

# Implementasi Decision Tree dan *Prim Algorithm* dalam Pemilihan Tempat *Hangout* Mahasiswa

Muchammad Ibnu Sidqi 13518072  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13518072@stei.itb.ac.id

**Abstract**— Sejatinya mahasiswa sebagai suatu aspek dalam tatanan sosial tentu pernah berada dalam suatu titik jenuh atau suatu titik yang mana mahasiswa membutuhkan suatu insight atau suatu inspirasi sehingga dominannya mahasiswa dalam mencari tempat *hangout* untuk sekedar melepas penak sekaligus memenuhi kebutuhan mahasiswa sebagai seorang manusia yang tidak terlepas dari hakikat makhluk sosial ataupun mencari suatu insight untuk memenuhi kebutuhannya. Banyak faktor yang memengaruhi pemilihan tempat mulai dari jarak, estetika, hingga harga yang tentu akan melekat pada tempat tersebut sehingga penulis termotivasi untuk menerapkan *Decision Tree* dan *Prim Algorithm* dalam pemilihan tempat tersebut agar pemilihan menjadi mangkus serta memuat semua pilihan yang diinginkan mahasiswa.

**Keywords**— *Decision Tree*, *Hangout*, Mahasiswa, *Prim Algorithm*.

## I. PENDAHULUAN

Manusia dapat dikategorikan sebagai mahasiswa bila dalam prosesnya sedang menempuh pendidikan di perguruan tinggi untuk meraih ilmu serta agar dapat dikategorikan kembali menjadi insan akademis.

Dalam menjalani kehidupan perkuliahan mahasiswa seringkali mengalami penat serta mengalami hal yang tidak terduga sehingga mahasiswa memerlukan suatu tempat untuk sekedar melepas penat hingga memenuhi kebutuhan ego serta kebutuhan sebagai makhluk sosial.

Dalam pemilihan tempat *hangout*, dominannya mahasiswa memiliki *list* pilihan tempat masing-masing mulai dari estetika tempat, fasilitas yang didapat, harga yang melekat pada tempat tersebut hingga jarak yang akan ditempuh untuk mengunjungi tempat tersebut.

*Decision Tree* atau pohon keputusan adalah suatu metode prediksi yang digunakan untuk menangani keputusan-keputusan yang akan terjadi dengan menggunakan implementasi struktur dari *Tree* sehingga masing-masing keputusan yang akan terjadi dapat dianalisis dan didapatkan suatu kesimpulan yang tepat.

*Prim Algorithm* atau Algoritma Prim adalah suatu tatanan langkah atau sebuah algoritma yang digunakan untuk mengoptimasi serta membuat pohon berentang dari sebuah graf sirkuit dengan besar graf yang minimum.

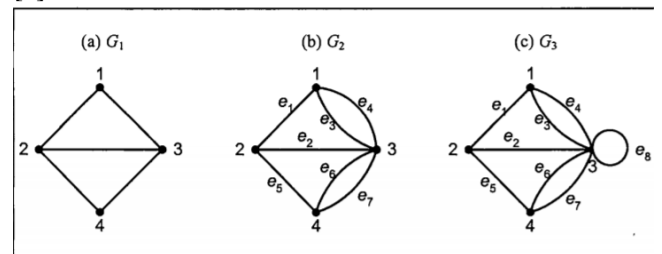
Untuk itu diperlukan *Decision Tree* untuk penanganan *list* pilihan tempat masing-masing pada mahasiswa serta *Prim*

*Algorithm* untuk optimasi jarak yang ditempuh untuk mencapai tempat tersebut sehingga pilihan yang dihasilkan akan lebih optimal dan memberikan keuntungan yang sebesar-besarnya bagi mahasiswa.

## II. TEORI DASAR

### A. Graf

graf (graph)  $G$  dapat didefinisikan pasangan himpunan  $(V, E)$ , dapat dinotasikan  $G = (V, E)$ , dengan  $V$  merupakan himpunan tidak kosong dari simpul-simpul (vertices atau nodes) dan  $E$  adalah himpunan sisi atau busur (edges atau arcs) yang menghubungkan sepasang simpul atau node [1].



Gambar 8.3 Tiga buah graf (a) graf sederhana, (b) graf ganda, dan (c) graf semu

Gambar 1 Contoh Graf [1]

### A. Jenis Graf

- Berdasarkan ada tidaknya gelang atau busur ganda pada graf, maka graf dapat digolongkan menjadi:

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

Graf yang tidak memiliki gelang atau busur ganda disebut sebagai graf sederhana.  $G_1$  pada Gambar 1 adalah contoh graf sederhana.

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

Graf yang memiliki busur ganda atau gelang disebut graf tak-sederhana (*unsimple graph*).  $G_2$  dan  $G_3$  pada Gambar 1 adalah contoh graf tak-sederhana.

- Berdasarkan adanya arah pada busur, maka secara umum graf dapat dibedakan atas 2

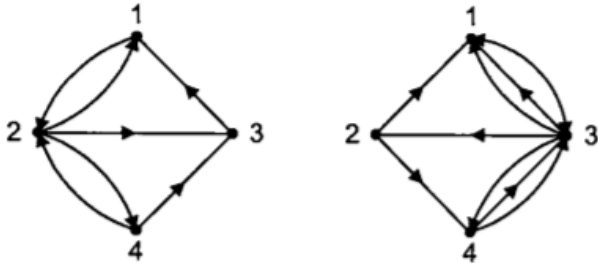
jenis:

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

Graf yang setiap busurnya tidak memiliki arah disebut graf tak-berarah. Tiga buah graf pada Gambar 1 adalah graf tak-berarah.

2. *Graf berarah (directed graph atau digraph)*

Graf yang setiap busurnya memiliki arah disebut graf berarah. Dua buah graf pada Gambar 2 adalah graf berarah [1].



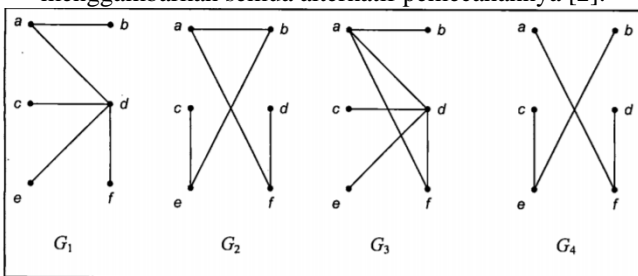
Gambar 2 : Contoh Graf berarah [1]

**B. Tree**

Tree (Pohon) yakni suatu graf tidak memiliki arah yang terhubung tetapi tidak mengandung sirkuit. Contohnya suatu graf tidak berarah sederhana  $G = (V, E)$  yang memiliki simpul berjumlah  $n$ , maka semua pernyataan di bawah ini adalah ekuivalen:

1.  $G$  merupakan *Tree*.
2. Setiap pasang *node* di  $G$  memiliki keterhubungan dengan lintasan tunggal.
3.  $G$  memiliki keterhubungan dan mempunyai  $m = n - 1$  busur.
4.  $G$  tidak membentuk sirkuit.
5. Penambahan satu busur pada pohon akan menyebabkan terbentuknya 1 sirkuit.
6.  $G$  saling terhubung dan setiap busur adalah jembatan (jembatan adalah busur jika dihapus akan menyebabkan pohon terpecah menjadi dua struktur). [1]

*Tree Diagram* dapat digunakan sebagai sebuah alat untuk memaparkan logika persoalan dengan menggambarkan semua alternatif pemecahannya [2].



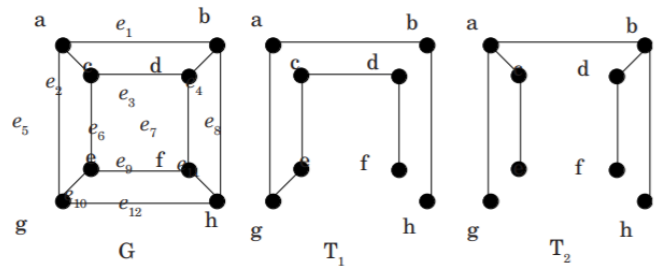
Gambar 9.1  $G_1$  dan  $G_2$  adalah pohon, sedangkan  $G_3$  dan  $G_4$  bukan pohon

Gambar 3: Contoh *Tree* (Pohon) [1]

**A. Spinning Tree (Pohon Merentang)**

sebuah pohon misalkan  $T$  disebut

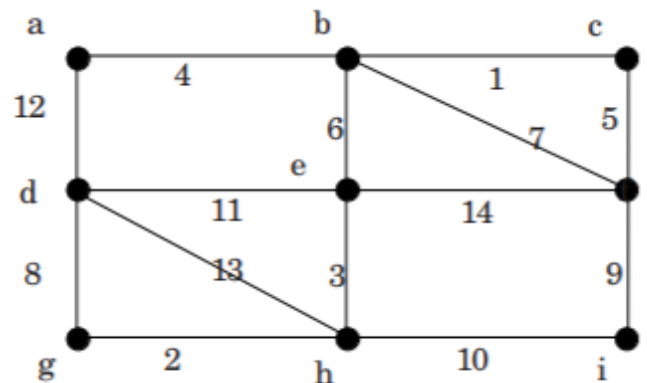
spanning tree dari sebuah graph  $G$ , jika  $T$  adalah subgraph dari  $G$  yang mencakup semua titik graph  $G$  [2].



Gambar 4: Graf  $G$  dengan contoh  $T_1$ , dan  $T_2$  sebagai *spanning tree* [2].

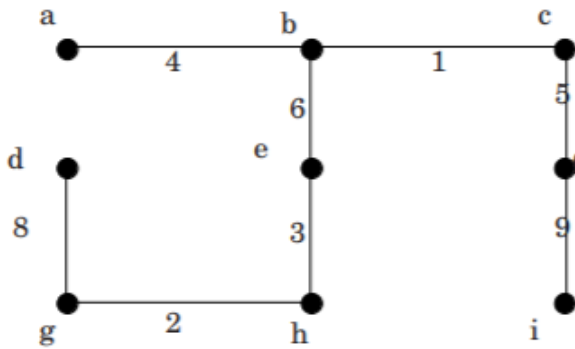
- *Minimum Spanning Tree*

*Tree* (Pohon) merentang dari suatu graf yang memiliki besar atau bobot minimum disebut sebagai *minimum spanning tree* atau pohon berentang minimum [1]. Misalkan pada suatu graf dibawah titik-titik yang merepresentasikan kota dan rusuk merepresentasikan jaringan jalan raya yang akan dibangun dengan bobot / label merepresentasikan rencana biaya antar kota, maka untuk mencari biaya minimal rencana pembuatan jalan menghubungkan semua kota diperlukan perubahan graf menjadi pohon berentang minimum.



Gambar 5: Contoh graf rencana biaya untuk menghubungkan antar kota [2].

Didapatkan bahwa minimal pohon berentang yang dibutuhkan untuk menggambarkan jaringan jalan raya yang menghubungkan 9 kota dengan biaya minimum adalah 38 [2].

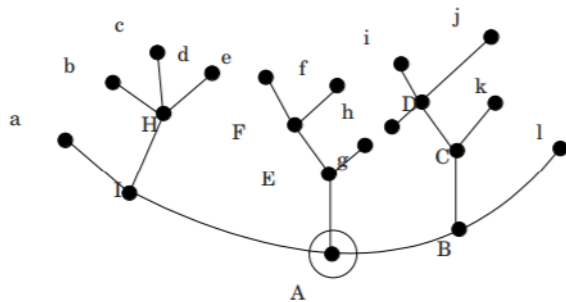


Gambar 6: Graf berentang minimum dari gambar 5 [2].

**B. Rooted Tree (Pohon Berakar)**

Pohon(*Tree*) yang salah satu simpulnya ditentukan sebagai sebuah akar dan setiap busurnya diberikan arah menjauh dari akar tersebut maka pohon(*Tree*) tersebut dapat dinamakan sebagai pohon berakar (*rooted tree*)[1].

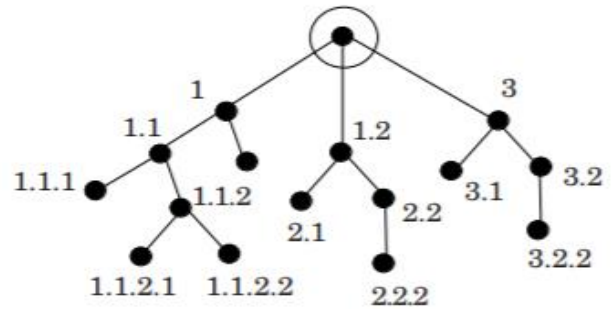
Seperti *Tree*(pohon) dalam dunia nyata, dalam graf pohon berakar juga mempunyai akar, cabang, dan daun. Akar dari suatu pohon adalah simpul yang derajat masuknya sama dengan nol (simpul sumber). Setiap simpul dapat dianggap atau dijadikan akar, simpul yang dianggap sebagai akar akan ditandai dengan lingkaran yang mengelilingi titik tersebut. Daun dari suatu pohon adalah setiap simpul yang tentunya bukan akar yang memiliki derajat masuk 1 serta derajat keluar nol. Tinggi pohon adalah panjang dari busur atau sisi yang bersinggungan dari akar hingga daun yang besarnya atau bobotnya maksimum[2].



Gambar 7: Contoh pohon berakar dengan akar pada simpul A[2].

**C. Ordered Tree (Pohon Terurut)**

Pohon terurut (*Ordered Tree*) adalah sebuah pohon yang memiliki akar dan diberi sebuah label yang terurut serta sistematis, dimulai dari akar sebagai simpul awal, semua cabang dari simpul awal diberi nomor berurut seperti 1,2,3, ... sesuai dengan banyak cabang. Sistem tersebut dapat dinamakan sebagai *universal adress system* [2].



Gambar 8: Contoh *Lexiographic Order* yang merupakan penerapan dari *Ordered Tree*[2].

**D. N-ary Tree (Pohon n-ary)**

*N-ary Tree* atau Pohon n-ary adalah sebuah pohon berakar yang setiap titik atau node pada cabang memiliki maksimal *n* anak. Pohon n-ary disebut sebagai pohon yang memiliki keteraturan (*full*) jika pada setiap node cabangnya memiliki *n* buah anak[1].

**C. Prim Algorithm**

Algoritma Prim atau Prim Algorithm adalah sebuah algoritma yang dibuat untuk digunakan dalam pencarian pohon merentang minimum atau minimum spanning tree dari sebuah graf sirkuit. Algoritma tersebut ada sejak tahun 1930 yang dibuat oleh seorang matematikawan bernama Vojtěch Jarník. Dalam implementasinya terdapat beberapa langkah.

Langkah pertama ambil edge dari suatu graf *G* yang memiliki besar atau bobot minimum lalu masukkan ke dalam *T*. Langkah kedua yaitu pilihlah edge(*u,v*) yang memiliki besar atau bobot yang minimum serta bersisian dengan node di *T*, akan tetapi (*u,v*) bila disatukan dalam *T* tidak akan membentuk suatu sirkuit. Lalu, masukkan pada *T*. Langkah terakhir adalah ulangi langkah kedua sebanyak *n-2* kali[1].

**D. Decision Tree**

*Decision Tree* atau pohon keputusan adalah suatu metode prediksi yang digunakan untuk menangani keputusan-keputusan yang akan terjadi dengan menggunakan implementasi struktur dari *Tree* sehingga masing-masing keputusan yang akan terjadi dapat dianalisis dan didapatkan suatu kesimpulan yang tepat. *Decision Tree* merupakan sebuah metode yang menggunakan penerapan dari pohon berakar (*Ordered Tree*). Dalam penggunaannya, *Decision Tree* bisa saja menggunakan *N-ary Tree* bisa juga tidak sehingga harus disesuaikan dengan keputusan apa yang akan ditangani. Penerapan *Decision Tree* sangat beragam salah satunya pada penentuan tempat *Hangout* mahasiswa.

**III. METODOLOGI DAN PENERAPAN**

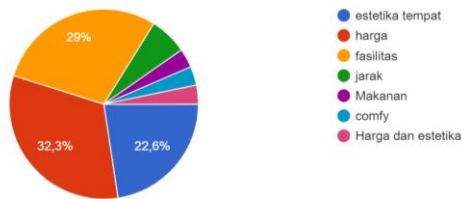
**A. Pengambilan Data**

**1. Data Mahasiswa**

Dalam karya tulis ini, penulis menggunakan mahasiswa sebagai basis sehingga penulis mengadakan survei dengan bantuan *google form* kepada beberapa

mahasiswa ITB serta mahasiswa kampus lain untuk menghimpun *list* yang terkait pada tempat *hangout* sehingga didapatkan data sebanyak 31 orang,

Apa yang menjadi faktor penentu tempat kalo kalian hangout?  
31 tanggapan



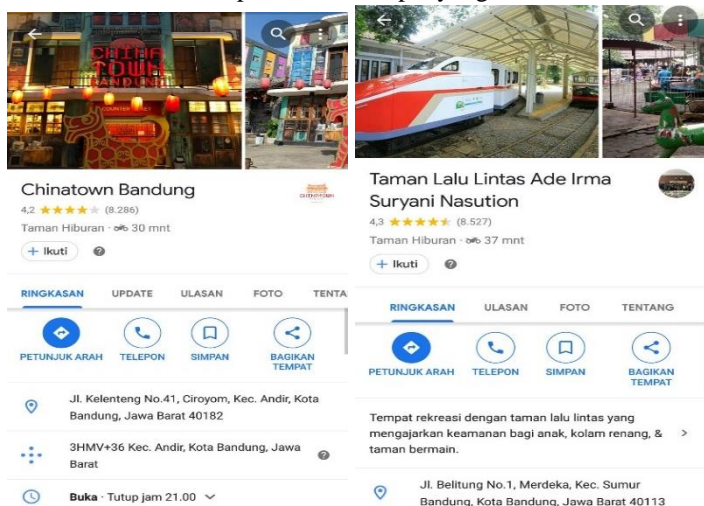
Gambar 9: Data Faktor penentu tempat *hangout* mahasiswa.

(Sumber: dokumen pribadi.)

10 dari 31 mahasiswa memilih tempat *hangout* berdasarkan harga, 9 dari 31 mahasiswa memilih tempat *hangout* berdasarkan fasilitas yang disuguhkan. 7 dari 31 mahasiswa memilih tempat *hangout* berdasarkan estetika tempat tersebut, 2 dari 31 mahasiswa memilih tempat *hangout* berdasarkan jarak, 1 dari 31 mahasiswa memilih tempat *hangout* berdasarkan makanan, 1 dari 31 mahasiswa memilih tempat *hangout* berdasarkan kenyamanan tempat, dan 1 dari 31 mahasiswa memilih tempat *hangout* berdasarkan harga serta estetika. Untuk itu, penulis memilih untuk menerapkan algoritma prim dan *Decision Tree* pada tempat *hangout* yang memiliki harga serta fasilitas yang bersahabat.

## 2. Data Tempat Hangout

Dari data yang diperoleh pada data mahasiswa, penulis akan menganalisis mengenai tempat *hangout* yang berdasarkan harga serta fasilitas yang bersahabat. Dalam pencarian melalui *google*, penulis mendapatkan beberapa tempat yang akan dianalisis yaitu kafe/restoran, taman, museum, tempat hiburan dan wisata alam. Dari keempat jenis tempat tersebut penulis akan memilih tempat hiburan. Salah satu tempat hiburan yang akan dianalisis adalah tempat hiburan yang memiliki *rating* yang cukup tinggi di aplikasi *google maps* yaitu Chinatown Bandung, serta Taman Lalu Lintas Ade Irma Suryani Nasution. Berikut cuplikan dari tempat yang akan dianalisis:



Gambar 10: Cuplikan Tempat yang akan dianalisis.

(Sumber: Google Maps.)

## B. Analisis Tempat dengan Decision Tree

Dengan menggunakan *decision tree*, akan dianalisis dua tempat tersebut dengan memperhatikan hanya dua buah komponen yaitu harga dan fasilitas yang akan diberikan. Dalam analisis menggunakan *decision tree*, dapat menggunakan algoritma salah satunya adalah algoritma berikut dalam memilih tempat berdasarkan harga, banyak fasilitas, serta *rating* yang ada.

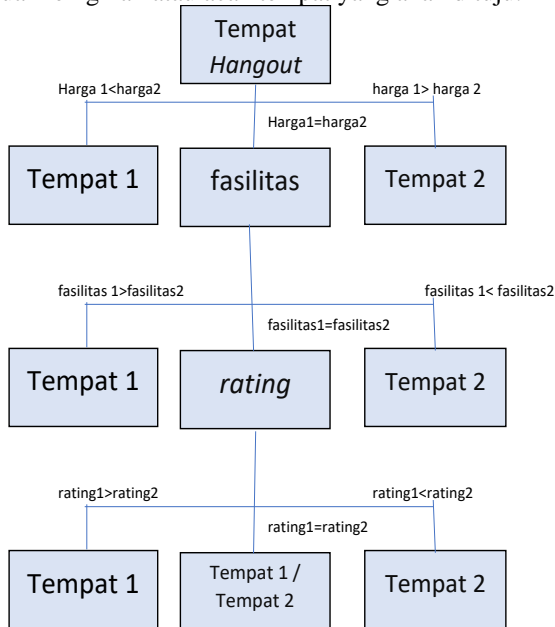
```
{ Notasi Algoritmik pada pemilihan tempat yaitu chinatown dan taman lalu lintas dengan decision tree }
KAMUS LOKAL
constant N          : ...
harga1,harga2,rating1,rating2 : integer
fasilitas1,fasilitas2 : array[1..N]of string
take                : integer {untuk memilih tempat}
ALGORITMA
depend on harga1,harga2:
  harga1>harga2: take <- 2
  harga1<harga2: take <- 1
  harga1=harga2:
    depend on fasilitas1,fasilitas2:
      len(fasilitas1)>len(fasilitas2): take <- 1 { len untuk mengambil banyak elemen dari fasilitas }
      len(fasilitas1)<len(fasilitas2): take <- 2
      len(fasilitas1)=len(fasilitas2):
        depend on rating1,rating2:
          rating1>rating2: take <- 1
          rating1<rating2: take <- 2
          rating1=rating2: take <- choose[1,2] { akan dipilih secara bebas bila ketiga indikator sama }
```

Dalam notasi algoritmik tersebut, dapat dilihat secara seksama bahwa bila harga yang diharuskan untuk Chinatown Bandung lebih besar dari Taman Lalu Lintas Ade Irma Suryani Nasution maka keputusan yang akan diambil yaitu memilih tempat Taman Lalu Lintas Ade Irma Suryani Nasution sebagai tempat *hangout* sedangkan bila harga yang diharuskan untuk Chinatown Bandung lebih kecil dibandingkan Taman Lalu Lintas Ade Irma Suryani Nasution maka keputusan yang akan diambil adalah memilih tempat Chinatown Bandung sebagai tempat *hangout*. Bila didapatkan hasil bahwa harga yang diperuntukkan untuk Chinatown Bandung sama dengan Taman Lalu Lintas Ade Irma Suryani Nasution maka akan dilakukan analisis lanjutan yaitu dengan menganalisis fasilitas.

Pada analisis fasilitas diterapkan hal yang sama yaitu bila fasilitas yang didapatkan untuk Chinatown Bandung lebih banyak dari Taman Lalu Lintas Ade Irma Suryani Nasution maka keputusan yang akan diambil yaitu memilih tempat Chinatown Bandung sebagai tempat *hangout* sedangkan bila fasilitas yang didapatkan untuk Chinatown Bandung lebih sedikit dibandingkan Taman Lalu Lintas Ade Irma Suryani Nasution maka keputusan yang akan diambil adalah memilih tempat Taman Lalu Lintas Ade Irma Suryani Nasution sebagai tempat *hangout*. Bila didapatkan hasil bahwa fasilitas yang didapatkan untuk Chinatown Bandung sama dengan Taman Lalu Lintas Ade Irma Suryani Nasution maka akan dilakukan analisis lanjutan yaitu dengan menganalisis *rating*.

Pada analisis *rating* pun diterapkan hal yang sama yaitu bila *rating* yang didapatkan untuk Chinatown Bandung

lebih banyak dari Taman Lalu Lintas Ade Irma Suryani Nasution maka keputusan yang akan diambil yaitu memilih tempat Chinatown Bandung sebagai tempat *hangout* sedangkan bila *rating* yang didapatkan untuk Chinatown Bandung lebih sedikit dibandingkan Taman Lalu Lintas Ade Irma Suryani Nasution maka keputusan yang akan diambil adalah memilih tempat Taman Lalu Lintas Ade Irma Suryani Nasution sebagai tempat *hangout*. Bila didapatkan hasil bahwa *rating* yang didapatkan untuk Chinatown Bandung sama dengan Taman Lalu Lintas Ade Irma Suryani Nasution maka penanganan keputusan yang akan diambil adalah memilih sesuai keinginan atau acak tempat yang akan dituju.

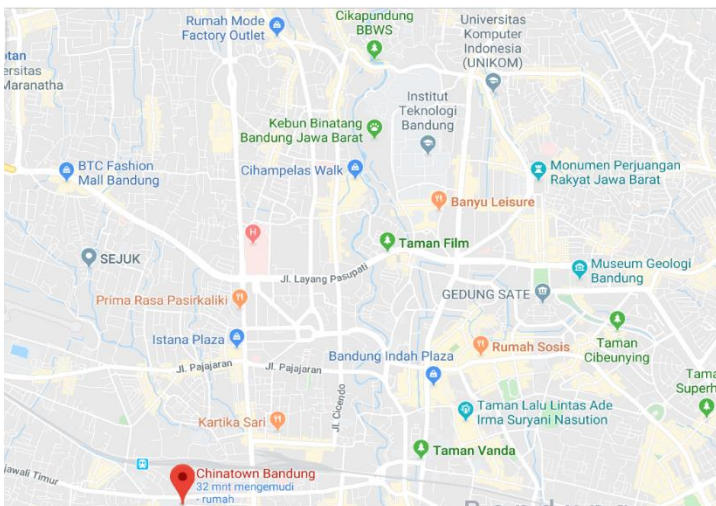


Gambar 11: Decision Tree dengan 3-ary Tree pada pemilihan tempat *hangout*.  
(Sumber: Dokumen pribadi).

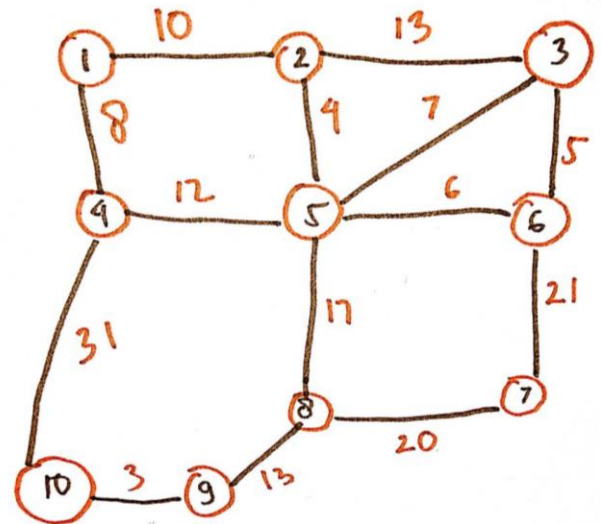
### C. Analisis Jarak dengan Prim Algorithm

Dalam analisis jarak dengan *Prim Algorithm*, penulis mengambil tempat Chinatown Bandung untuk dianalisis. Berikut peta Chinatown Bandung bila diambil dengan menggunakan *google maps*:

Gambar 12: Peta Chinatown Bandung.

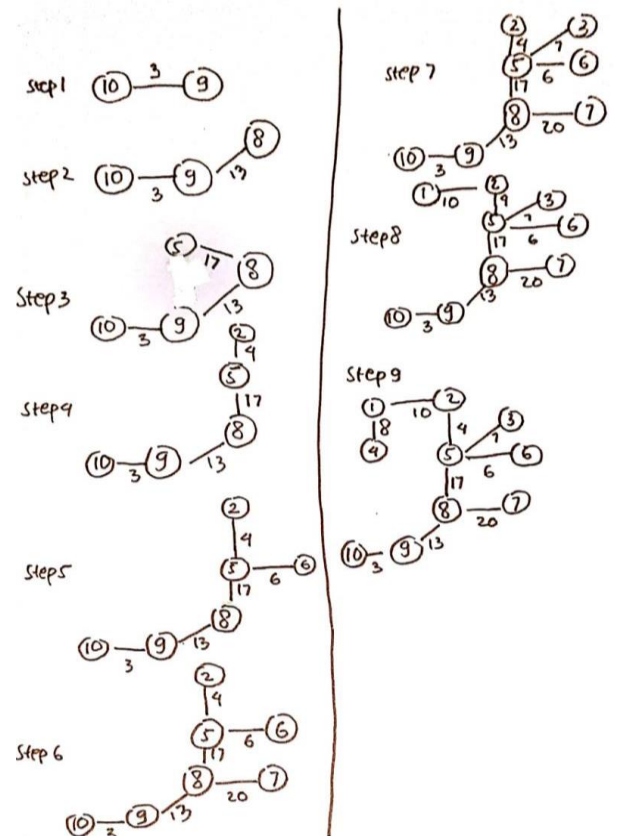


Dengan mengubah peta tersebut menjadi suatu graf sirkuit dengan bobot masing masing simpul adalah jarak. Maka didapatkan graf berupa:



Gambar 12: Graf pada peta Chinatown Bandung.  
(Sumber: Dokumen pribadi).

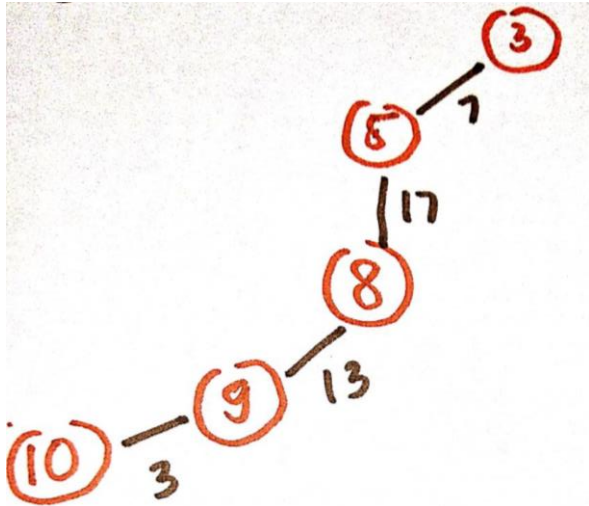
Dengan keterangan bahwa simpul 1,2,4,5,6,7,8,9 merupakan simpul tikungan pada jalan; simpul 3 merupakan ITB (simpul awal); simpul 10 merupakan Chinatown Bandung. Maka dengan menggunakan *Prim Algorithm*,



Gambar 13: Prim Algorithm step pada graf Chinatown Bandung.  
(Sumber: Dokumen pribadi).

Didapatkan pohon rentang minimum yaitu pada *step* akhir atau *step* 9, akan tetapi bila diamati dengan seksama terjadi beberapa jalan (busur) yang membuat simpul awal memiliki jarak tempuh yang lebih jauh saat mengunjungi simpul 6,7,2,1, serta 4. Untuk itu, perlu dilakukan modifikasi pada *Prim Algorithm* dengan membuang beberapa simpul sekaligus beberapa busur yang terikat langsung dengan simpul tersebut agar jalan yang ditempuh menjadi mangkus.

Dengan membuang simpul-simpul tersebut maka akan didapatkan jarak tempuh yang baru serta sesuai dengan efisiensi yang ada.

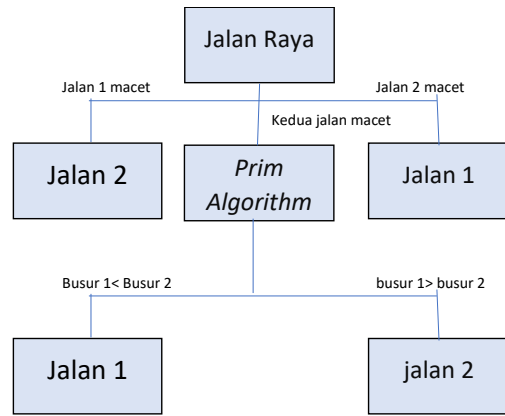


Gambar 14: Modifikasi pada *Spanning Minimum Tree*.  
(Sumber: Dokumen pribadi).

#### D. Penggunaan *Prim Algorithm* dan *Decision Tree* pada Kemacetan

Tentu dalam setiap keseharian, manusia tidak bisa lepas dari hubungan sosial atas konsekuensi manusia sebagai makhluk sosial. Untuk itu, manusia melakukan perpindahan jarak agar dapat melakukan komunikasi. Tidak jarang, setiap orang memiliki cara masing-masing untuk melakukan komunikasi. Agar efisien, beberapa orang melakukan komunikasi jarak jauh karena dinilai sudah cukup untuk memenuhi inti dari komunikasi yaitu ketersediaan informasi. Tetapi, beberapa orang tetap harus melakukan perpindahan tersebut agar kebutuhan hidupnya tercapai seperti dalam melakukan pekerjaan. Perpindahan manusia dari satu tempat ke tempat lain, baik seorang maupun beberapa orang, secara teratur dan sistematis dapat disebut dengan mobilisasi. Mengetahui hal tersebut, penulis termotivasi untuk menerapkan secara langsung *Decision Tree* serta *Prim Algorithm* dalam mobilisasi agar mobilisasi berjalan secara mangkus.

Implementasi pada mobilisasi dapat dilakukan dengan menggunakan *Prim Algorithm* terlebih dahulu untuk mendapatkan busur yang berbobot minimal lalu dengan menggunakan *Decision Tree* untuk mempertimbangkan apakah busur yang didapat dari *Prim Algorithm* dapat digunakan secara langsung.



Gambar 15: Implementasi *Decision Tree* dan *Prim Algorithm* pada kemacetan jalan raya.  
(Sumber: Dokumen pribadi).

## IV. KESIMPULAN

Manusia sejatinya tidak dapat lepas dari interaksi sosial sehingga untuk memenuhi kebutuhannya tidak jarang beberapa dari mereka melakukan mobilisasi salah satunya mobilisasi untuk mencari *insight*, wawasan, hingga mencari tempat untuk melepaskan penat yakni mencari tempat *hangout*. Untuk itu, penulis menerapkan metode *Decision Tree* dalam pemilihan tempat *hangout* untuk menangani beberapa keputusan akan faktor-faktor yang melekat pada tempat *hangout* tersebut. Tidak lupa juga, penulis menerapkan *Prim Algorithm* agar jarak yang ditempuh saat mengunjungi tempat tersebut menjadi minimum sehingga diharapkan mobilisasi yang berlangsung dapat berjalan dengan mangkus. Selain itu, penulis juga menerapkan *Decision Tree* serta *Prim Algorithm* untuk optimisasi waktu tempuh saat terjadi kemacetan. Diharapkan, dengan adanya karya tulis ini dapat menjadi bahan refleksi bagi pembaca serta penulis agar ketersediaan informasi dapat berlangsung dengan validitas yang mumpuni. Untuk itu, penulis menerima kritik serta saran atas karya tulis ini agar karya tulis ini dapat berguna serta tidak adanya kesalahpahaman atau kesalahan informasi dalam penyampaian.

## IV. UCAPAN TERIMAKASIH

Atas berkat rahmat Tuhan Yang Maha Esa pengerjaan karya tulis "*Implementasi Decision Tree dan Prim Algorithm dalam Pemilihan Tempat Hangout Mahasiswa*" dapat berlangsung dengan baik. Untuk itu, penulis juga mengucapkan terimakasih yang tidak terhingga kepada Fariska Zakhralatifa Ruskanda, S.T., M.T. selaku dosen mata kuliah matematika diskrit dan semua kerabat yang telah memberi saran kritik atas keberjalanan pembuatan karya tulis ini.

## REFERENCES

- [1] Munir, Rinaldi. 2010. Matematika Diskrit, edisi 3 revisi keempat. Bandung: Informatika Bandung.
- [2] Wibisono, Samuel, 2008. Matematika Diskrit, edisi 2, Yogyakarta: Graha Ilmu.

## 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, 5 Desember 2019



Muchammad Ibnu Sidqi  
13518072