

## BAB II. LANDASAN TEORI

### 2.1 Algoritma A\* (A Star)

Algoritma A\* pertama kali ditemukan pada tahun 1968 oleh Peter Hart, Nils Nilsson dan Bertram Raphael. Algoritma A\* merupakan salah satu dari heuristic search, adalah algoritma untuk mencari estimasi jalur dengan cost terkecil dari node awal ke node berikutnya sampai mencapai node tujuan. A\* memiliki suatu fungsi yang didenotasikan dengan  $f(x)$  untuk menetapkan estimasi cost yang terkecil dari jalur yang dilalui node  $x$  dengan rumus sebagai berikut.

$f(x) = h(x) + g(x)$  (1) Fungsi  $h(x)$  adalah hyphotesis cost atau heuristic cost atau estimasi cost terkecil dari node  $x$  ke tujuan, yang disebut juga sebagai future pathcost. Fungsi  $g(x)$  adalah geographical cost atau cost sebenarnya dari node  $x$  ke node tujuan, yang disebut juga sebagai past path-cost. Dengan metode atau algoritma A\*, cost untuk mencapai node berikutnya didapat dari fungsi  $f(x)$ , sehingga pada pemilihan jalur terpendek dapat langsung diketahui node berikutnya dengan cost terkecil sampai mencapai node tujuan tanpa kembali ke node yang sudah dikunjungi.[3] Dalam notasi matematika dituliskan sebagai :

$$f(n) = g(n) + h(n) \dots\dots\dots(2.1)$$

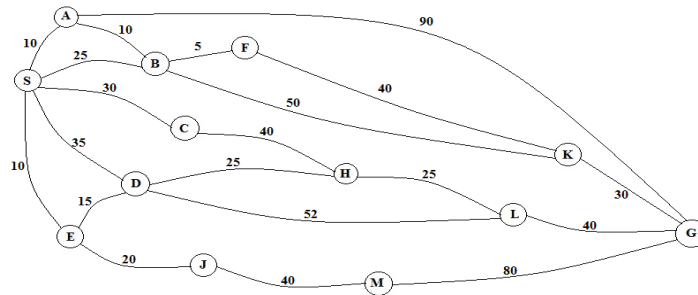
Dengan perhitungan biaya seperti ini, algoritma A\* adalah *complete* dan *optimal*.

$f(n)$  = jumlah dari  $g(n)$  dan  $h(n)$ . ini adalah perkiraan jalur terpendek sementara.

$g(n)$  = **Geographical Cost**, total jarak yang didapat dari node awal ke node sekarang.

$h(n)$  = **Heuristic Cost**, perkiraan jarak dari node sekarang ( yang sedang dikunjungi ) ke node tujuan.

Dengan perhitungan jarak seperti ini, algoritma A\* adalah *complete* dan optimal. Misalkan terdapat 13 kota yang dinyatakan oleh simpul-simpul dalam suatu *graph* dua arah. Setiap angka pada busur menyatakan jarak sebenarnya (*actual cost*) antara satu kota dengan yang lainnya. Nilai  $h(n)$  adalah fungsi heuristik, yaitu jarak garis lurus dari simpul  $n$  menuju simpul G dalam satuan kilometer. Contoh kasus pencarian rute terpendek ditampilkan pada Gambar 2.1



**Gambar 2.1 Contoh Kasus Algoritma A\***

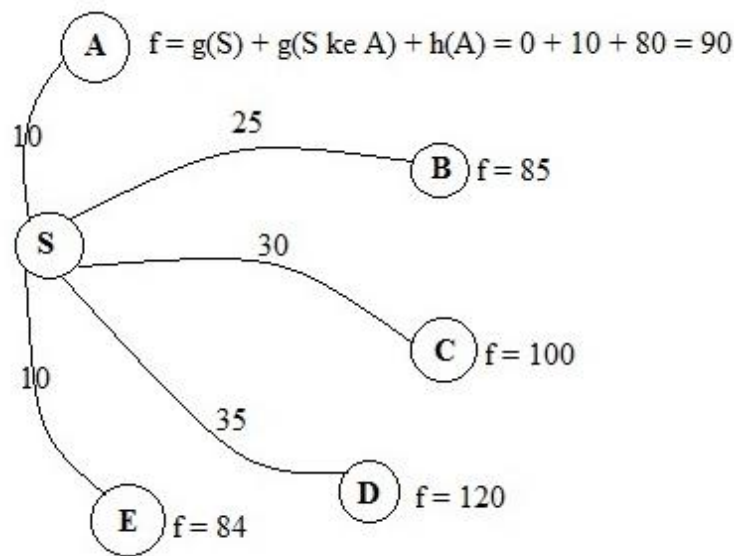
Tabel 2.1 adalah tabel nilai  $h(n)$  yang menyatakan jarak perkiraan dari setiap simpul-simpul menuju simpul G :

Tabel 2.1 Nilai  $h(n)$  contoh algoritma A\*

N	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

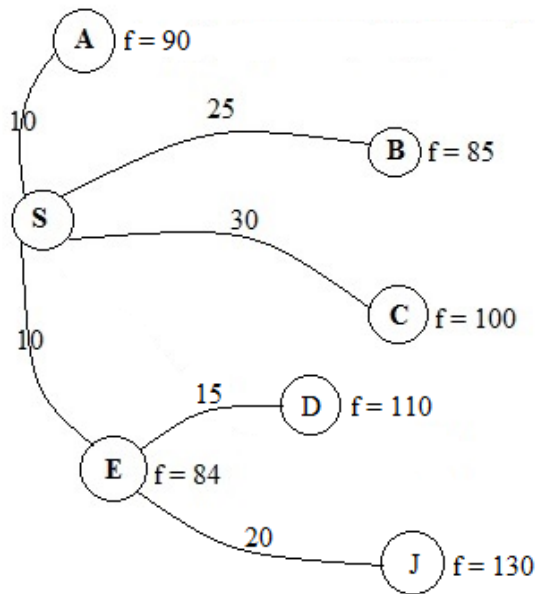
Langkah-langkah pencarian rute berdasarkan algoritma tersebut adalah sebagai berikut:

1. Langkah pertama, karena di *OPEN* hanya terdapat satu simpul (S), maka S terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Kemudian bangkitkan semua suksesor S, yaitu: A, B, C, D, E. Karena kelima suksesor tidak ada di *OPEN* maupun *CLOSED*, maka kelimanya dimasukkan ke *OPEN*. Menghasilkan *OPEN* = [A, B, C, E] dan *CLOSED* = [S]. Gambar 2.2 menjelaskan langkah pertama dari pencarian rute terpendek



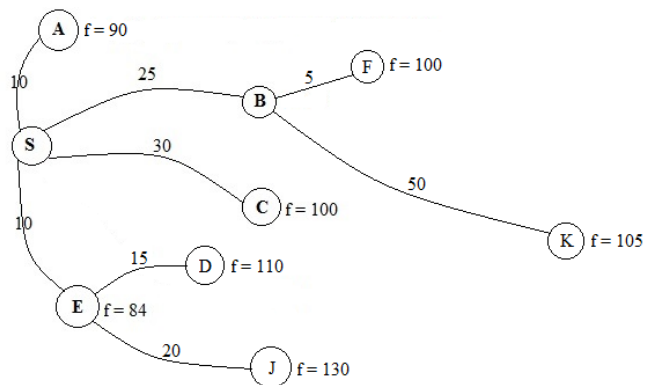
**Gambar 2.2 Langkah Pertama contoh kasus algoritma A\***

- Langkah kedua, E dengan biaya terkecil (yaitu 84) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor E dibangkitkan, yaitu: D dan J. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka J dimasukkan ke *OPEN*. Sedangkan simpul D sudah ada di *OPEN*, maka harus dicek apakah *parent* dari D perlu diganti atau tidak. Ternyata, jarak dari S ke D melalui E (yaitu  $10 + 15 = 25$ ) lebih kecil daripada jarak dari S ke D (yaitu 35). Oleh karena itu, *parent* dari D harus diubah, yang semula S menjadi E. Dengan perubahan *parent* ini, maka nilai *g* dan *f* pada D juga diperbarui (nilai *g* yang semula 35 menjadi 25, dan nilai *f* dari 120 menjadi 110). Langkah kedua ini menghasilkan *OPEN* = [A, B, C, D, J] dan *CLOSED* = [S, E]. Penjelasan langkah kedua dari pencarian rute terpendek ditampilkan pada Gambar 2.3.



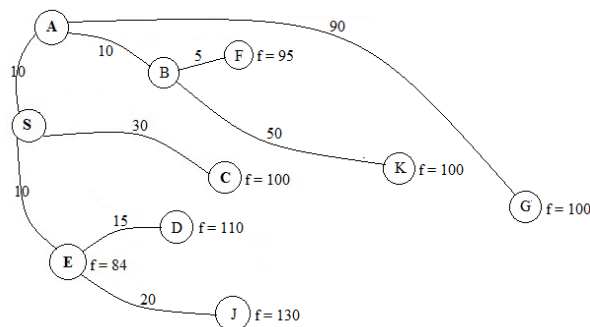
**Gambar 2.3 Langkah kedua contoh kasus algoritma A\***

3. Langkah ketiga, B dengan jarak terkecil (yaitu 85) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor B dibangkitkan, yaitu: A, F dan K. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka F dan K dimasukkan ke *OPEN*. Sedangkan simpul A sudah ada di *OPEN*, maka harus dicek apakah *parent* dari A perlu diganti atau tidak. Ternyata, jarak dari S ke A melalui B (yaitu  $25 + 10 = 35$ ) lebih besar daripada jarak dari S ke A (yaitu 10). Oleh karena itu, *parent* dari A tidak perlu diubah (tetap S). Akhir dari langkah ketiga ini menghasilkan *OPEN* = [A, C, D, F, J, K] dan *CLOSED* = [S, E, B]. Gambar 2.4 menjelaskan langkah ketiga dari pencarian rute terpendek.



**Gambar 2.4 Langkah ketiga algoritma A\***

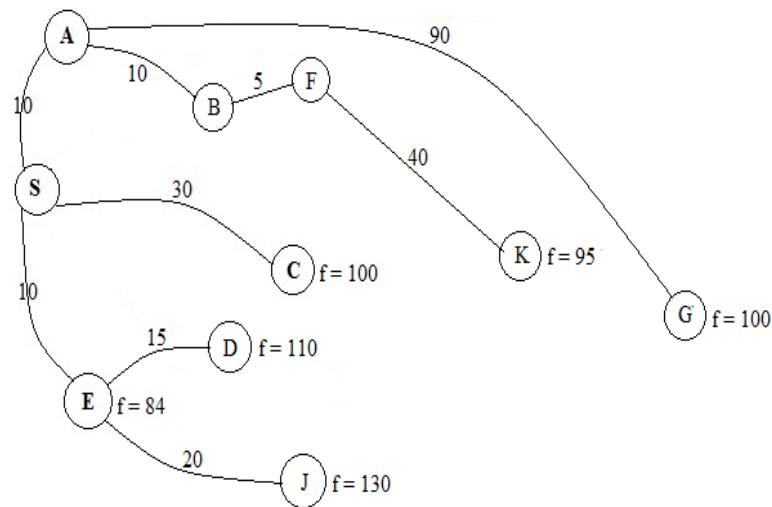
4. Langkah keempat, A dengan jarak terkecil (yaitu 90) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor A dibangkitkan, yaitu: B dan G. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka G dimasukkan ke *OPEN*. Sedangkan simpul B sudah ada di *CLOSED*, maka harus dicek apakah *parent* dari B perlu diganti atau tidak. Ternyata, jarak dari S ke B melalui A (yaitu  $10 + 10 = 20$ ) lebih kecil daripada jarak dari S ke B (yaitu 25). Oleh karena itu, *parent* dari B harus diubah, yang semula S menjadi A, nilai  $g$  dan  $f$  pada B juga harus diperbarui (nilai  $g$  yang semula 25 menjadi 20, dan nilai  $f$  dari 85 menjadi 80). Dalam kasus ini, B hanya mempunyai dua jalur, yaitu F dan K. Nilai  $g(F)$  yang semula 30 diubah menjadi 25, dan nilai  $f(F)$  dari 100 menjadi 95. Nilai  $g(K)$  yang semula 75 diubah menjadi 70, dan nilai  $f(K)$  dari 105 menjadi 100. Akhirnya,  $OPEN = [C, D, F, G, J, K]$  dan  $CLOSED = [S, E, B, A]$ . Gambar 2.5 menjelaskan langkah keempat dari pencarian rute terpendek.



**Gambar 2.5 Langkah keempat algoritma A\***

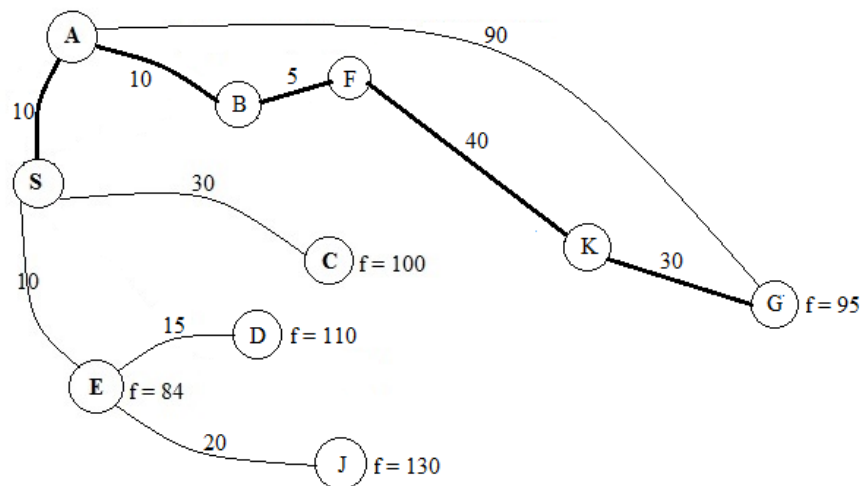
5. Langkah kelima, F dengan jarak terkecil (yaitu 95) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor F dibangkitkan, yaitu: K. Karena K sudah ada di *OPEN*, maka harus dicek apakah *parent* dari K perlu diganti atau tidak. Jarak dari S ke K melalui F ternyata lebih kecil daripada jarak dari S ke K melalui *parent* lama (B). Oleh karena itu, *parent* dari K harus diubah, yang semula B menjadi F. Selanjutnya, nilai  $g(K)$  yang semula 70 diubah menjadi 65, dan nilai  $f(K)$  dari 100 menjadi 95. Akhirnya,  $OPEN = [C, D, F, G, J, K]$  dan  $CLOSED =$

[S, E, B, A, F]. Penjelasan langkah kelima dari pencarian rute terpendek ditunjukkan pada Gambar 2.6.



**Gambar 2.6 Langkah kelima algoritma A\***

6. Langkah keenam, K dengan jarak terkecil (yaitu 95) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor K dibangkitkan, yaitu: G. Karena G sudah ada di *OPEN*, maka harus dicek apakah *parent* dari G perlu diganti atau tidak. Jarak dari S ke G melalui K ternyata lebih kecil daripada jarak S ke G melalui *parent* lama (A). Oleh karena itu, *parent* dari G harus diubah, yang semula A menjadi K. Selanjutnya, nilai  $g(G)$  yang semula 100 diubah menjadi 95, dan nilai  $f(G)$  dari 100 menjadi 95. Pada akhir langkah keenam ini,  $OPEN = [C, D, G, J]$  dan  $CLOSED = [S, E, B, A, F, K]$ . Gambar 2.7 menjelaskan langkah keenam dari pencarian rute terpendek.



**Gambar 2.7** langkah keenam contoh kasus algoritma A\*

Selanjutnya, G dengan jarak terkecil (yaitu 95) terpilih sebagai *BestNode*. Karena *BestNode* sama dengan *goal*, berarti solusi sudah ditemukan. Rute dan total jarak bisa ditelusuri balik dari G menuju S karena setiap simpul hanya memiliki satu *parent* dan setiap simpul memiliki informasi jarak sebenarnya (*g*). Penelusuran menghasilkan rute S-A-B-F-K-G dengan total jarak sama dengan 95. [4]

## 2.2 Android

Android sistem operasi untuk telepon selular yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan untuk digunakan oleh berbagai macam piranti bergerak (*mobile device*). Hal ini memungkinkan para pengembang menulis kode terkelola (*managed code*) dalam bahasa pemrograman java, mengontrol *device* via *Library Java* yang dikembangkan oleh Google.[5] Untuk aplikasi berbasis android yang di buat, akan menggunakan *tools Android Studio*.

Kelebihan android :

- *User Friendly*
- *Notifications*
- Tampilan
- *Open Source*
- Aplikasi

Kekurangan android :

- *Update System*
- Baterai cepat habis [6]

### 2.2.1 Java

Dalam mendiskusikan Java, kiranya penting sekali untuk membedakan antara bahasa pemrograman Java, *Java Virtual Machine*, dan *Platform Java*. Bahasa pemrograman Java adalah bahasa yang digunakan untuk menghasilkan aplikasi - aplikasi Java. Pada umumnya, bahasa pemrograman hanya mendefinisikan sintaks dan perilaku bahasa.

Pada saat program Java dikompilasi, ia akan dikonversi ke bentuk *bytecode*, yang merupakan bahasa mesin yang *portable*. Selanjutnya, *bytecode* tersebut dijalankan di *Java Virtual Machine* (atau disebut Java VM atau JVM). Meskipun JVM dapat diimplementasikan langsung di perangkat keras, namun biasanya diimplementasikan dalam bentuk program perangkat lunak yang mengemulasi mesin(komputer) dan digunakan untuk menginterpretasi *bytecode*.

*Platform* dapat didefinisikan sebagai perangkat lunak pendukung untuk aktivitas - aktivitas tertentu. *Platform Java* sendiri pada prinsipnya berbeda dengan bahasa Java atau JVM. *Platform Java* adalah himpunan kelas - kelas Java yang sudah didefinisikan sebelumnya dan eksis sejak instalasi Java. *Platform Java* juga mengacu pada lingkungan *runtime* atau API (*Application Programming Interface*) Java.[4]

### 2.2.2 MySQL

MySQL tergolong sebagai DBMS (*Database Mangement System*) dan banyak dipakai untuk kepentingan penanganan database karena selain handal juga bersifat *open source*. Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat. Misalnya: menyimpan, menghapus, mengubah, mengambil dan menyortir data dalam tabel.[4]

Pada aplikasi ini MySQL digunakan sebagai DBMS untuk menyimpan data informasi Fasilitas Kesehatan di Kota Malang.

Kelebihan MySQL :

- *Source* MySQL dapat diperoleh dengan mudah dan gratis
- Sintaksnya lebih mudah dipahami dan tidak rumit



- Pengaksesan database dapat dilakukan dengan mudah
- MySQL merupakan program yang *multithreaded*, sehingga dapat dipasang pada server yang memiliki *multiCPU*
- Didukung program – program umum seperti C, C++, Java, Perl, PHP, Python, dsb
- Bekerja pada berbagai *platform*. (tersedia berbagai versi untuk berbagai sistem operasi)
- Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database
- Memiliki sistem sekuriti yang cukup baik dengan verifikasi host
- Mendukung ODBC untuk sistem operasi windows
- Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi
- Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi[7]

### 2.3 Pencarian Jalur Terpendek

Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat tertentu. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan *Graf*. *Graf* adalah sekumpulan titik di dalam bidang dua dimensi yang dihubungkan dengan sekumpulan garis (*edge*). Sebuah *graf* dibentuk dari kumpulan titik yang dihubungkan dengan garis – garis.[8]

Pada Aplikasi ini pencarian jalur terpendek digunakan untuk mencari jalur terpendek menuju lokasi fasilitas kesehatan yang ada di Kota Malang yang terdapat di android. Dimana *Client* memilih kriteria terlebih dahulu sebelum mencari fasilitas kesehatan, kriteria tersebut seperti berdasarkan kecamatan dan berdasarkan fasilitas kesehatan yang bekerja sama dengan BPJS Kota Malang.

## 2.4 Web Service

*Web service* adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi yang bisa beroperasi *machine-to-machine* di atas jaringan. *Web service* mempunyai alat penghubung yang diuraikan di dalam format *machine - processable* (secara spesifik WSDL). Sistem lain saling berhubungan dengan *Web service* di dalam cara yang ditentukan oleh deskripsinya yang menggunakan pesan SOAP.[9]

Keuntungan *web service* :

- Web Service mempunyai sifat interoperability sehingga bisa diakses oleh aplikasi yang berjalan pada platform yang berbeda-beda.
- Web Service menggunakan standar dan protocol terbuka pada Internet.
- Dengan menggunakan HTTP atau SMTP, Web Service bisa menembus pengamanan firewall suatu organisasi tanpa mengubah konfigurasi firewall.
- Web Service memungkinkan fungsi-fungsi pada banyak perangkat lunak di Internet untuk dipadukan menjadi satu Web Service baru.
- Web Service memungkinkan penggunaan ulang layanan dan komponen.
- Dukungan interface yang stabil.
- Secure [10]

## 2.5 Fasilitas kesehatan

Fasilitas Kesehatan adalah segala sarana dan prasarana alat atau tempat yang dapat menunjang kesehatan atau yang dapat digunakan untuk menyelenggarakan pelayanan kesehatan yang dilakukan oleh pemerintah, pemerintah daerah dan atau masyarakat.

Di sistem BPJS fasilitas kesehatan ini dikategorikan menjadi beberapa kategori (faskes tingkat 1, 2 dan 3), pengkategorian ini dikarenakan sistem BPJS menggunakan sistem pelayanan berjenjang, artinya ketika peserta BPJS ingin berobat guna mendapatkan layanan kesehatan yang ditanggung oleh BPJS maka fasilitas kesehatan yang harus pertama kali dikunjungi adalah fasilitas kesehatan tingkat 1, jika di faskes tingkat 1, tidak memungkinkan untuk dilayani maka

dokter faskes tingkat 1 akan merujuk ke faskes tingkat 2, dan jika di faskes tingkat 2 masih tidak memungkinkan untuk dilayani karena sarana dan prasarana kurang memadai maka dokter fasilitas kesehatan tingkat 2 akan merujuk ke fasilitas kesehatan tingkat 3.

- Fasilitas Kesehatan tingkat pertama (FKTP 1)

Fasilitas kesehatan tingkat 1 terdiri dari puskesmas, optik, Apotek, klinik, klinik Tni / Polri praktek dokter, praktek dokter gigi dan rumah sakit tipe D.

- Fasilitas Kesehatan tingkat dua

Tingkat 2 menurut Sistem Rujukan Berjenjang diisi oleh dua tipe 2 rumah sakit yaitu C, B. Di lapangan, BPJSK akan mengarahkan bahwa jika dari PPK 1 pasien tidak bisa ditangani maka akan dirujuk secara berjenjang ke tipe D atau C lebih dulu, baru ke tipe B. Bila diperlukan baru ke tipe A.

- Fasilitas Kesehatan tingkat 3

Fasilitas kesehatan tingkat 3 diisi oleh rumah sakit tipe A, rumah sakit ini adalah rumah sakit yang paling lengkap dengan sarana dan prasarana ini adalah rujukan terakhir pasien BPJS jika pasien tidak bisa ditangani di PPK1 dan juga PPK2.[4]