

# Penentuan Jalur Pariwisata Kota Batu dengan Menggunakan Algoritma Dijkstra

Tony Eko Yuwono - 13518030  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13518030@std.stei.itb.ac.id

**Abstract**—Kota Batu adalah kota pariwisata. Terdapat banyak sekali destinasi wisata yang dapat dikunjungi di Kota Batu. Namun, sering kali para wisatawan belum mengetahui jalur-jalur yang efektif dan efisien untuk sampai ke destinasi wisata sehingga para wisatawan lebih banyak menghabiskan waktu di jalan. Oleh karena itu penulis membuat makalah mengenai pencarian jalur pariwisata yang diharapkan dapat membantu pembaca menentukan jalur objek wisata dengan efektif dan efisien.

Untuk menyelesaikan masalah ini penulis menggunakan teori graf. Teori graf banyak digunakan untuk membantu manusia menyelesaikan berbagai hal. Salah satu permasalahan dalam kehidupan sehari-hari yang dapat diselesaikan menggunakan teori graf adalah pencarian rute terdekat dengan menggunakan Algoritma Dijkstra. Dengan mengetahui jarak terdekat, setiap orang yang melakukan perjalanan wisata dapat menghemat waktu, tenaga, dan biaya untuk sampai ke tujuan wisata.

**Kata Kunci**— Algoritma Dijkstra, jarak terdekat, Kota Batu, pariwisata, teori graf.

## I. PENDAHULUAN

Wisata merupakan hal menyenangkan yang ditunggu-tunggu oleh banyak orang. Wisata dapat dilakukan oleh pribadi (perseorangan) maupun dalam kelompok. Berbiaca mengenai pariwisata, Kota Batu adalah salah satu kota wisata terbesar di Indonesia. Di kota ini terdapat banyak sekali destinasi wisata baik destinasi wisata alam maupun wisata buatan. Oleh karena itu diperlukan perencanaan agar wisatawan dapat menikmati destinasi-destinasi wisata di Kota Batu tanpa banyak membuang waktu untuk transportasi.

Untuk mengatasi masalah tersebut, penulis memberikan solusi dengan menggunakan algoritma dijsktra pada teori graf untuk diaplikasikan pada peta perjalanan wisata di Kota Batu, sehingga diharapkan pembaca dapat mengetahui jalur yang efektif dan efisien, dapat mengatur jadwal untuk menentukan alternatif objek wisata dari satu destinasi menuju destinasi lain, dan pembaca dapat menikmati destinasi-destinasi wisata di Kota Batu tanpa banyak menghabiskan waktu untuk transportasi di jalan.

## II. TEORI GRAF

Graf adalah suatu representasi visual yang terdiri dari simpul (*vertex*) dan sisi (*edge*). Secara umum, graf didefinisikan:

$$G = (V, E)$$

dengan  $V$  adalah himpunan tidak kosong dari simpul-simpul ( $V \neq \emptyset$ ), dan  $E$  adalah himpunan sisi yang menghubungkan sepasang simpul (boleh himpunan kosong) [3].

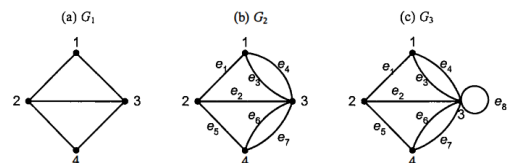
Graf dibagi menjadi dua jenis berdasarkan adanya sisi ganda atau gelang (sisi yang menghubungkan sebuah simpul yang sama):

### 1. Graf sederhana (*simple graph*)

Graf sederhana adalah graf yang tidak mengandung sisi ganda maupun gelang. Gambar 1(a) adalah contoh graf sederhana. Pada graf sederhana sisi  $(a, b) = (b, a)$ , seperti contoh pada Gambar 1(a) sisi  $(1,2)$  sama dengan  $(2,1)$ .

### 2. Graf tak-sederhana (*unsimple graph*)

Graf tak-sederhana adalah graf yang mengandung sisi ganda atau gelang (*loop*). Graf yang mengandung sisi ganda dinamakan graf ganda, sedangkan graf yang memiliki sisi gelang (*loop*) dinamakan graf semu. Gambar 1 (b) dan (c) adalah contoh graf ganda dan graf semu.



Gambar 1. (a) Graf sederhana, (b) graf ganda, (c) graf semu

Sumber: Matematika Diskrit Edisi 3, Rinaldi Munir, Halaman 356

Selain itu, berdasarkan orientasi arah, graf dibedakan menjadi dua jenis, yaitu:

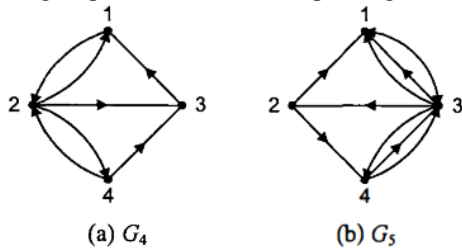
### 1. Graf tak-berarah (*undirected graph*)

Graf tak berarah adalah graf yang tidak mempunyai orientasi arah sehingga urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Misalkan terdapat sisi  $(u, v)$  pada graf  $G$ , maka sisi  $(v, u)$  adalah sisi yang sama. Semua graf pada Gambar 1 merupakan graf tak-berarah.

### 2. Graf berarah (*directed graph/digraph*)

Graf berarah adalah graf yang setiap sisinya diberikan

orientasi arah. Misalkan terdapat sisi  $(u, v)$  pada graf  $G$ , maka sisi  $(v, u)$  adalah sisi yang berbeda, dengan kata lain  $(u, v) \neq (v, u)$ . Untuk sisi  $(u, v)$ , simpul  $u$  dinamakan simpul asal dan simpul  $v$  dinamakan simpul terminal. Semua graf pada Gambar 2 merupakan graf berarah.



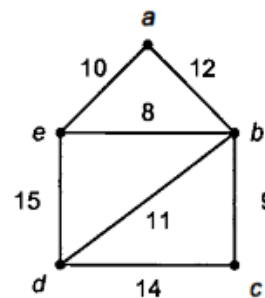
Gambar 2. (a) Graf berarah, (b) graf-ganda berarah  
 Sumber: Matematika Diskrit Edisi 3, Rinaldi Munir, Halaman 359

Graf memiliki beberapa terminologi (istilah) dasar, antara lain:

- Bertetangga (Adjacent)**  
 Simpul  $u$  dan  $v$  dikatakan bertetangga jika terdapat sisi  $(u, v)$  pada graf. Pada Gambar 1(a), simpul 1 bertetangga dengan simpul 2, begitu pula sebaliknya.
- Bersisian (Incident)**  
 Untuk sembarang  $e = (u, v)$ , sisi  $e$  bersisian dengan simpul  $u$  dan  $v$ . Pada Gambar 1(a), sisi  $(1,2)$  bersisian dengan simpul 1 dan simpul 2.
- Simpul Terpencil (Isolated Vertex)**  
 Simpul terpencil adalah simpul yang tidak memiliki sisi yang bersisian dengannya. Dengan kata lain simpul terpencil adalah simpul yang tidak bertetangga dengan simpul manapun.
- Graf Kosong (Null/Empty Graph)**  
 Graf  $G$  merupakan graf kosong jika himpunan sisinya merupakan himpunan kosong. Graf kosong memiliki notasi  $N_n$  dengan  $n$  adalah jumlah simpul.
- Derajat (Degree)**  
 Pada sebuah graf tak-berarah  $G$  dengan simpul  $u$ , derajat  $u$  adalah jumlah semua sisi yang bersisian dengan simpul  $u$ . Notasi yang menyatakan derajat suatu simpul  $u$  adalah  $d(u)$ . Jika terdapat  $g$  buah gelang dan  $e$  buah sisi bukan gelang yang bersisian dengan simpul  $u$  maka derajat simpul  $u$  adalah
 
$$d(u) = 2g + e$$
 Sedangkan pada graf berarah, derajat suatu simpul dibagi menjadi dua macam, yaitu derajat masuk ( $d_{in}$ ) dan derajat keluar ( $d_{out}$ ). Pada simpul  $u$ , Derajat masuk adalah jumlah busur yang masuk ke simpul  $u$ , sedangkan derajat keluar adalah jumlah busur yang keluar dari simpul  $u$ .
- Lintasan (Path)**  
 Lintasan  $n$  dari simpul  $v_0$  ke simpul  $v_n$  pada graf  $G$  adalah barisan berselang-seling simpul dan sisi yang membentuk  $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$  sedemikian

sehingga  $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$  adalah sisi-sisi dari graf  $G$ .

- Sirkuit (Circuit)**  
 Sirkuit adalah sebuah lintasan yang berawal dan berakhir pada simpul yang sama. Pada Gambar 1(a) terdapat sirkuit 1, 2, 3, 1. Sebuah sirkuit dikatakan sirkuit sederhana jika setiap sisi yang dilalui berbeda.
- Terhubung (Connected)**  
 Simpul  $u$  dan simpul  $v$  dikatakan terhubung jika terdapat lintasan dari simpul  $u$  ke simpul  $v$ . Jika setiap pasang simpul di dalam graf tak-berarah terhubung satu sama lain, maka graf tak-berarah tersebut dikatakan graf terhubung. Sedangkan pada graf berarah, graf tersebut dikatakan terhubung jika graf tak-berarahnya terhubung.
- Upagraf (Subgraph) dan Komplemen Upagraf**  
 Pada sebuah graf  $G = (V, E)$ , terdapat graf  $G_1 = (V_1, E_1)$  yang merupakan upagraf dari  $G$  jika  $V_1 \subseteq V$  dan  $E_1 \subseteq E$ . Sedangkan komplemen upagraf  $G_1$  adalah  $G_2 = (V_2, E_2)$  sedemikian sehingga  $E_2 = E - E_1$  dan  $V_2$  adalah himpunan simpul dengan anggota-anggota  $E_2$  berisikan dengannya.
- Upagraf Merentang (Spanning Subgraph)**  
 Upagraf  $G_1 = (V_1, E_1)$  dikatakan graf merentang dari graf  $G = (V, E)$  jika  $V_1 = V$ , yang berarti graf  $G_1$  berisi semua simpul dari graf  $G$ .
- Graf Berbobot (Weighted Graph)**  
 Graf berbobot adalah graf yang setiap sisinya diberi suatu nilai atau bobot tertentu. Bobot pada graf dapat digunakan untuk menunjukkan jarak, waktu, ongkos produksi, dan sebagainya.



Gambar 2. Graf Berbobot  
 Sumber: Matematika Diskrit Edisi 3, Rinaldi Munir, Halaman 376

### III. ALGORITMA DIJKSTRA

Algoritma Dijkstra adalah sebuah algoritma yang berguna untuk memecahkan permasalahan jarak terpendek pada graf (*shortest path problem*) yang ditemukan oleh Edsger Dijkstra. Algoritma ini dipublikasikan tahun 1959 pada jurnal matematika yang berjudul "A Note on Two Problems in Connexion with Graphs" dan dianggap sebagai algoritma *greedy*, yaitu algoritma yang digunakan untuk memecahkan

masalah yang berhubungan dengan suatu optimasi [2]. Dalam pencarian jalur terpendeknya, algoritma dijkstra mencari bobot terkecil dari suatu graf berbobot tidak negatif. Jalur terpendek dapat diperoleh dari dua atau lebih titik dari suatu graf dengan nilai total yang didapat bernilai paling kecil.

Misalkan graf  $G$  adalah graf berarah dan berbobot tidak negatif dengan definisi  $G = \{V, E\}$ . Algoritma Dijkstra akan membentuk suatu jalur dari satu simpul ke simpul lain secara optimal (pencarian bobot terkecil) setiap langkah. Langkah-langkah yang dilakukan pada algoritma Dijkstra adalah sebagai berikut [4]:

1. Menginisiasi semua simpul dengan bobot tak terhingga (*infinite*) dan simpul awal dengan bobot nol,
2. Membandingkan bobot sisi semua simpul yang berhubungan dengan simpul awal kemudian mengambil simpul dengan bobot terkecil dan mengembalikan bobot awalnya (sebelum menjadi tak terhingga),
3. Mengulangi langkah satu dan langkah dua untuk simpul yang belum pernah dilalui dan menghitung jaraknya dari simpul awal hingga semua simpul terhubung.

Berikut *source code* dalam *syntax* bahasa Python yang merepresentasikan algoritma Dijkstra:

```
function Dijkstra(Graph, source):
    dist[source] := 0 // Distance from source to source is set to 0
    for each vertex v in Graph: // Initializations
        if v != source
            dist[v] := infinity // Unknown distance function from source to each node set to infinity
            add v to Q // All nodes initially in Q

    while Q is not empty: // The main loop
        v := vertex in Q with min dist[v] // In the first run-through, this vertex is the source node
        remove v from Q

        for each neighbor u of v: // where neighbor u has not yet been removed from Q
            alt := dist[v] + length(v, u)
            if alt < dist[u]: // A shorter path to u has been found
                dist[u] := alt // Update distance of u

    return dist[]
end function
```

Gambar 3. Source Code Algoritma Dijkstra

Sumber: <https://brilliant.org/wiki/dijkstras-short-path-finder/>

#### IV. KOTA BATU

Kota Batu adalah sebuah kota yang terletak di Jawa Timur, Indonesia. Kota ini terletak 15 km sebelah barat Kota Malang, berada di jalur Malang-Kediri dan Malang-Jombang. Kota Batu berbatasan dengan Kabupaten Mojokerto dan Pasuruan di sebelah utara dan dengan Kabupaten Malang di sebelah timur, selatan, dan barat. Kota ini berada di ketinggian 680-1200 meter di atas permukaan laut dengan suhu udara rata-rata 15-19°C [1].

Dengan luas wilayah sekitar 202,30 km<sup>2</sup>, sebagian besar keadaan topografi Kota Batu didominasi kawasan dataran tinggi dan perbukitan yang berlembah-lembah yang terletak di lereng dua pegunungan besar, yakni Arjuno-Welirang dan Butak-Kawi-Panderman. Di sebelah utara pusat Kota Batu, terdapat sebuah hutan lindung, yakni Taman Hutan Raya Raden Suerjo. Sebagai wilayah pegunungan yang subur, Kota Batu juga memiliki panorama alam yang indah dan berudara sejuk. Hal ini tentunya akan menarik minat wisatawan untuk menikmati Kota Batu sebagai kawasan pegunungan yang mempunyai daya tarik sendiri.



Gambar 4. Kota Wisata Batu

Sumber:

<https://www.malangtimes.com/baca/36843/20190308/195600/ke-kota-batu-jangan-lewatkan-taman-taman-hutan-kota-dengan-fasilitas-gratis>

Kota Batu merupakan salah satu kota wisata terbesar di Indonesia. Kota Batu memiliki banyak sekali destinasi wisata dengan jumlah kunjungan wisatawan ke kota ini merupakan salah satu yang terbesar bersama dengan Bali dan Yogyakarta. Hal ini berdasar pada kondisi alam dan letak geografis yang mendukung. Destinasi wisata di Kota Batu bervariasi satu sama lain sehingga satu dengan yang lainnya tidak terjadi persaingan yang cukup berarti. Objek wisata di Kota Batu sangat beragam, mulai dari objek sejarah, retail, pendidikan, hingga kawasan alam. Beberapa objek wisata Kota Batu antara lain:

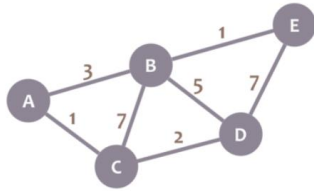
1. Taman Wisata Selecta
2. Museum Angkut
3. Museum D'Topeng
4. Omah Kayu Batu Malang
5. Wisata Paralayang Batu
6. Wisata Pujon Kidul
7. Kusuma Agrowisata
8. Labirin Coban Rondo
9. Candi Songgoriti
10. Bukit Bulu Coban Rais
11. BNS (Batu Night Spectacular)
12. Jawa Timur Park 1
13. Jawa Timur Park 2
14. Batu Secret Zoo
15. Eco Green Park
16. Museum Satwa
17. Kaliwatu Rafting

Letak destinasi wisata di Kota Batu relatif berdekatan sehingga dengan mudah ditemui oleh para wisatawan. Hal ini membuat

#### V. PEMBAHASAN

1. Penggunaan Algoritma Dijkstra untuk Menentukan Jarak Terpendek

Algoritma dijkstra adalah algoritma yang berguna untuk mencari lintasan terpendek dari sebuah simpul yang disebut sebagai simpul asal menuju ke seluruh simpul yang ada pada graf terkait. Misalkan graf  $G$  yang merupakan graf berbobot tidak negatif:



Gambar 5. Graf G

Sumber:

<https://www.codingame.com/playgrounds/1608/shortest-paths-with-dijkstras-algorithm/dijkstras-algorithm>

Pada Graf tersebut akan dicari jarak terpendek dari simpul C menuju semua simpul yang ada pada graf tersebut. Berikut langkah-langkah yang dilakukan dengan menggunakan algoritma dijkstra untuk mendapatkan jarak terpendek dari simpul C menuju simpul A, B, D, dan E:

1. Menginisiasi C sebagai simpul awal dengan bobot nol dan simpul lain dengan bobot tak terhingga (*infinite*)

Simpul	A	B	C	D	E
Bobot	$\infty$	$\infty$	0	$\infty$	$\infty$

Tabel 1. Inisiasi C sebagai simpul awal

Keterangan: **Merah**: Simpul saat ini

2. Mengecek jarak minimum simpul C menuju simpul tetangganya (simpul A, B, dan D)

Simpul	A	B	C	D	E
Bobot	<del><math>\infty</math></del> <b>1</b>	<del><math>\infty</math></del> <b>7</b>	0	<del><math>\infty</math></del> <b>2</b>	$\infty$

Tabel 2. Perubahan Bobot pada simpul tetangga dari C (simpul A, B, dan D)

Keterangan: **Merah**: Simpul saat ini  
**Kuning**: Perubahan bobot pada graf

Pertama-tama pilih simpul B. Tambahkan jarak minimum dari simpul saat ini (simpul C) dan simpul B, didapat jarak dari simpul C dan B adalah  $0 + 7 = 7$ . Selanjutnya lakukan perbandingan terhadap bobot simpul B pada tabel 1. Karena bobot simpul B pada tabel 1 bernilai tak terhingga, maka nilai tersebut diganti dengan nilai yang lebih kecil, yaitu 7. Kemudian lakukan metode yang dilakukan pada simpul B kepada simpul A dan D sehingga didapat bobot simpul A saat ini adalah 1 dan bobot simpul D saat ini adalah 2.

Karena seluruh simpul tetangga dari simpul C telah dicek, maka akan dilakukan pengecekan terhadap simpul yang lain. Misalkan simpul yang akan dicek adalah simpul A.

3. Mengecek semua simpul tetangga dari simpul A

Simpul	A	B	C	D	E
Bobot	1	<del>7</del> <b>4</b>	0	2	$\infty$

Tabel 3. Perubahan Bobot pada simpul tetangga dari C (simpul A, B, dan D)

Keterangan: **Merah**: Simpul saat ini  
**Hijau**: Simpul yang sudah dicek  
**Kuning**: Perubahan bobot pada graf

Untuk mengecek semua simpul tetangga dari simpul A, ulangi langkah 1 dan langkah 2 pada simpul A. Karena simpul C sudah dikunjungi, maka simpul tetangga yang belum dicek adalah simpul B. Pada Gambar 7 dapat dilihat bahwa sisi (A, B) berbobot 3. Tambahkan bobot sisi tersebut dengan bobot dari simpul A:  $1+3 = 4$ . Karena  $4 < 7$ , maka bobot simpul B kita ubah menjadi bernilai 4.

4. Ulangi langkah 1-3 untuk simpul lain yang belum dicek

Simpul	A	B	C	D	E
Bobot	1	4	0	2	5

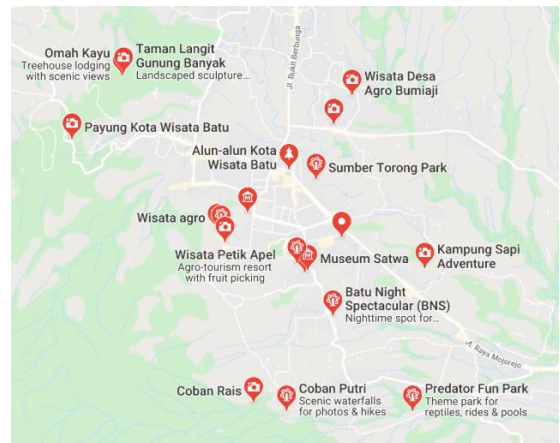
Tabel 4. Hasil akhir setelah semua simpul dicek

Keterangan: **Hijau**: Simpul yang sudah dicek

Setelah semua simpul dicek, didapat bahwa simpul A berbobot 1, simpul B berbobot 4, simpul C berbobot 0 (simpul awal), simpul D berbobot 2, dan simpul E berbobot 5. Bobot-bobot tersebut merupakan jarak minimum dari simpul C (sebagai simpul awal) menuju simpul terkait (simpul A, B, D, dan E).

2. Penentuan Jalur Pariwisata Kota Batu Menggunakan Algoritma Dijkstra

Kota Batu adalah kota dengan berbagai macam destinasi wisata yang tersebar di seluruh daerah di Kota Batu. Di Kota Batu terdapat wisata alam dan wisata buatan. Beberapa destinasi wisata memiliki letak yang berdekatan seperti pada gambar berikut ini:

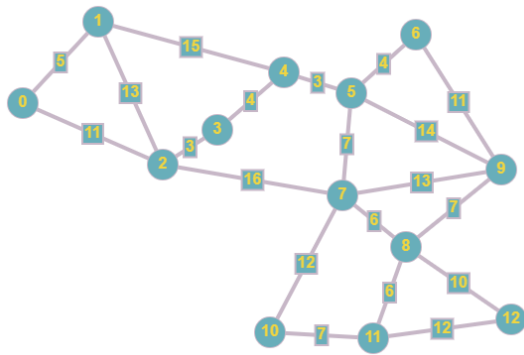


Gambar 6. Peta Tempat Wisata di Kota Batu

Sumber:

<https://www.google.com/maps/search/wisata+kota+batu/@-7.8771421,112.5025298,13.26z>

Oleh karena letaknya yang berdekatan, penulis memutuskan untuk menggambarkan destinasi yang berdekatan tersebut sebagai satu titik simpul yang sama. Graf hasil visualisasi dari peta tersebut adalah sebagai berikut:



Gambar 7. Graf visualisasi peta tempat wisata Kota Batu

Keterangan:

- Simpul ke- 0 : Payung Kota Wisata Batu
- 1 : Omah Kayu, Taman Langit Gunung Banyak
- 2 : Wisata Agro, Wisata Petik Apel
- 3 : Museum Angkut
- 4 : Alun-alun Kota Wisata Batu
- 5 : Sumber Torong Park
- 6 : Wisata Desa Argo Burniaji
- 7 : Museum Satwa
- 8 : Batu Night Spectacular (BNS)
- 9 : Kampung Sapi Adventure
- 10 : Coban Rais
- 11 : Coban Putri
- 12 : Predator Fun Park

Untuk menemukan jarak terpendek dari graf tersebut menggunakan algoritma djisktra, penulis mengasumsikan simpul awal berada pada simpul ke-4, yaitu alun-alun Kota Wisata Batu. Berikut merupakan tabel inisiasi dalam algoritma djisktra:

Simpul	Bobot
0	$\infty$
1	$\infty$
2	$\infty$
3	$\infty$
4	0
5	$\infty$
6	$\infty$
7	$\infty$
8	$\infty$
9	$\infty$
10	$\infty$
11	$\infty$
12	$\infty$

Tabel 5. Inisiasi Simpul 4 Sebagai Simpul Awal

Keterangan: **Merah**: Simpul saat ini

Setelah menginisiasi tabel dengan simpul 4 sebagai simpul awal, akan dilakukan pengecekan terhadap simpul tetangga dari simpul awal, yaitu simpul 1, 3, dan 5. Pertama-tama simpul yang akan dicek adalah simpul 1. Berdasarkan gambar 9 dapat diketahui bobot dari sisi (1, 4) adalah 15. Jika dibandingkan dengan bobot simpul 1 saat ini yaitu tak terhingga ( $\infty$ ), dapat

diketahui bahwa  $15 < \infty$ . Oleh karena itu, bobot pada simpul 1 akan diganti menjadi bernilai 15. Selanjutnya untuk simpul 3, diketahui dari gambar 9 bahwa sisi (3, 4) berbobot 4, sedangkan bobot pada simpul 3 bernilai  $\infty$ . Oleh karena  $4 < \infty$ , bobot pada simpul 3 diganti menjadi bernilai 4. Kemudian dilakukan perhitungan yang sama untuk simpul 5 dan didapatkan bobot dari simpul 5 berganti menjadi 3.

Simpul	Bobot
0	$\infty$
1	<del><math>\infty</math></del> 15
2	$\infty$
3	<del><math>\infty</math></del> 4
4	0
5	<del><math>\infty</math></del> 3
6	$\infty$
7	$\infty$
8	$\infty$
9	$\infty$
10	$\infty$
11	$\infty$
12	$\infty$

Tabel 6. Perubahan Bobot pada Simpul 1, 3, dan 5

Keterangan: **Merah** : Simpul saat ini  
**Kuning** : Perubahan bobot pada graf

Karena seluruh simpul tetangga dari simpul 4 telah dicek, maka akan dilakukan pengecekan terhadap simpul yang lain. Misalkan simpul yang akan dicek adalah simpul 1. Dapat dilihat pada gambar 9 bahwa simpul 1 bertetangga dengan simpul 0, 2, dan simpul 4. Karena simpul 4 telah dicek, maka tersisa simpul 0 dan simpul 2 untuk diperiksa jarak terdekatnya dari simpul 1.

Berdasarkan gambar 9, dapat dilihat bahwa jarak terdekat dari simpul 0 ke simpul 1 adalah sisi (0, 1) yang berbobot 5, dengan perhitungan terhadap bobot simpul 1 didapat bobot simpul 0 adalah  $5 + 15 = 20$ . Jika dibandingkan dengan bobot simpul 0 saat ini yaitu  $\infty$ , maka  $20 < \infty$  sehingga bobot dari simpul 0 menjadi bernilai 20. Sedangkan pada simpul 2, dapat diketahui bahwa jarak terdekat dari simpul 1 ke simpul 2 adalah sisi (1, 2) yang berbobot 13, dengan perhitungan sederhana didapat bobot simpul 2 adalah  $13 + 15 = 28$ . Oleh karena  $28 < \infty$ , maka bobot dari simpul 2 menjadi bernilai 28.

Simpul	Bobot
0	<del><math>\infty</math></del> 20
1	15
2	<del><math>\infty</math></del> 28
3	4
4	0
5	3
6	$\infty$
7	$\infty$
8	$\infty$
9	$\infty$
10	$\infty$
11	$\infty$

12	$\infty$
----	----------

Tabel 7. Perubahan Bobot pada Simpul 0 dan 2

Keterangan: **Merah** : Simpul saat ini  
**Hijau** : Simpul yang sudah dicek  
**Kuning** : Perubahan bobot pada graf

Dengan mengulangi langkah sebelumnya untuk simpul-simpul lain yang belum dicek, maka didapat tabel jarak terdekat adalah sebagai berikut:

Simpul	Bobot
0	18
1	15
2	7
3	4
4	0
5	3
6	7
7	10
8	16
9	17
10	22
11	22
12	26

Tabel 8. Tabel Akhir Jarak Terdekat Relatif terhadap Simpul 4 (Alun-alun Kota Batu)

Keterangan: **Merah** : Simpul saat ini  
**Hijau** : Simpul yang sudah dicek  
**Kuning** : Perubahan bobot pada graf

Dari tabel tersebut dapat diketahui bahwa jarak terpendek yang dapat ditempuh dari alun-alun Kota Batu yaitu adalah bobot paling minimum relatif terhadap bobot simpul 4. Oleh karena itu, objek wisata terdekat dari alun-alun Kota Batu adalah Sumber Torong Park (simpul 5), kemudian Museum Angkut (simpul 3), Wisata Agro, Wisata Petik Apel, atau Wisata Desa Agro Burniaji (simpul 2 atau 6), Museum Angkut (simpul 7), Omah Kayu atau Taman Langit Gunung Banyak (simpul 1), Kampung Sapi Adventure (simpul 9), Omah Kayu atau Taman Langit Gunung Banyak (simpul 0), Coban Rais atau Coban Putri (simpul 10 atau 11), dan yang terakhir, yang paling jauh adalah Predator Fun Park (simpul 12).

## VI. KESIMPULAN

Penentuan jalur pariwisata dapat diselesaikan menggunakan algoritma djisktra pada graf yang merepresentasikan peta Kota Batu. Dengan tiap simpul pada graf merepresentasikan setiap objek wisata dan bobot sisi yang menghubungkan simpul-simpul menyatakan jarak antar simpul, maka dapat diketahui jarak terpendek untuk sampai ke suatu objek wisata. Dengan asumsi wisatawan saat ini berada di Alun-alun Kota Batu, dapat diketahui jarak terpendek untuk mencapai objek wisata selanjutnya. Dengan hasil yang didapatkan pada pembahasan, pembaca dapat merancang suatu jadwal perjalanan wisata yang efektif dan efisien sehingga pembaca yang berwisata tidak perlu

menghabiskan banyak waktu untuk transportasi di jalan.

## VII. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena karunia-Nya telah memberikan kesempatan untuk menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada semua pihak yang telah membantu pengerjaan makalah ini baik secara langsung maupun tidak langsung. Secara khusus penulis mengucapkan terima kasih kepada Ibu Fariska, selaku dosen dari Mata Kuliah Matematika Diskrit kelas 03 yang telah membantu saya untuk memahami materi-materi mengenai matematika diskrit, khususnya teori graf. Selain itu, penulis juga mengucapkan terima kasih kepada seluruh sumber referensi yang telah membantu penulis menyelesaikan makalah ini.

## REFERENSI

- [1] BPK Provinsi Jawa Timur. "Kota Batu", (Online), [https://surabaya.bpk.go.id/?page\\_id=8299](https://surabaya.bpk.go.id/?page_id=8299), diakses 1 Desember 2019.
- [2] Girsang, Abba Suganda, "Algoritma Dijkstra", (Online), <https://mti.binus.ac.id/2017/11/28/algoritma-dijkstra/>, diakses 1 Desember 2019.
- [3] Munir, Rinaldi. "Matematika Diskrit Edisi 3". Informatika, Bandung: 2010.
- [4] Williams, Christopher. "Dijkstra's Shortest Path Algorithm", (Online), diakses 1 Desember 2019.

## 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, 4 Desember 2019



Tony Eko Yuwono  
13518030