

# Aplikasi Graf dan Algoritma Prim Untuk Membuat Rute Pemandu Wisata Bandung

Muhammad Ivan Rahmansyah M/13517143

*Program Studi Teknik Informatika*

*Sekolah Teknik Elektro dan Informatika*

*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*

*mivanrm@itb.ac.id*

**Abstrak**—Bandung adalah salah satu kota terbesar di Indonesia, yang menjadi ibukota Jawa Barat. Namun, selain itu Bandung juga merupakan sebuah kota wisata yang sangat terkenal dan ramai dikunjungi oleh wisatawan baik dalam negeri maupun luar negeri. Karena itulah banyak orang juga semakin tertarik untuk mengunjungi Bandung, tetapi ada suatu masalah yaitu jalanan di Bandung sangatlah macet. Apabila para wisatawan tidak menggunakan waktunya dengan baik, maka pasti akan terbuang percuma di jalanan. Oleh karena itu, saya membuat suatu aplikasi menggunakan graf dan algoritma Prim, agar bisa mengetahui tempat wisata apa saja yang bisa dijangkau dalam waktu yang singkat.

**Kata Kunci**—Graf, Waktu, Pohon, Algoritma Prim.

## I. PENDAHULUAN

Bandung, sebagai salah satu kota besar di Indonesia, memiliki jumlah penduduk yang padat seperti kota pada umumnya, tetapi ada hal yang berbeda dari Bandung ini, selain kota metropolitan, Bandung juga merupakan salah satu kota wisata terfavorit untuk dikunjungi orang pada akhir pekan. Hal ini didukung oleh suasana kota Bandung yang nyaman, dan juga banyak tempat-tempat menarik yang bisa dikunjungi disini.

Hal ini membawa dampak positif dan negatif, salah satu contoh dampak positif adalah semakin ramai orang yang mengunjungi dan menganggap Bandung adalah kota yang sangat indah, disisi lain banyaknya wisatawan membuat jalanan macet dan ramai, sehingga orang bingung mau pergi ke tempat apa.

Saking banyaknya tempat yang bisa dikunjungi, orang-orang tidak akan sempat untuk mengunjungi seluruh tempat-tempat tersebut dan banyak membuang waktu mereka di jalan

karena tidak tahu jalan yang benar.

Dan saat pelajaran matematika diskrit ini, saya mendapat pelajaran tentang algoritma Prim dan Graf, menurut saya pengaplikasian kedua hal ini dapat mengurangi waktu yang terbuang bagi para wisatawan, karena dengan menggunakan kedua hal ini, bisa membuat wisatawan lebih mudah untuk memilih dan mencapai tujuan wisata yang terdekat dengan mereka.

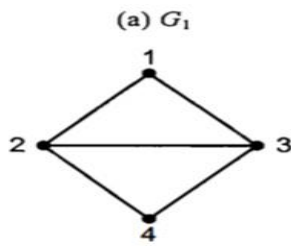
Solusinya adalah membuat sebuah pohon terentang minimum, yang dapat dibuat menggunakan pengaplikasian algoritma Prim pada graf peta wisata yang telah dibuat, sehingga nantinya tiap simpul memiliki waktu minimum untuk mencapai ke simpul terdekat lainnya.

## II. LANDASAN TEORI

### 1. Graf

Graf didefinisikan pasangan himpunan yang terdiri dari himpunan simpul (node) dan himpunan sisi (edge) yang menghubungkan simpul. Graf dapat ditulis dengan notasi  $G = (V, E)$  dengan  $V$  merupakan himpunan tidak kosong dari simpul dan  $E$  merupakan himpunan yang mungkin kosong dari sisi.

Definisi diatas menyatakan bahwa  $V$  tidak boleh kosong, sedangkan  $E$  boleh kosong. Jadi, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi simpulnya harus ada, minimal satu. Graf yang hanya mempunyai satu buah simpul tanpa sebuah sisi pun dinamakan graf trivia.



Gambar 1. Contoh graf dengan 4 simpul dan 5 sisi.

Simpul pada graf dapat dinomori dengan huruf, atau bilangan asli, atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul  $u$  dengan simpul  $v$  dinyatakan dengan pasangan  $(u, v)$  atau dinyatakan dengan lambang  $e_1, e_2, \dots$ . Dengan kata lain, jika  $e$  adalah sisi yang menghubungkan simpul  $u$  dengan simpul  $v$ , maka  $e$  dapat ditulis sebagai

$$e = (u, v)$$

Kardinalitas graf adalah jumlah simpul pada graf. Kardinalitas graf dapat dinyatakan dengan  $n = |V|$ , sedangkan jumlah sisi pada graf dinyatakan dengan  $m = |E|$ .

a. Jenis-jenis Graf

Graf dapat dikelompokkan menjadi beberapa kategori (jenis) bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi kalang, berdasarkan jumlah simpul, atau berdasarkan orientasi arah pada sisi.

Berdasarkan ada atau tidaknya sisi ganda:

1. Graf sederhana (simple graph).

Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana.  $G_1$  pada Gambar 8.3(a) adalah contoh graf sederhana yang merepresentasikan jaringan komputer. Simpul menyatakan komputer, sedangkan sisi menyatakan saluran telepon untuk berkomunikasi. Saluran telepon dapat beroperasi pada dua arah. Pada graf sederhana, sisi adalah pasangan tak-terurut (unordered pairs). Jadi, menuliskan sisi  $(u, v)$  sama saja dengan  $(v, u)$ . Kita dapat juga mendefinisikan graf sederhana  $G = (V, E)$  terdiri dari himpunan tidak kosong simpul-simpul dan  $E$  adalah himpunan pasangan tak-terurut yang berbeda yang disebut sisi.

Contoh dari graf sederhana ini adalah graf pada gambar 1.

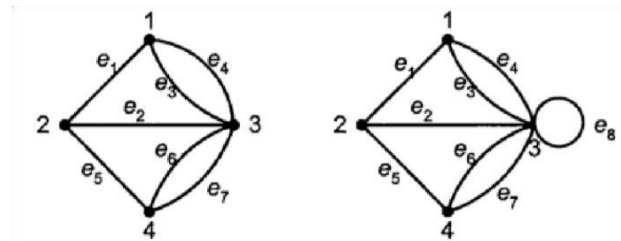
2. Graf tak-sederhana (unsimple-graph).

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (unsimple graph). Ada dua macam graftak-

sederhana, yaitu graf ganda (multigraph) dan graf semu (pseudograph).

Graf ganda adalah graf yang mengandung sisi ganda. Sisi ganda yang menghubungkan sepasang simpul bisa lebih dari dua buah. Sisi ganda dapat diasosiasikan sebagai pasangan tak-terurut yang sama.

Graf semu adalah graf yang mengandung gelang (loop).  $G_3$  adalah graf semu (termasuk bila memiliki sisi ganda sekalipun). Sisi gelang pada  $G_3$  dapat dianggap sebagai saluran telepon tambahan yang menghubungkan komputer dengan dirinya sendiri (mungkin untuk tujuan diagnostik). Graf semu lebih umum daripada graf ganda, karena sisi pada graf semu dapat



terhubung ke dirinya sendiri.

Gambar 2. Contoh graf tak sederhana, graf ganda(kiri), graf semu(kanan).

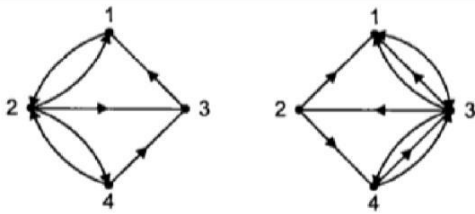
Berdasarkan orientasi arah pada sisi:

1. Graf tak berarah

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah. Pada graf tak-berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhalikan. Jadi,  $(u, v) = (v, u)$  adalah sisi yang sama. Graf berarah menyatakan bahwa saluran telepon dapat beroperasi pada dua arah.

2. Graf berarah (directed graph atau digraph)

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Kita lebih suka menyebut sisi berarah dengan sebutan busur (arc). Pada graf berarah,  $(u, v)$  dan  $(v, u)$  menyatakan dua buah busur yang berbeda, dengan kata lain  $(u, v) \neq (v, u)$ . Graf berarah sering dipakai untuk menggambarkan aliran proses, peta lalu lintas suatu kota Galan searah atau dua arah), dan sebagainya. Pada graf berarah, gelang diperbolehkan, tetapi sisi ganda tidak.



Gambar 3. Contoh graf berarah(kiri),graf berarah sederhana(kanan)

## 2. Terminologi Dasar Graf

### a. Bertetangga (Adjacent)

Dua buah simpul pada graf tak-berarah  $G$  dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain,  $u$  bertetangga dengan  $v$  jika  $(u, v)$  adalah sebuah sisi pada graf  $G$ . Pada graf berarah, sisi kita sebut busur. Jika  $(u, v)$  adalah busur maka  $u$  dikatakan bertetangga dengan  $v$  dan  $v$  dikatakan tetangga dari  $u$ .

### b. Bersisian

Sebuah sisi dikatakan bersisian dengan kedua simpul yang dihubungkannya. Dengan kata lain, sebuah sisi  $e = (u, v)$ , sisi  $e$  bersisian dengan simpul  $u$  dan simpul  $v$ .

### c. Simpul Terpencil (Isolated Vertex)

Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya. Atau, dapat juga dinyatakan bahwa simpul terpencil adalah simpul yang tidak satupun bertetangga dengan simpul-simpul lainnya.

### d. Graf Kosong

Graf Kosong merupakan graf yang kumpulan sisinya merupakan himpunan kosong, jadi Graf kosong tidak memiliki sisi satupun pada grafnya, dan semua simpulnya merupakan simpul terpencil.

### e. Derajat(Degree)

Pada sebuah graf tak berarah, jumlah sisi yang bersisian dengan simpul tersebut disebut dengan degree atau derajat dari simpul itu. Yang dinotasikan dengan  $d(v)$

Dimana  $d$  menyatakan simpul derajat dari  $v$ . Pada graf berarah, derajat suatu simpul dibedakan menjadi dua macam untuk mencerminkan jumlah busur dengan simpul tersebut sebagai simpul asal dan jumlah busur dengan simpul tersebut sebagai simpul terminal

pada graf berarah, derajat simpul  $v$  dinyatakan dengan  $d_{in}(v)$  dan  $d_{out}(v)$ , yang dalam hal ini  $d_{in}(v) = \text{derajat-masuk} = \text{jumlah busur yang masuk ke simpul}$  dan  $d_{out}(v) = \text{derajat-keluar} = \text{jumlah busur yang keluar dari simpul}$ .

Dari keduanya kita bisa mendapatkan,

$$d(v) = d_{in}(v) + d_{out}(v)$$

### f. Lintasan

Lintasan dengan panjang  $n$  dari simpul awal  $x$  ke simpul tujuan  $y$  dalam suatu graf adalah barisan simpul-simpul dan sisi-sisi yang menunjukkan arah dari simpul  $x$  ke simpul  $y$  melewati beberapa simpul dan beberapa sisi sebanyak  $n$ . Terdapat dua buah lintasan dalam graf, yaitu

- A. Lintasan Euler
- B. Lintasan Hamilton

### g. Sirkuit

Lintasan dengan titik awal, dan berakhir pula di titik awal tersebut, disebut dengan sirkuit. Sebuah sirkuit dikatakan sirkuit sederhana apabila hanya pernah melewati 1 sisi saja.

Sama seperti Lintasan, terdapat dua macam sirkuit dalam graf, yaitu

- A. Sirkuit Euler
- B. Sirkuit Hamilton

### h. Terhubung

Keterhubungan dua buah simpul adalah penting di dalam graf. Dua buah simpul  $u$  dan simpul  $v$  dikatakan terhubung jika terdapat lintasan dari  $u$  ke  $v$ . Jika dua buah simpul terhubung maka pasti simpul yang pertama dapat dicapai dari simpul yang kedua. Dua simpul terminal pada jaringan komputer hanya dapat berkomunikasi bila keduanya terhubung. Jika setiap pasang simpul di dalam graf terhubung, maka graf tersebut kita katakan graf terhubung

## 3. Pohon

Graf terhubung yang tidak mengandung sirkuit disebut pohon. Di antara sekian banyak konsep dalam teori graf, konsep pohon (tree) mungkin merupakan konsep yang paling penting, karena terapan yang luas dalam berbagai bidang ilmu. Banyak terapan, baik dalam bidang ilmu komputer maupun di luar bidang ilmu komputer, yang telah mengkaji pohon secara intensif sebagai objek matematik

### A. Sifat-sifat Pohon

Sifat-sifat (properties) pohon dinyatakan dengan teorema di bawah ini. Misalkan  $G = (V, E)$  adalah graf tak-berarah sederhana dan jumlah simpulnya  $n$ . Maka, semua pernyataan di bawah ini adalah ekuivalen

1.  $G$  adalah pohon.
2. Setiap pasang simpul di dalam  $G$  terhubung dengan lintasan tunggal.
3.  $G$  terhubung dan memiliki  $m = n - 1$  buah sisi.

4. G tidak mengandung sirkuit dan memiliki  $m = n - 1$  buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan. Jembatan adalah sisi yang bila dihapus menyebabkan graf terpecah menjadi dua komponen.

Jumlah langkah seluruhnya di dalam algoritma Prim adalah  $1 + (N - 2) = n - 1$ , yaitu sebanyak jumlah sisi di dalam pohon merentang dengan  $N$  buah simpul.

## B. Pohon Merentang

Misalkan  $G = (V, E)$  adalah graf tak-berarah terhubung yang bukan pohon, yang berarti di  $G$  terdapat beberapa sirkuit.  $G$  dapat diubah menjadi pohon  $T = (V, E')$  dengan cara memutuskan sirkuit-sirkuit yang ada. Caranya, mula-mula dipilih sebuah sirkuit, lalu hapus satu buah sisi dari sirkuit ini.  $G$  akan tetap terhubung dan jumlah sirkuitnya berkurang satu. Bila proses ini dilakukan berulang-ulang sampai semua sirkuit di  $G$  hilang, maka  $G$  menjadi sebuah pohon  $T$ , yang dinamakan pohon merentang (spanning tree).

## C. Pohon Merentang Minimum

Jika  $G$  adalah graf berbobot, maka bobot pohon merentang  $T$  dari  $G$  didefinisikan sebagai jumlah bobot semua sisi di  $T$ . Pohon merentang yang berbeda mempunyai bobot yang berbeda pula. Di antara semua pohon merentang di  $G$ , pohon merentang yang berbobot minimum - dinamakan pohon merentang minimum (minimum spanning tree) - merupakan pohon merentang yang paling penting. Pohon merentang minimum mempunyai terapan yang luas dalam praktek di dunia nyata ini.

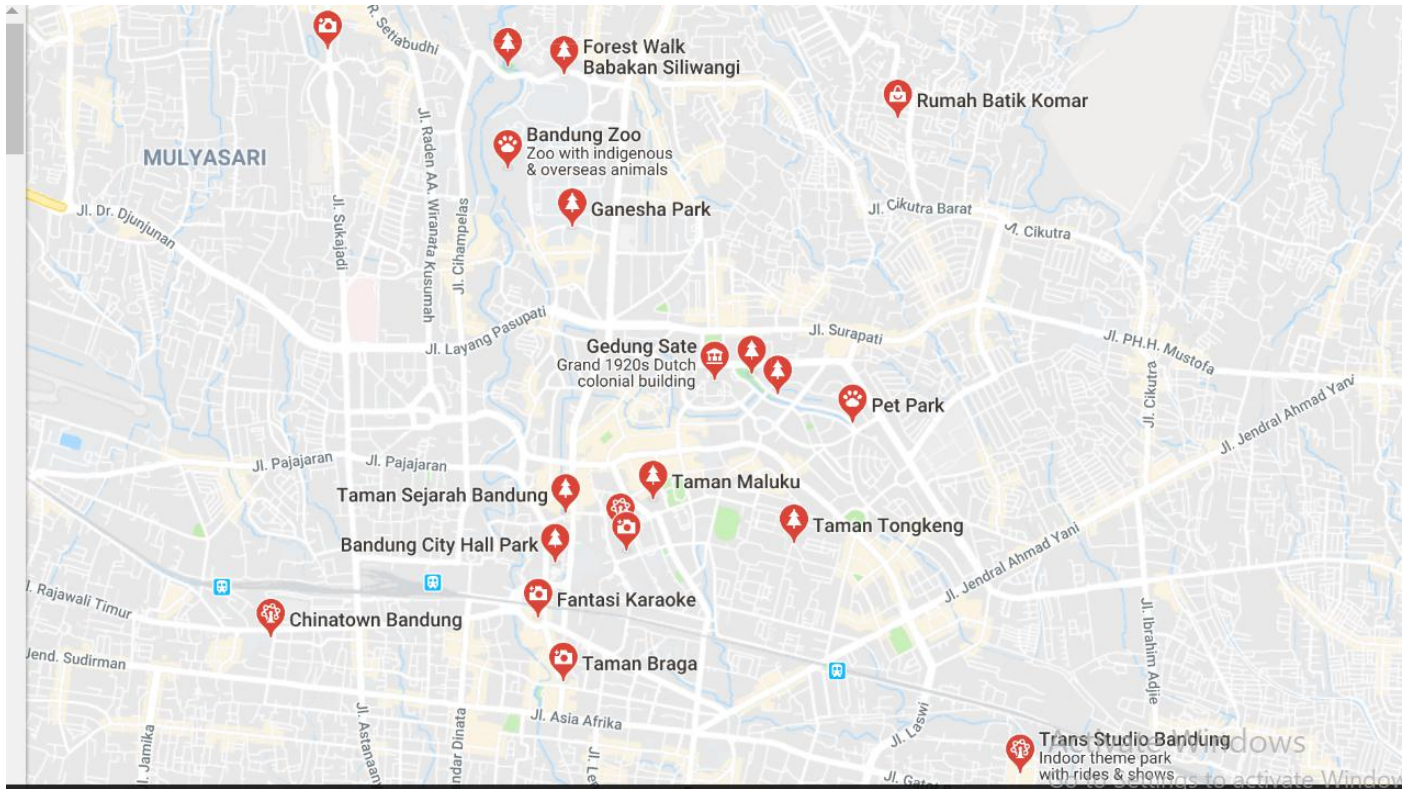
## D. Algoritma Prim

Misalkan  $T$  adalah pohon merentang yang sisinya diambil dari graf  $G$ . Algoritma Prim membentuk pohon merentang minimum langkah per langkah. Pada setiap langkah kita mengambil sisi  $e$  dari graf  $G$  yang mempunyai bobot minimum dan bersisian dengan simpul-simpul di dalam  $T$  tetapi  $e$  tidak membentuk sirkuit di dalam  $T$ .

### CARA MELAKUKAN ALGORITMA PRIM

1. Ambil sisi dari graf  $G$  yang berbobot minimum, masukkan ke dalam  $T$ .
2. Pilih sisi  $e$  yang mempunyai bobot minimum dan bersisian dengan simpul di  $T$ , tetapi  $e$  tidak membentuk sirkuit di  $T$ . Masukkan  $e$  ke dalam  $T$ .
3. Ulangi 2 sebanyak  $N - 2$  kali.

### III. PETA WISATA BANDUNG



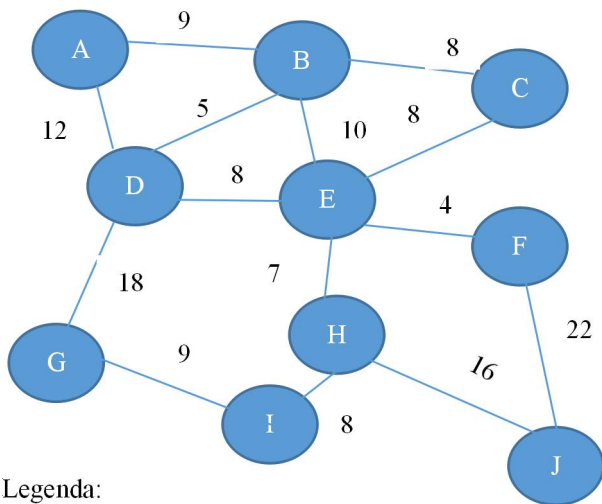
Mengubah dari peta tempat wisata Bandung menjadi sebuah graf berbobot, dengan bobotnya adalah waktu tempuh dari kedua tempat

G=Chinatown Bandung

H=Balai Kota Bandung

I=Jalan Braga

J=TransStudio Bandung



Legenda:

A=Paris Van Java

B=Hutan Babakan Siliwangi

C= Monumen Perjuangan/Gasibu

D=Kebun Binatang Bandung

E=Gedung Sate

F=Stadion Siliwangi

### IV. ANALISIS DAN PEMBAHASAN

Setelah Peta wisata telah diubah menjadi graf, terdapat waktu tempuh yang dimiliki antar tempat wisata yang berada di Bandung.

Untuk menentukan tempat apa yang terbaik untuk dikunjungi dari tempat sekarang, kita akan meminimumkan jalur-jalur yang tersedia pada graf, dengan menggunakan algoritma Prim. Algoritma ini mengubah dari bentuk suatu graf, menjadi minimum spanning tree.

Penerapan Algoritma Prim Pada Graf Peta Wisata:

Langkah dimulai dari simpul B.

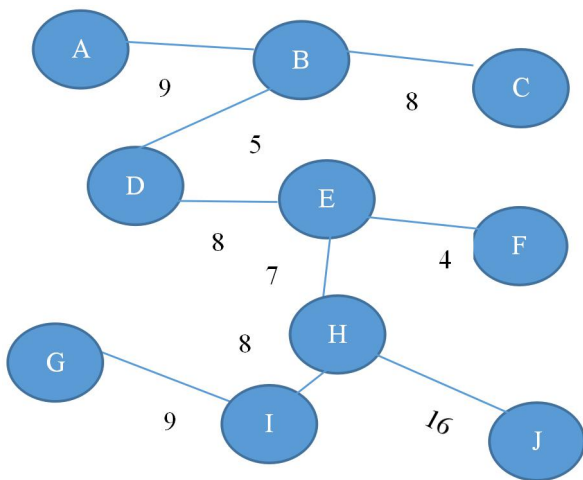
Langkah	Sisi	Bobot	Bobot Total
1	BD	5	5
2	BC	8	13
3	DE	8	21
4	EF	4	25
5	EH	7	32
6	HI	8	40

7	BA	9	49
8	IG	9	58
9	HJ	16	74

Tabel 1. Proses Algoritma Prim.

Pohon Merentang dapat dibuat dari hasil penghitungan menggunakan algoritma prim yang ada pada tabel 1. Pohon merentang ini membuat kita mengetahui tempat wisata apa saja yang paling dekat untuk kita kunjungi dari tempat kita sekarang.

Berikut adalah hasil minimum panning tree dari penerapan algoritma prim pada graf tempat wisata Bandung yang kita buat.



Legenda:

- A=Paris Van Java
- B=Hutan Babakan Siliwangi
- C= Monumen Perjuangan/Gasibu
- D=Kebun Binatang Bandung
- E=Gedung Sate
- F=Stadion Siliwangi
- G=Chinatown Bandung
- H=Balai Kota Bandung
- I=Jalan Braga
- J=TransStudio Bandung

Dari tree diatas kita bisa mengetahui, apabila sedang berada di titik E, maka pilihan rute terbaik adalah menuju D,F,atau H. Apabila berada di titik B maka rute terbaik selanjutnya adlaah menuju A,C,atau D. Rute ini menunjukkan kita tempat terdekat yang bisa kita capai dari tempat kita sekarang.

## V. CONCLUSION

Dari hasil diatas, dapat disimpulkan dengan menerapkan graf dan algoritma prim, kita dapat membantu menyelesaikan yaitu mempermudah dan mempercepat para pendatang yang baru datang ke Bandung.

Dapat dibentuk suatu pohon spanning minimum yang nantinya akan menjadi patokan suatu tempat, dan berapa lama waktu yang mereka butuh untuk sampai ke tempat wisata yang lain dengan berbagai macam rekomendasi tempat dengan waktu yang cepat.

## REFERENCES

- [1] Munir, Rinaldi. 2010. Matematika Diskrit, edisi 3 revisi keempat. Bandung: Informatika Bandung.
- [2] <http://7valley.blogspot.com/2014/04/algoritma-prim.html>
- [3] <https://www.google.com/maps>

## 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, 10 Desember 2018

Muhammad Ivan Rahmansyah Maulana

13517143