# Aplikasi Algoritma Prim dan Lintasan Hamilton dalam Penentuan Jadwal Balapan F1 Agar Lebih Efisien

Fauzan Yubairi Indrayadi 13519171[1]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
[1]*13519171@std.stei.itb.ac.id*

***Abstraksi*—Formula Satu atau F1 sebagai ajang olahraga balap mobil termahal memiliki jadwal balapan yang tidak terurut antar kota terdekat. Hal tersebut dapat menambah waktu dan biaya operasional melebihi yang diperlukan. Dalam masa pandemi COVID-19, akan lebih efisien jika biaya yang dikeluarkan berkurang. Dengan aplikasi algoritma Prim dapat dibentuk lintasan Hamilton berbobot minimum untuk menentukan jadwal balapan yang lebih efisien berdasarkan bobot rute yang minimum.**

***Keywords*—Algoritma Prim, Formula Satu, jadwal balapan, lintasan Hamilton.**

## I. PENDAHULUAN

Formula Satu atau F1 adalah ajang olahraga balap mobil kursi tunggal kelas tertinggi yang terkenal dan disebut termahal. Balapan Formula Satu diselenggarakan di sirkuit atau jalanan umum pada kota-kota tertentu dan diatur oleh Federasi Otomotif Internasional (FIA). Jalanan umum yang digunakan akan ditutup sementara untuk umum demi melancarkan seri balapan. Seri balapan atau Grand Prix sebagian besar berada di Eropa sebagai pusat tradisi Formula Satu hingga sekarang. Namun, sekarang sudah semakin banyak negara di seluruh penjuru dunia yang membuat sirkuit untuk mengadakan Grand Prix dengan bertambahnya negara-negara Asia seperti Bahrain dan Malaysia.

Dari segi olahraganya dan produksi, ajang olahraga ini adalah balapan termahal. Sebuah tim mobil dapat mengeluarkan triliunan rupiah setiap tahunnya. Berdasarkan sebuah artikel pada detik, biaya operasional logistik dan pengangkutan dapat menghabiskan Rp 335,5 miliar.

Pada tahun 2020, dunia dilanda dengan musibah pandemi COVID-19. Hal ini mengakibatkan berhentinya berbagai kegiatan seperti Formula Satu dan berkurangnya pendapatan. Jadwal balapan yang biasa dimulai awal tahun berubah dan baru dimulai pada pertengahan tahun, sehingga jadwal antar Grand Prix harus lebih dipadatkan dan berubah menjadi setiap minggu melainkan pada tahun lain yang biasanya tiap dua minggu. Maka, FIA dan tim mobil F1 harus berpindah negara dan bersiap-siap untuk melaksanakan Grand Prix selanjutnya dalam jangka waktu yang sebentar.

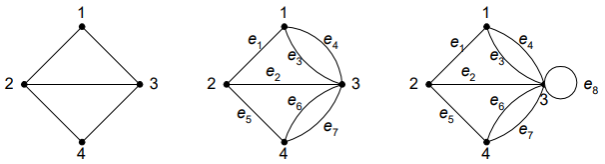
Jadwal Grand Prix musim 2020 yang digunakan dibagi berdasarkan region dari Eropa ke Asia. Namun, negara pada jadwal tidak terurut berdasarkan jarak antar negara, seperti saat ada Grand Prix di Italia, lalu pindah ke negara lain, dan kembali lagi ke Italia. Akan lebih efisien jika semua Grand Prix yang ada di Italia dapat dilakukan terlebih dahulu sebelum berpindah ke negara lain.

Selain untuk efisiensi waktu, perubahan jadwal juga dapat mengurangi biaya operasional seperti biaya perjalanan, logistik, dan pengangkutan, dengan asumsi biaya berkurang seiring dengan semakin dekat jarak perjalanan. Biaya yang digunakan juga akan berkurang jika perjalanan dapat dicapai hanya dengan transportasi darat dibandingkan transportasi udara.

## II. LANDASAN TEORI

### A. Graf

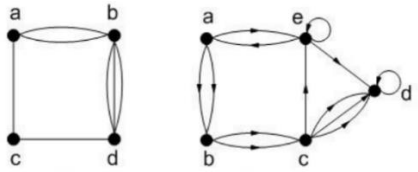
Graf adalah sekumpulan objek yang dinamakan simpul dan saling dihubungan dengan sebuah garis yang disebut sisi. Graf $G$ dapat didefinisikan $G = (V, E)$ dengan $V$ sebagai himpunan tidak kosong dari simpul-simpul dan $E$ sebagai himpunan sisi yang menghubungkan sepasang simpul.




Gambar 1 : Contoh graf
Sumber : [3]

Terdapat beberapa jenis graf yang dapat dibedakan berdasarkan beberapa kategori. Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf dapat digolongkan menjadi dua jenis, yaitu graf sederhana dan graf tak-sederhana. Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda seperti graf $G_1$ pada Gambar 1. Graf tak sederhana merupakan kebalikan dari graf sederhana dan dapat dibedakan lagi menjadi graf ganda yang mengandung sisi ganda seperti graf $G_2$ pada Gambar 1 dan graf gelang yang mengandung sisi gelang seperti graf

$G_3$ pada Gambar 1.

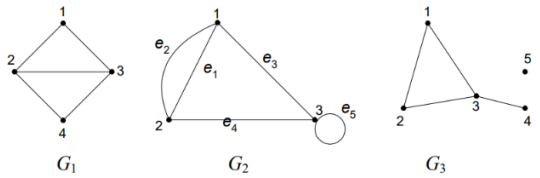



Gambar 2 : Jenis graf berdasarkan arah
Sumber : [3]

Berdasarkan orientasi arah pada sisi dapat dibedakan atas 2 jenis, yaitu graf tak-berarah seperti graf $G_2$ pada Gambar 2 dan graf berarah seperti graf $G_1$ pada Gambar 2.

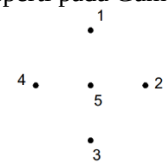
## *B. Terminologi Graf*

Graf memiliki berbagai macam terminologi sebagai berikut.



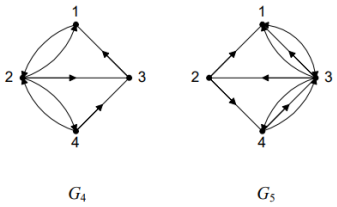

Gambar 3 : Contoh graf
Sumber : [3]

1. Ketetanggan (Adjacent)
   Bila dua buah simpul terhubung langsung dengan sisi, kedua buah simpul tersebut dapat dikatan bertetangga, Pada Gambar 3, graf $G_1$ memiliki simpul 1 yang bertetangga dengan simpul 2 dan 3.
2. Bersisian (Incidency)
   Pada sebuah graf dengan sembarang sisi $e = (v_j, v_k)$, dapat dikatakan sisi $e$ bersisian dengan simpul $v_j$ dan $v_k$. Pada Gambar 3, graf $G_2$ memiliki sisi $e_1$ yang bersisian dengan simpul 1 dan 2.
3. Simpul Terpencil (Isolated Vertex)
   Simpul yang tidak memiliki sisi yang bersisian dengannya disebut simpul terpencil. Pada Gambar 3, Graf $G_3$ memiliki simpul 5 sebagai simpul terpencil.
4. Graf Kosong (null graph atau empty graph)
   Graf kosong merupakan graf yang memiliki himpunan sisi kosong ($N_n$), seperti pada Gambar 4.



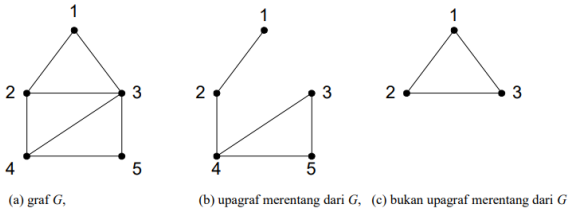

Gambar 4 : Contoh graf kosong
Sumber : [3]

5. Derajat (Degree)
   Derajat suatu simpul dengan notasi $d(v)$ merupakan jumlah sisi yang bersisian dengan simpul tersebut. Pada Gambar 3, graf $G_2$ memiliki $d(2) = 3$ dan $d(3) = 4$, sedangkan graf $G_5$ memiliki $d(5) = 0$.



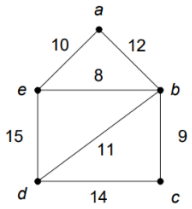

Gambar 5 : Contoh graf berarah
Sumber : [3]

   Sebuah simpul pada graf berarah memiliki 2 jenis derajat yaitu $d_{in}$ dan $d_{out}$. Pada Gambar 5, graf $G_5$ memiliki $d_{in}(1) = 2$ dan $d_{out}(1) = 1$.
6. Lintasan (Path)
   Sebuah graf $G$ dengan lintasan yang panjangnya $n$ dari simpul awal $v_0$ ke simpul tujuan $v_n$ adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$ sedemikian sehingga sisi-sisi dari graf $G$ adalah $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ... , $e_n = (v_{n-1}, v_n)$.
7. Siklus (Cycle) atau Sirkuit (Circuit)
   Siklus merupakan lintasan yang berawal dan jugaq berakhir pada simpul yang sama.
8. Kerterhubungan (Connected)
   Jika terdapat lintasan dari simpul $v_1$ ke simpul $v_2$, dua buah simpul tersebut dapat dikatakan terhubung. Graf $G$ dapat disebut graf terhubung jika setiap pasang simpul pada graf memiliki lintasan yang menghubungkan kedua simpul. Pada Gambar 3, graf $G_3$ merupakan graf tak-terhubung karena simpul 5 tidak terhubung dengan yang lainnya.
9. Upagraf (Subgraph) dan Komplemen Upagraf
   Upagraf dari graf $G_1 = (V_1, E_1)$ adalah graf $G_2 = (V_2, E_2)$ jika $V_2 \subseteq V_1$ dan $E_2 \subseteq E_1$, seperti pada Gambar 6(b).




Gambar 6 : Contoh upagraf
Sumber : [3]

10. Upagraf Merentang (Spanning Subgraph)
    Upagraf $G_2$ dari graf $G_1$ dikatakan upagraf rentang jika $G_2$ mengandung semua simpul dari $G_1$, seperti pada Gambar 6(b).
11. Cut-Set
    Himpunan sisi yang dibuang dari graf terhubung $G$ dan menyebabkan $G$ tidak terhubung merupakan *cut-set*. Pada Gambar 6(b), sisi (2,4) merupakan salah satu *cut-set* dari garf tersebut.
12. Graf Berbobot (Weighted Graph)
    Sebuah graf yang setiap sisinya diberi harga (bobot) seperti pada Gambar 7 merupakan graf berbobot.




Gambar 7 : Contoh graf berbobot
Sumber : [3]

### *C. Lintasan Hamilton*

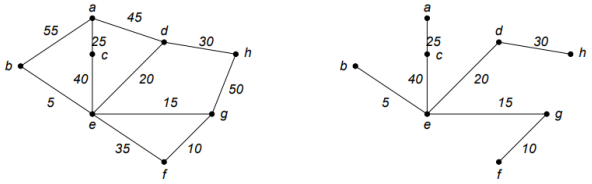
Lintasan Hamilton merupakan lintasan yang melalui setiap simpul di dalam graf *G* tepat satu kali. Sirkuit Hamilton serupa dengan lintasan Hamilton, tetapi simpul asal yang sekaligus simpul akhir dilalui dua kali. Graf Hamilton adalah sebuah graf yang memiliki sirkuit Hamilton, sedangkan graf smi-Hamilton adalah graf yang hanya memiliki lintasan Hamilton.

Gambar 8 : Contoh sirkuit Hamilton
Sumber : [4]

Pada Gambar 8, dapat dibentuk lintasan Hamilton bila salah satu sisi pada sirkuit dibuang.

### *D. Pohon minimum berentang*

Pohon adalah sebuah graf tak-berarah yang tidak memiliki sirkuit. Graf terhubung memiliki upagraf yang tidak ada sirkuit dan dapat disebut pohon merentang. Pohon minimum berentang merupakan pohon merentang dari graf terhubung-berbobot seperti yang ditunjuk pada Gambar 9.




Gambar 9 : Contoh pohon minimum berentang
Sumber : [5]

### *E. Algoritma Prim*

Algoritma Prim merupakan suatu algoritma untuk mencari pohon merentang minimum dari graf terhubung-berbobot dan memiliki langkah-langkah sebagai beriku.

Langkah 1 : ambil sisi dari graf *G* yang berbobot minimum dan masukkan ke dalam *T*.

Langkah 2 : pilih sisi (*u*, *v*) yang mempunyai bobot minimum dan bersisian dengan simpul di *T*, tetapi (*u*, *v*) tidak membentuk sirkuit di *T*. Masukkan (*u*, *v*) ke dalam *T*.

Langkah 3 : ulangi langkah 2 sebanyak *n* – 2 kali.

Algoritma Prim juga dapat ditulis dalam sepuah program. Berikut adalah program dari algoritma Prim dalam notasi algoritmik.

```
procedure Prim(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-
berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
  i, p, q, u, v : integer

Algoritma
  Cari sisi (p,q) dari E yang berbobot terkecil
  T ← {(p,q)}
  for i←1 to n-2 do
    Pilih sisi (u,v) dari E yang bobotnya terkecil namun
    bersisian dengan simpul di T
    T ← T ∪ {(u,v)}
  endfor
```

## III. APLIKASI ALGORITMA PRIM DAN LINTASAN HAMILTON DALAM PENENTUAN JADWAL BALAPAN F1 AGAR LEBIH EFISIEN

### *A. Jadwal Balapan F1*

Jadwal balapan F1 musim 2020 terdiri atas 17 Grand Prix yang berawal di Austria dan berakhir di Uni Emirat Arab.


Gambar 10 : Jadwal Balapan F1 Musim 2020
Sumber:https://sports.okezone.com/read/2020/08/25/37/2267366/resmi-miliki-17-seri-ini-jadwal-lengkap-f1-2020

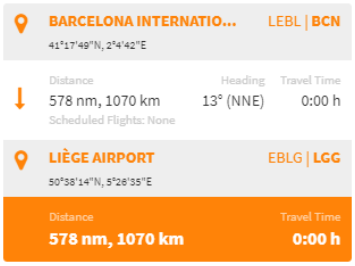
Pada jadwal terdapat beberapa minggu yang memiliki Grand Prix pada sirkuit yang sama. Penulis mempersingkat jadwal pada sirkuit yang sama dan digabung menjadi satu, sehingga jumlah simpul kota dapat berkurang menjadi 14. Berikut adalah tabel jadwal balapan F1 musim 2020 yang telah dipersingkat.

| Simpul | Kota | Bandara |
|---|---|---|
| A | Spielberg, Austria | GRZ |
| B | Budapest, Hongaria | BUD |
| C | Silverstone, Inggris | LTN |
| D | Montmelo, Spanyol | BCN |
| E | Stavelot, Belgia | LGG |
| F | Milan, Italia | MXP |
| G | Scarperia e San Piero, Italia | BLQ |
| H | Sochi, Rusia | AER |
| I | Nürburg, Jerman | FRA |
| J | Portimão, Portugal | FAO |
| K | Bologna, Italia | BLQ |
| L | Istanbul, Turki | IST |
| M | Sakhir, Bahrain | BAH |

| | | |
|---|---|---|
| N | Abu Dhabi, Uni Emirat Arab | AUH |

Tabel 1 : Representasi Kota pada simpul graf

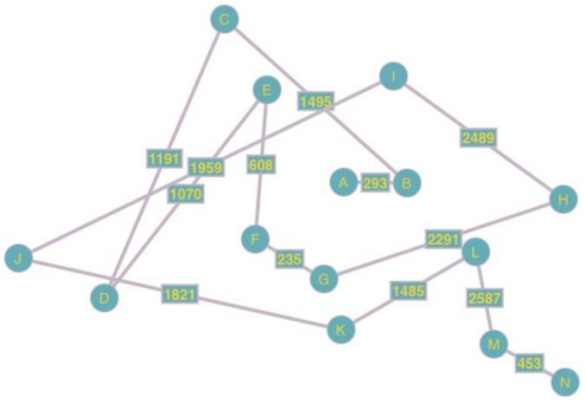
Jarak antar kota dicari dengan menghitung jarak antar bandara setiap kota dengan menggunakan *Great Circle Mapper*. Kota dengan bandara yang sama dicari jarak antar kota berdasarkan *Google Maps*.




Gambar 11 : Jarak antar bandara

Sumber : https://www.greatcirclemapper.net/

Setelah menghitung jarak antar kota, Penulis membuat sebuah lintasan Hamilton dari jadwal tersebut dengan letak simpul yang dibuat serupa mungkin dengan lokasi asli pada peta.

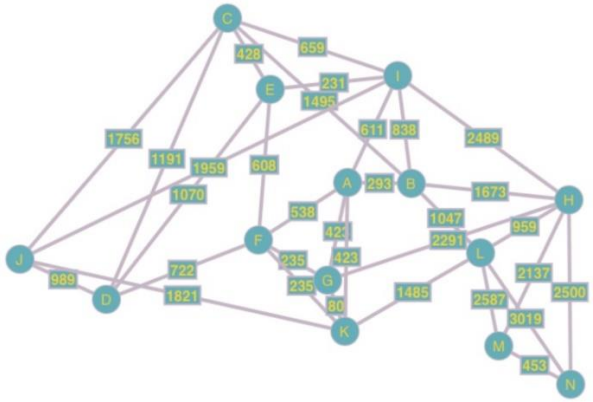



Gambar 12 : Visualisasi rute jadwal dalam graf

Graf yang dihasilkan pada Gambar 12 memiliki bobot total 17977 km.

## *B. Aplikasi Algoritma Prim dan Lintasan Hamilton*

Untuk mencari jadwal yang lebih efisien digunakan algoritma Prim untuk mencari pohon merentang minimum yang juga merupakan lintasan Hamilton. Penulis membuat jalur antar negara yang dekat satu sama lain. Berikut adalah graf dari jalur antar negara tersebut.




Gambar 13: Graf *Rute*

Kejuaraan musim ini akan diakhiri di Abu Dhabi (N). Oleh karena itu, pencarian pohon merentang minimum dapat dimulai dari simpul N hingga mendapatkan simpul terakhir yang menjadi lokasi Grand Prix pertama pada jadwal balapan musim ini. Terdapat 14 simpul pada graf, sehingga akan dihasilkan pohon merentang minimum dalam 13 langkah sebagai berikut.

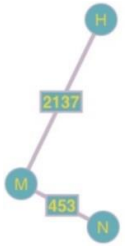
Langkah 1 - 13:

Dengan algoritma Prim, langkah 1 hingga langkah 10 dapat diulangi metodenya dengan membuat pohon *T* yang mengambil sisi dari graf *Rute* yang berbobot minimum dan bersisian dengan simpul pada pohon *T*.

Pada langkah 1 dicari sisi berbobot minimum yang bersisian dengan N pada graf *Rute* karena simpul tersebut merupakan akhir dari lintasan Hamilton yang ingin dicari. Diperoleh sisi (M, N) dengan bobot 453 sebagai sisi berbobot minimum.

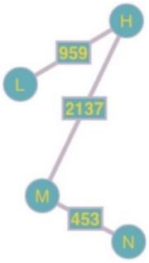

Gambar 14 : Pohon merentang langkah 1

Pada langkah 2, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T*. Diperoleh sisi (H, M) dengan bobot 2137 sebagai sisi berbobot minimum.



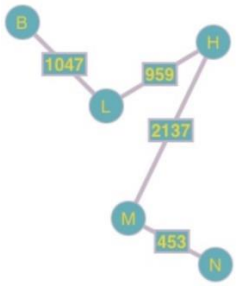

Gambar 15 : Pohon merentang langkah 2

Pada langkah 3, dilakukan cara yang serupa dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan kali ini tidak membentuk sirkuit pada pohon *T* karena jumlah simpul sudah lebih dari 2, sehingga sudah mulai ada sisi yang dapat membentuk sirkuit. Diperoleh sisi (L, H) dengan bobot 959 sebagai sisi berbobot minimum.



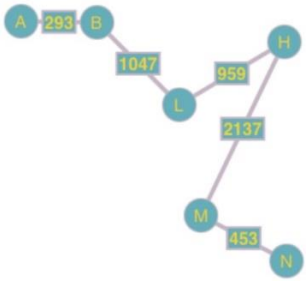

Gambar 16 : Pohon merentang langkah 3

Pada langkah 4, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Diperoleh sisi (B, L) dengan bobot 1047 sebagai sisi berbobot minimum.



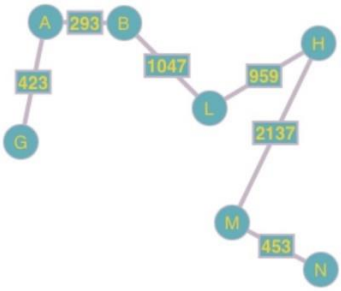

Gambar 17 : Pohon merentang langkah 4

Pada langkah 5, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Diperoleh sisi (A, B) dengan bobot 293 sebagai sisi berbobot minimum.



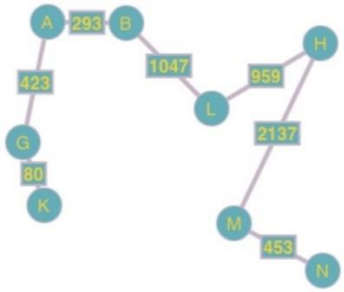

Gambar 18 : Pohon merentang langkah 5

Pada langkah 6, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Ditemukan bahwa ada dua sisi dengan bobot yang sama dan keduanya merupakan bobot minimum yaitu sisi (G, A) dan sisi (K, A). Dalam hal ini, Penulis memilih sisi (G, A) dengan bobot 423 sebagai sisi berbobot minimum karena kota di simpul G muncul terlebih dahulu dibandingkan kota di simpul K pada jadwal balapan asli.



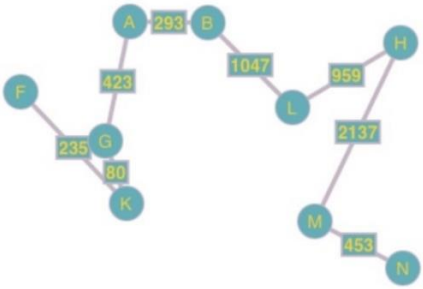

Gambar 19 : Pohon merentang langkah 6

Pada langkah 7, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Diperoleh sisi (K,G) dengan bobot 80 sebagai sisi berbobot minimum. Jika tidak ditentukan di awal untuk memulai pohon *T* dari simpul N, maka sisi ini akan menjadi sisi pertama karena memiliki bobot paling minimum dibandingkan sisi yang lain pada graf *Rute*.




Gambar 20 : Pohon merentang langkah 7

Pada langkah 8, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Diperoleh sisi (F, K) dengan bobot 235 sebagai sisi berbobot minimum.




Gambar 21 : Pohon merentang langkah 8

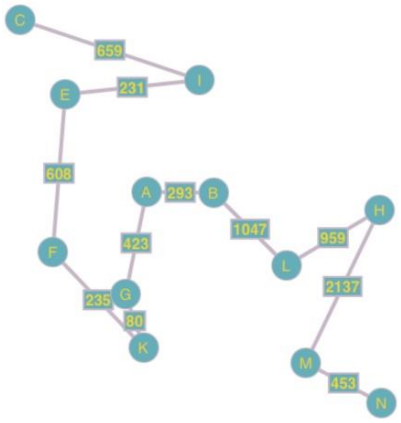
Pada langkah 9, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Ditemukan bahwa sisi berbobot minimum terdapat pada sisi (A, F) dengan bobot 538. Namun, sisi tersebut akan membentuk sebuah sirkuit jika diambil, sehingga diambil sisi (E, F) dengan bobot 608 sebagai sisi berbobot minimum karena tidak membentuk sirkuit.


Gambar 22 : Pohon merentang langkah 9

Pada langkah 10, dilakukan cara yang sama dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Diperoleh sisi (I, E) dengan bobot 231 sebagai sisi berbobot minimum.

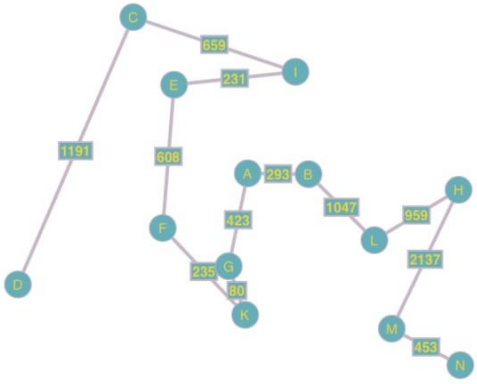

Gambar 23 : Pohon merentang langkah 10

Pada langkah 11, dilakukan modifikasi dari algoritma Prim. Algoritma Prim biasa akan mengambil sisi (C, E) karena memiliki bobot minimum yaitu 428, tetapi hal tersebut tidak akan menghasilkan sebuah lintasan Hamilton. Jadi, agar dapat menghasilkan sebuah lintasan Hamilton, diambil sisi berbobot minimum yang bersisian dengan simpul terakhir yang dimasukkan ke dalam pohon *T*, yaitu sisi (C, I) dengan bobot 659.



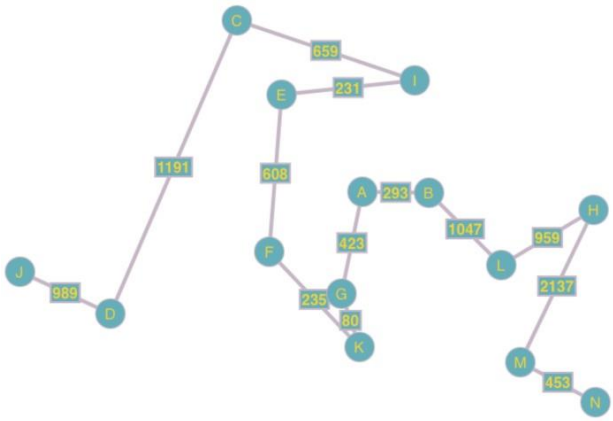

Gambar 24 : Pohon merentang langkah 11

Pada langkah 12 dapat dicari sisi tanpa memodifikasi algoritma Prim karena sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit, tetap dapat menghasilkan sebuah lintasan Hamilton. Diperoleh sisi (D, C) dengan bobot 1191 sebagai sisi berbobot minimum.




Gambar 25 : Pohon merentang langkah 12

Pada langkah 13, dilakukan cara yang sama seperti langkah 12 tanpa perlu memodifikasi algoritma Prim, yaitu dengan mencari sisi berbobot minimum dari graf *Rute* yang bersisian dengan simpul pada pohon *T* dan tidak membentuk sirkuit. Diperoleh sisi (J, D) dengan bobot 989 sebagai sisi berbobot minimum.




Gambar 26 : Pohon merentang langkah 13

Setelah melakukan 13 langkah pencarian lintasan Hamilton dengan memodifikasi algoritma Prim, hasil dari algoritma tersebut dapat dilihat pada Gambar 26 dengan bobot total 9305 km. Graf tersebut merepresentasikan jalur dari jadwal balapan F1 musim 2020 yang lebih efisien dan dapat ditampilkan jadwal tersebut dalam bentuk tabel sebagai berikut.

| Ronde | Kota |
|---|---|
| 1 | Portimão, Portugal |
| 2 | Montmelo, Spanyol |
| 3 | Silverstone, Inggris |
| 4 | Nürburg, Jerman |
| 5 | Stavelot, Belgia |
| 6 | Milan, Italia |
| 7 | Bologna, Italia |
| 8 | Scarperia e San Piero, Italia |
| 9 | Spielberg, Austria |
| 10 | Budapest, Hongaria |
| 11 | Istanbul, Turki |
| 12 | Sochi, Rusia |
| 13 | Sakhir, Bahrain |
| 14 | Abu Dhabi, Uni Emirat Arab |

Tabel 2 : Jadwal balapan yang lebih efisien

Tabel jadwal balapan ini akan mengurangi waktu dan biaya operasional pada musim 2020 karena memiliki bobot minimum yaitu 9305 km dibandingkan jadwal asli yang berbobot 17977 km. Hal ini juga menunjukkan bahwa penerapan algoritma Prim dan lintasan Hamilton telah menghasilkan sebuah jadwal yang memiliki bobot jarak hingga setengah dari awalnya.

## IV. KESIMPULAN

Lintasan Hamilton dengan bobot minimum dapat dicari dengan memodifikasi algoritma Prim agar dapat mencari sisi berbobot minimum setelah simpul yang terakhir dimasukkan ke dalam pohon merentang.

Penerapan algoritma ini telah menghasilkan jadwal balapan Formula Satu dengan rute berbobot minimum yang lebih efisien baik dari segi waktu dan biaya operasional, sehingga dapat meringankan beban biaya terutama pada masa pandemi ini.

## V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih dan puji syukur kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya, Penulis dapat menyelesaikan tugas makalah berjudul "Aplikasi Algoritma Prim dan Lintasan Hamilton dalam Penentuan Jadwal Balapan F1 Agar Lebih Efisien" ini dengan baik. Penulis juga ingin mengucapkan terima kasih kepada kedua orang tua, kakak, dan adik atas semangat yang selalu diberikan dan bantuannya. Tak lupa diucapkan juga terima kasih kepada Ibu Fariska Zakhralativa Ruskanda, ST., MT. selaku dosen IF2120 Matematika Diskrit Kelas 03 atas bimbingan dan ajarannya yang baik di kelas.

## REFERENSI

[1] https://id.wikipedia.org/wiki/Formula_Satu, diakses pada 8 Desember 2020, pukul 21.20.
[2] https://sport.detik.com/f1/d-3161957/menghitung-mahalnya-biaya-balapan-f1, diakses pada 8 Desember 2020, pukul 21.43.
[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf, diakses pada 9 Desember 2020, pukul 16.24
[4] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian3.pdf, diakses pada 9 Desember 2020, pukul 16.48.
[5] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf, diakses pada 9 November 2020, pukul 17.05.

## 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.

Jakarta, 9 Desember 2020

Fauzan Yubairi Indrayadi 13519171