dan 3A = 7 menit.

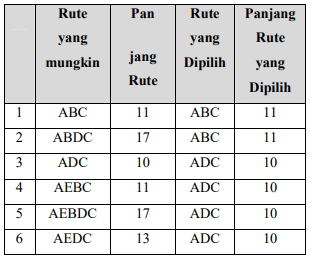


Berikut ini adalah langkah-langkah penyelesaian kasus di atas adalah :

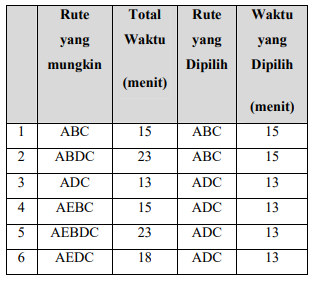
1. Pada gambar dibawah menjabarkan satu per satu kemungkinan yang ada.

****

1. Membuat daftar tabel beserta perhitungan jarak dan waktu untuk setiap jalur yang telah dilewati.

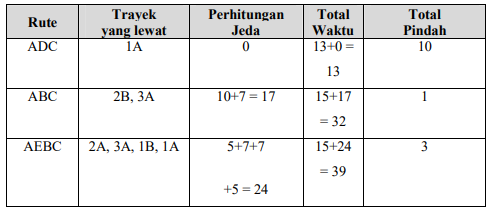
****

Tabel alur pencarian jarak



Tabel alur pencarian waktu

1. Dari Tabel pencarian jarak ditemukan rute terpendek berdasarkan jarak adalah A – D – C dengan total jarak 10 km. Rute alternatif 1 adalah A – B – C dengan total jarak 11 km. Rute alternatif 2 adalah A – E – B – C dengan total jarak 11 km.
2. Dari Tabel pencarian waktu ditemukan rute terpendek berdasarkan waktu adalah A – D – C dengan total waktu 13 menit. Rute alternatif 1 adalah A – B – C dengan total waktu 15 menit. Rute alternatif 2 adalah A – E – B – C dengan total waktu 15 menit.
3. Setelah melakukan perhitungan jarak dan waktu, langkah selanjutnya adalah menentukan trayek bus yang digunakan pada masing-masing rute. Tabel penetuan trayek menunjukkan penentuan trayek pada masing–masing rute.



Tabel penentuan trayek

Berdasarkan analisis perhitungan yang dilakukan maka rute terpendek berdasarkan jarak adalah A – D – C dengan total jarak tempuh 10 km tanpa perpindahan.

Berdasarkan analisis perhitungan yang dilakukan maka rute terpendek berdasarkan waktu adalah A – D – C dengan total waktu tempuh 13 menit tanpa perpindahan trayek.