

# Aplikasi Graf pada Sistem Parkir Otomatis

Chrestella Stephanie - 13512005  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13512005@std.stei.itb.ac.id

**Abstrak**—Makalah ini membahas penggunaan teori graf dalam sistem parkir otomatis atau bisa juga disebut dengan *robotic parking system*. Teori graf yang digunakan pada teknologi ini adalah teori mengenai mencari lintasan terpendek (*shortest path*) pada graf. Teori ini diimplementasi untuk menentukan lintasan mana yang harus ditempuh oleh mesin penggerak mobil sehingga jarak yang dilalui mesin tersebut adalah jarak yang terpendek. Dengan digunakannya teori tersebut, diharapkan waktu yang dibutuhkan untuk melakukan 1 (satu) kali peletakan mobil dapat diminimalisasi.

**Kata Kunci**—*shortest path*, graf, parkir, *robotic parking system*.

## I. PENDAHULUAN

Saat ini, jumlah kendaraan yang ada di dunia bertambah dengan sangat pesat. Hal ini tentu saja membawa berbagai dampak bagi penduduk dunia, baik itu dampak buruk maupun dampak yang baik. Salah satu dampak buruk yang ditimbulkan adalah semakin banyaknya lahan parkir yang dibutuhkan untuk menampung mobil-mobil, sedangkan lahan parkir terbatas dan hampir tidak mungkin lahan untuk parkir bertambah jumlahnya. Namun, dengan kemajuan teknologi yang pesat, saat ini sudah ada teknologi untuk mengatasi masalah ini, yaitu sistem parkir otomatis atau disebut juga *robotic parking system*.

*Robotic parking system* ini dapat memaksimalkan lahan parkir untuk kapasitas yang banyak di ruang minimum sehingga teknologi ini sangat cocok untuk digunakan di pusat-pusat perbelanjaan, perkantoran, hotel, apartemen, dan semacamnya. *Robotic parking system* ini sudah digunakan di beberapa negara, diantaranya Jerman, Dubai, dan Mumbai.

Sistem *robotic parking system* adalah sebagai berikut :

1. Pengendara memarkirkan mobilnya di tempat yang sudah tersedia. Di sekitar tempat mobil tersebut diparkirkan diberi beberapa sensor untuk memastikan mobilnya sudah berada pada posisi yang tepat.
2. Pengendara meninggalkan mobilnya dengan keadaan terkunci.

3. Pengendara mengambil kartu parkir dari sebuah mesin otomatis.
4. Mobil diletakan oleh mesin ke tempat yang masih kosong.
5. Untuk mengambil kembali mobilnya, pengemudi hanya perlu menge-*scan* kartu parkir dan menunggu hingga mobil siap di ambil di tempat yang sudah ditentukan.



Gambar 1. Sistem parkir otomatis

Sumber : [www.travelsservicesindia.blogspot.com/2011/11/hi-tech-parking-multi-level-automated.html](http://www.travelsservicesindia.blogspot.com/2011/11/hi-tech-parking-multi-level-automated.html)

Berbagai keuntungan dapat dirasakan dengan menggunakan teknologi ini. Diantaranya adalah lahan parkir yang terbatas dapat menampung kendaraan sebanyak puluhan kali lipat lebih banyak dibanding menggunakan lahan parkir konvensional. Tingkat keamanan kendaraannya pun lebih terjamin karena untuk mengakses kendaraan diperlukan kartu parkir yang memiliki kode unik untuk setiap blok parkir kendaraan tersebut, sehingga tidak sembarangan orang dapat mengakses kendaraan yang bukan miliknya. Dan hal yang terpenting adalah efisiensi waktu. Waktu yang dibutuhkan untuk memarkirkan mobil menjadi lebih sedikit. Pengemudi hanya perlu memarkirkan

kendaraannya di tempat yang telah disediakan dan dapat langsung meninggalkan kendaraannya dan melanjutkan aktivitasnya. Dengan demikian, waktu yang biasanya digunakan untuk mencari tempat parkir yang masih kosong dapat digunakannya untuk keperluan lain.

Karena waktu merupakan hal penting untuk diperhatikan, maka proses yang dilakukan mesin ini untuk 1 kali proses meletakkan mobil harus diusahakan seminimal mungkin. Hal ini dapat dicapai dengan menggunakan teori graf untuk menentukan lintasan terpendek. Dengan demikian, waktu yang dibutuhkan untuk melakukan 1 kali proses adalah waktu minimal yang dibutuhkan.

## II. DASAR TEORI

### A. Definisi dan Jenis Graf

Menurut buku diktat Matematika Diskrit[1], secara matematis, graf didefinisikan sebagai pasangan himpunan  $(V,E)$ , yang dalam hal ini :

$V$  = himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) =  $\{v_1, v_2, v_3, \dots, v_n\}$

$E$  = himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul =  $\{e_1, e_2, \dots, e_n\}$

atau dapat ditulis singkat notasi  $G=(V,E)$ .

Graf ada berbagai macam jenisnya berdasarkan sifat-sifatnya. Berdasarkan ada tidaknya sisi ganda atau gelang pada suatu graf, graf dibedakan menjadi

1. Graf sederhana  
Graf yang tidak mengandung sisi ganda maupun sisi gelang
2. Graf tidak sederhana  
Graf yang mengandung sisi ganda dan/atau sisi gelang.

Berdasarkan jumlah simpul pada graf :

1. Graf berhingga  
Graf yang jumlah simpulnya berhingga
2. Graf tidak berhingga  
Graf yang simpulnya tak terhingga banyaknya

Berdasarkan orientasi arah pada sisi :

1. Graf berarah
2. Graf tidak berarah

### B. Beberapa Terminologi Dasar yang Digunakan dalam Makalah Ini

1. Bertetangga  
Definisi : dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain,  $v_i$  bertetangga dengan  $v_k$  jika  $(v_i, v_k)$  adalah sebuah sisi pada graf  $G$ .
2. Lintasan  
Definisi : Lintasan  $v_0-v_1$  pada graf  $G$  adalah sebuah barisan berhingga  $v_0, e_1, v_1, e_2, \dots, e_n, v_n$  bergantian titik dan sisi pada  $G$  sedemikian sehingga  $e_i = v_{i-1} v_i$  untuk setiap  $i, 1 \leq i \leq n$

### 3. Graf kosong

Definisi : graf yang himpunan sisinya merupakan himpunan kosong disebut sebagai graf kosong dan ditulis sebagai  $N_n$ , yang dalam hal ini  $n$  adalah jumlah simpul

Graf kosong pada kasus *robotic parking system* berarti sudah tidak ada lagi lahan parkir yang tersedia dan hanya ada sebuah simpul saja, yaitu simpul  $A1$  sebagai titik awal atau *start point*.

### C. Representasi Graf

Untuk pemrosesan dengan komputer atau otomatisasi, graf direpresentasikan dalam bentuk matriks. Ada beberapa representasi yang sering digunakan untuk merepresentasikan graf dalam bentuk matriks, diantaranya matriks ketetangaan (*adjacency matrix*), matriks bersisian (*incidency matrix*), dan senarai ketetangaan (*adjacency list*) (diambil dari diktat Matematika Diskrit[1].

Matriks Bersisian atau *incidency matrix* menyatakan ketetangaan simpul-simpul di dalam graf, maka matriks bersisian menyatakan kebersisian simpul dengan sisi. Matriks bersisian ini dapat digunakan untuk merepresentasikan graf yang mengandung sisi ganda atau sisi gelang. Derajat setiap simpul  $i$  dapat dihitung dengan menghitung jumlah seluruh elemen pada baris  $i$  kecuali pada graf yang mengandung gelang.

Senarai ketetangaan atau *adjacency list* digunakan bila graf memiliki jumlah sisi yang relatif sedikit, karena matriks nya bersifat jarang, yaitu mengandung banyak elemen 0. Sedangkan elemen yang bukan 0 hanya sedikit. Jika ditinjau dari segi memori, representasi jenis ini dikatakan boros memori karena menyimpan banyak elemen 0 yang tidak perlu.

Matriks ketetangaan atau dikenal juga dengan sebutan *adjacency matriks* adalah salah satu representasi graf yang sering digunakan. Matriks ketetangaan ini menyatakan status ketetangaan sebuah simpul apakah ia bertetangga dengan simpul yang lainnya atau tidak.

Definisi suatu simpul bertetangga dengan simpul lainnya menurut buku diktat Matematika Diskrit[1] ialah "Dua buah simpul pada graf tak-berarah  $G$  dikatakan **bertetangga** bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain,  $v_i$  bertetangga dengan  $v_k$  jika  $(v_i, v_k)$  adalah sebuah sisi pada graf  $G$ ". Untuk graf tidak berbobot, matriks hanya berisikan angka 0 atau 1, yang berarti 0 jika antar simpulnya tidak bertetangga, dan angka 1 jika antar simpulnya bertetangga. Untuk graf berbobot, matriks tidak diisi dengan angka 0 atau 1 saja, tetapi menggunakan bobot masing-masing sisi graf tersebut. Untuk simpul yang tidak memiliki sisi yang menghubungkan dengan simpul lain, maka matriks bagian kolom dan baris tersebut diisi dengan  $\infty$ .

Matriks ketetangaan untuk graf sederhana dan

tidak berarah selalu simetri, sedangkan untuk graf berarah matriks nya belum tentu simetri.

#### D. Algoritma Dijkstra

Algoritma Dijkstra ini ditemukan oleh Edsger Wybe Dijkstra. Algoritma ini dapat digunakan untuk mencari lintasan terpendek antar simpul dengan menggunakan prinsip *greedy*. Prinsip *greedy* yang digunakan pada algoritma ini adalah memilih lintasan mana yang terpendek dan memasukkan lintasan-lintasan tersebut ke himpunan solusi. Berikut ini adalah *pseudocode* algoritma Dijkstra.

Misalkan sebuah graf berbobot dengan  $n$  buah simpul dinyatakan dengan matriks ketetanggaan  $M=[m_{ij}]$ , yang dalam hal ini

$$m_{ij} = \text{bobot sisi } (i,j) \text{ (pada graf tidak berarah)}$$

$$(m_{ij} = m_{ji})$$

$$m_{ii} = 0$$

$$m_{ij} = \infty$$

Selain matriks  $M$ , kita juga menggunakan larik  $S=[s_i]$  yang dalam hal ini,

$$s_i = 1, \text{ jika simpul } i \text{ termasuk ke dalam lintasan terpendek.}$$

$$s_i = 0, \text{ jika simpul } i \text{ tidak termasuk ke dalam simpul terpendek}$$

dan larik/tabel  $D = [d_i]$  yang dalam hal ini,

$$d_i = \text{panjang lintasan dari simpul awal } a \text{ ke simpul } i$$

Algoritma Dijkstra dinyatakan dalam notasi *pseudocode* sebagai berikut :

#### Algoritma

```
{langkah 0 : inisialisasi}
for i ← 1 to n do
    si ← 0
    di ← mai
end for

{langkah 1}
sa ← 1 {karena simpul a adalah simpul
        asal lintasan terpendek, jadi
        simpul a sudah pasti terpilih
        dalam lintasan terpendek}
da ← ∞ {tidak ada lintasan terpendek dari
        a ke a}

{langkah 2,3,...,n-1}
for i ← 2 to n-1 do
    cari j sedemikian sehingga sj=0 dan
        dj=min{d1,d2,...,dn}
    sj ← 1 {simpul j sudah terpilih ke
        dalam lintasan terpendek}
    perbarui di, untuk i=1,2,3,...,n
        dengan di(baru)=min{di(lama),dj+mji}
end for
```

### III. PENJELASAN

Tipe graf yang digunakan dalam *robotic parking system* adalah tipe graf berbobot dan tidak berarah, dengan bobotnya adalah jarak antar simpul. Bentuk tempat parkir otomatis tidak selalu sama. Ada yang dibuat melingkar dan ada yang melebar ke samping atau ada pula yang ke atas membentuk sebuah persegi panjang. Namun pada umumnya, teori graf untuk mencari jarak terpendek dapat diterapkan pada kedua bentuk tempat parkir tersebut. Untuk bentuk yang melingkar, misalkan banyaknya lahan parkir yang tersedia adalah sebanyak 8x6, bentuk tersebut dapat digambarkan sama dengan lahan yang berbentuk melebar ke samping, yaitu sebagai berikut :

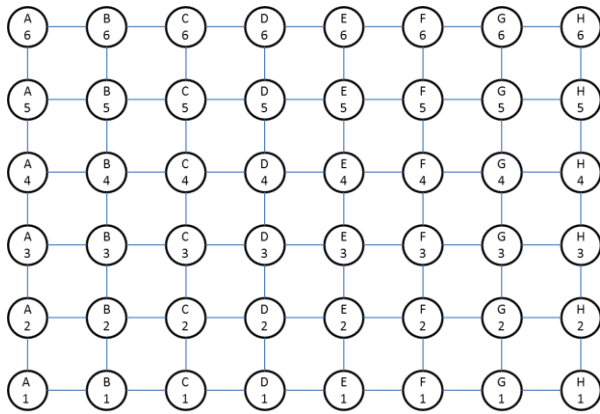
	A	B	C	D	E	F	G	H
6								
5								
4								
3								
2								
1								

Tabel 1. Representasi tempat parkir bentuk melingkar dalam bentuk melebar ke samping

Lahan yang kosong diibaratkan sebagai simpul sebuah graf, dan lintasan antar lahan yang kosong diibaratkan sebagai busur.

```
procedure Dijkstra (input m:matriks,
                    a:simpul awal)
{mencari lintasan terpendek dari simpul
awal a ke semua simpul lainnya.
input:matriks ketetanggaan(M) dari graf
berbobot G dan simpul awal a
output:lintasan terpendek dari a ke
semua simpul lainnya
}

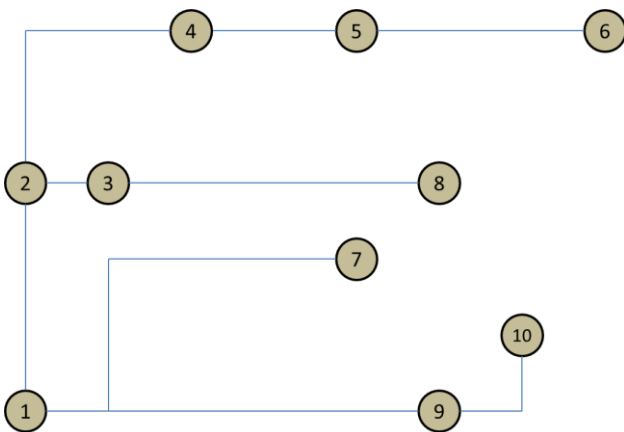
Deklarasi
s1,s2,...,sn : integer {larik integer}
d1,d2,...,dn : integer {larik integer}
i : integer
```



Gambar 2. Graf yang merepresentasikan lahan parkir kosong

j =	1	2	3	4	5	6	7	8	9	10
i = 1	0	3x	4x	8x	9x	12x	6x	8x	5x	7x
2	3x	0	x	4x	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
3	4x	x	0	$\infty$	$\infty$	$\infty$	$\infty$	4x	$\infty$	$\infty$
4	8x	4x	$\infty$	0	2x	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
5	9x	$\infty$	$\infty$	2x	0	3x	$\infty$	$\infty$	$\infty$	$\infty$
6	12x	$\infty$	$\infty$	$\infty$	3x	0	$\infty$	$\infty$	$\infty$	$\infty$
7	6x	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$
8	8x	$\infty$	4x	$\infty$	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$
9	5x	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	2x
10	7x	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	2x	0

Tabel 2. Matriks yang merepresentasikan sebuah graf berbobot



Gambar 3. Graf yang merepresentasikan lahan parkir kosong sebagian

Kita asumsikan *start point* atau titik awal adalah simpul A1, yaitu tempat awal mobil dimasukkan. Misalkan jarak antar lahan parkir baik horizontal maupun vertikal adalah X. Untuk mencari jarak terpendek dari simpul A1 ke simpul terdekat dapat menggunakan algoritma Dijkstra.

Sebelum mengaplikasikan algoritma Dijkstra, ubah dahulu graf tersebut dengan representasi matriks ketetanggaan (*adjacency matriks*). Kita tinjau graf tersebut ke dalam bentuk matriks sebagai berikut

Setelah diperoleh matriks tersebut, kita terapkan algoritma Dijkstra untuk mencari lintasan mana yang terpendek. Perhitungan lintasan terpendek dari simpul awal (1) ke semua simpul lainnya dapat ditentukan sebagai berikut.

1. Hitung semua jarak lintasan dari simpul 1 ke semua simpul lain. Dari tabel matriks di atas jelas terlihat bahwa simpul 2 memiliki bobot yang paling kecil. Hal ini berarti jarak dari simpul 1 ke simpul 2 adalah jarak yang terdekat.
2. Setelah menemukan jarak terpendek, cari kembali jarak mana yang paling pendek. Dari matriks di atas, simpul 3 adalah bobot terkecilnya.
3. Lakukan hal tersebut untuk semua simpulnya
4. Setelah jarak dari simpul 1 ke simpul lainnya sudah diketahui semua, maka tinggal dipilih satu yang paling pendek jaraknya. Dari perhitungan didapat bahwa :  
Lintasan terpendek dari 1 ke simpul lainnya adalah :  
1 ke 2, dengan jarak 3x  
1 ke 3 dengan jarak 4x  
1 ke 9 dengan jarak 5x  
1 ke 7 dengan jarak 6x  
1 ke 10 dengan jarak 7x  
1 ke 4 dengan jarak 8x  
1 ke 8 dengan jarak 8x  
1 ke 5 dengan jarak 9x  
1 ke 6 dengan jarak 12x

Dari hasil tersebut, maka diperoleh simpul yang memiliki jarak terpendek dengan simpul 1 adalah simpul ke 2, yaitu sebesar 3x. Jadi, lahan parkir terdekat adalah simpul dengan no 2. Hal ini menandakan bahwa lahan parkir kosong yang terdekat dengan simpul awal atau *start point* adalah lahan yang direpresentasikan oleh simpul 2, yaitu lahan no A4. Mesinipun akan meletakkan

mobil di lahan A4. Dengan demikian, waktu yang dibutuhkan untuk melakukan 1(satu) kali proses peletakkan mobil akan menjadi efisien, karena tidak perlu menelusuri satu persatu lahan parkir dan juga mesin tidak perlu berputar-putar mencari lahan mana yang kosong atau menuju suatu lahan kosong dengan jalan berputar-putar.

#### IV. KESIMPULAN

Akhir-akhir ini, salah satu masalah yang ada di dunia terutama di Indonesia adalah masalah kurangnya lahan parkir yang tersedia, namun jumlah permintaan semakin bertambah banyak karena semakin bertambahnya jumlah kendaraan yang digunakan, terutama kendaraan roda 4. Dengan adanya perkembangan teknologi yang pesat, maka diciptakan *robotic parking system* yang dapat membantu menanggulangi masalah tersebut.

Salah satu ilmu yang dipakai dalam teknologi ini adalah teori mengenai graf. Teori graf yang digunakan adalah untuk mencari lintasan terpendek atau *shortest path*. Teori ini menggunakan algoritma Dijkstra, yaitu algoritma untuk mencari lintasan terpendek. Dengan ditentukannya lintasan yang terpendek, robot akan memilih lahan tersebut untuk sebagai tempat parkir untuk mobil yang akan diparkirkan, sehingga diharapkan waktu yang dibutuhkan untuk meletakkan sebuah mobil dapat diminimalisasi dan menjadi lebih efisien.

#### DAFTAR PUSTAKA

- [1] Munir, Rinaldi. Matematika Diskrit. Bandung, Informatika Bandung, 2010.
- [2] <http://weburbanist.com/2009/01/22/creative-innovative-and-hilarious-parking-solutions/> 9/12/2013
- [3] [www.travelservicesindia.blogspot.com/2011/11/hi-tech-parking-multi-level-automated.html](http://www.travelservicesindia.blogspot.com/2011/11/hi-tech-parking-multi-level-automated.html) 9/12/2013
- [4] [http://www.roboticparking.com/robotic\\_parking\\_how\\_it\\_works.htm](http://www.roboticparking.com/robotic_parking_how_it_works.htm) 9/12/2013
- [5] <http://learn.unej.ac.id/courses/KPM1404/document/> 16/12/2013

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 17 Desember 2013



Chrestella Stephanie  
13512005