

# Analisis Pola Jungling ONIC Kairi pada Babak Final MPL ID S12 Menggunakan Algoritma Prim dan Teorema Kombinasi

Ahmad Rafi Maliki - 13522137<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

<sup>1</sup>13522137@std.stei.itb.ac.id

**Abstrak**—*Mobile Legends: Bang Bang* adalah salah satu *game* ponsel yang sangat populer di kalangan pemuda Indonesia. Kairi merupakan seorang pemain profesional yang berasal dari tim ONIC Esports. Sebagai pemegang *role jungler* pada timnya, Kairi dinilai sebagai *jungler* terbaik di kalangan pemain profesional. Hal ini disebabkan karena *path jungling* yang efisien sehingga menghasilkan posisi lebih unggul bagi diri dan timnya ketimbang tim lawan. Makalah ini bertujuan untuk menilai efisiensi *path jungling* Kairi ketimbang lawannya sehingga para pemain umum dapat mengaplikasikan hal serupa atau lebih baik ke dalam permainannya. Analisis dilakukan dengan pendekatan Algoritma Prim yaitu algoritma yang dapat menemukan *minimum spanning tree* pada sebuah graf berbobot.

**Kata kunci** — *Mobile Legends*, *minimum spanning tree*, graf berbobot, Algoritma Prim, kombinasi.

## I. PENDAHULUAN

*Mobile Legends: Bang Bang*, sebuah permainan ponsel yang sangat populer di kalangan pemuda Indonesia, memiliki begitu banyak pemain dengan antusiasme luar biasa. Dalam ekosistem ini, Kairi, seorang pemain profesional yang berasal dari tim ONIC Esports, telah menonjol sebagai *jungler* terbaik, menunjukkan keunggulan dalam *path jungling* yang efisien. *Jungler* memiliki peran yang krusial dalam permainan, dan efektivitas Kairi dalam mengelola *path*-nya membawa dampak positif bagi dirinya dan timnya.

Pada level profesional, Kairi berhasil menduduki posisi terdepan berkat keunggulan *path*-nya yang dapat menciptakan posisi lebih unggul dibandingkan dengan tim lawan. Keahliannya dalam mengoptimalkan setiap langkahnya menjadikannya pemain yang patut dicontoh oleh para pemain *Mobile Legends*, terutama yang tengah menjalani perjalanan dalam memahami permainan ini.

Makalah ini bertujuan untuk mendalami dan mengevaluasi efisiensi *path* Kairi, dengan harapan bahwa pemain umum dapat memahami dan mengaplikasikan prinsip-prinsip tersebut ke dalam permainan mereka. Analisis akan dilakukan dengan menggunakan pendekatan Algoritma Prim, sebuah metode yang digunakan untuk menemukan *minimum spanning tree* pada sebuah graf berbobot. Analisis yang dilakukan adalah pada *final match* *Mobile Legends: Bang Bang Professional League Season 12* dari detik nol sampai *turtle* pertama.

Melalui pendekatan ini, kita akan mencoba memahami struktur dan efisiensi *path* Kairi secara lebih mendalam, sehingga informasi yang dihasilkan dapat menjadi pedoman berharga bagi pemain *Mobile Legends* yang sedang berusaha meningkatkan keterampilan mereka.

Setelah melakukan analisis nantinya pula diharapkan kita dapat mengetahui apakah Algoritma Prim merupakan algoritma yang cocok untuk memecahkan masalah ini atau tidak.

## II. STUDI LITERATUR

### A. Teori Graf

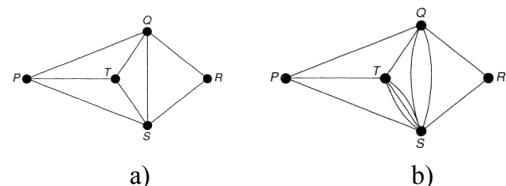
#### Definisi Graf

Graf yang merupakan representasi diagramatik dari hubungan antara titik-titik (simpul) yang dihubungkan dengan garis-garis (sisi). Simpul dan sisi membentuk struktur yang disebut graf.

Secara matematis, suatu graf dapat dinotasikan dengan  $G = (V, E)$ , yang dalam hal ini  $V$  adalah himpunan tidak kosong dari simpul-simpul (*vertices*)  $\{v_1, v_2, \dots, v_n\}$ , dan  $E$  adalah himpunan sisi (*edges*) yang menghubungkan sepasang simpul  $\{e_1, e_2, \dots, e_n\}$ .

#### Graf Sederhana dan Tidak Sederhana

Salah satu cara untuk menggolongkan sebuah graf adalah berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, berdasarkan cara ini graf dibagi menjadi graf sederhana dan graf tak sederhana. Graf sederhana adalah graf yang tidak mengandung gelang ataupun sisi ganda sedangkan, graf tak sederhana adalah graf yang mengandung gelang atau sisi ganda.

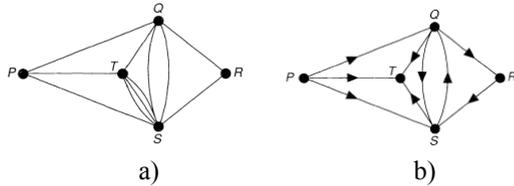


**Gambar 1.** Ilustrasi Graf Sederhana (a) dan Graf Tidak Sederhana (b)

Sumber: Wilson, R. J (1996), Introduction to Graph Theory Fourth Edition, Longman.

### Graf Berarah dan Tidak Berarah

Adapun penggolongan graf berdasarkan orientasi arah pada sisi graf yaitu graf tak berarah dan graf berarah. Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah sedangkan, graf berarah adalah graf yang tiap sisinya diberikan orientasi arah.



**Gambar 2.** Ilustrasi Graf Tak Berarah (a) dan Graf Berarah (b)

Sumber: Wilson, R. J (1996), Introduction to Graph Theory Fourth Edition, Longman.

### Graf Berbobot

Graf dapat memiliki nilai berupa angka pada sisi-sisinya yang disebut sebagai graf berbobot (*weighted graph*).

### Sirkuit pada Graf

Sebuah graf dikatakan memiliki sirkuit apa bila terdapat rangkaian sisi yang dapat dilewati dari sebuah simpul dan berakhir di simpul mula. Apa bila terdapat rangkaian sisi yang dapat dilewati namun tidak kembali ke simpul mula, maka hal tersebut disebut sebagai lintasan. Sebuah graf yang memiliki sirkuit pasti memiliki lintasan namun hal sebaliknya tidak berlaku.

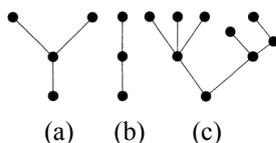
### Upagraf

Upagraf atau *subgraph* adalah sebuah graf yang diperoleh dari graf lain yang himpunan sisi dan simpulnya merupakan himpunan bagian dari graf lain tersebut. Sebuah upagraf dapat lebih kecil atau sama besarnya dengan graf utamanya.

## B. Teori Pohon

### Definisi Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Pohon yang diperoleh dari sebuah graf disebut dengan pohon merentang (*spanning tree*).



**Gambar 4.** Ilustrasi Pohon

Sumber: Wilson, R. J (1996), Introduction to Graph Theory Fourth Edition, Longman.

### Pohon Berakar

Jenis pohon yang umum digunakan dalam dunia komputasi adalah pohon berakar (*rooted tree*), yaitu pohon yang sisinya memiliki orientasi arah masuk dan keluar. Jumlah maksimal sisi masuk tiap simpul pada rooted tree adalah satu, sedangkan jumlah sisi keluarnya beragam. Pada sebuah pohon berakar tidak kosong pasti ada satu dan hanya satu simpul yang tidak memiliki sisi masuk, simpul tersebut disebut sebagai akar. Sedangkan simpul yang tidak memiliki sisi keluar disebut sebagai daun.

## C. Algoritma Prim

### Definisi Algoritma Prim

Sama halnya dengan graf, sebuah pohon dapat memiliki bobot pada sisinya. Algoritma Prim sebuah algoritma pada teori graf yang berguna untuk mencari pohon merentang minimum (pohon yang memiliki total bobot terkecil) dari sebuah graf.

Metode ini mengidentifikasi subset sisi yang mencakup setiap simpul di dalam graf dengan tujuan meminimalkan total bobot dari sisi-sisi tersebut.

### Cara Kerja Algoritma Prim

Langkah awal dari Algoritma Prim adalah dengan memilih simpul tunggal dan mengeksplorasi semua simpul yang bersebelahan dengan sisi penghubung pada setiap iterasi. Pada setiap langkah, sisi-sisi yang memiliki bobot terendah, sehingga tidak membentuk siklus dalam graf, akan diprioritaskan.

### Aplikasi Algoritma Prim

Aplikasi yang luas dari algoritma Prim mencakup perancangan jaringan yang efisien, pembentukan siklus dalam struktur jaringan, serta penentuan optimal letak kabel listrik untuk memastikan keandalan distribusi energi.

## D. Teorema Kombinasi

Kombinasi atau Kombinatorial adalah cabang matematika untuk menghitung jumlah penyusunan objek-objek tanpa harus mengenumerasi semua kemungkinan susunannya

### Kaidah Dasar Menghitung

Terdapat beberapa kaidah dasar menghitung untuk memahami konsep kombinatorial, yaitu kaidah perkalian dan kaidah penjumlahan. Kedua kaidah tersebut nantinya akan mengalami perluasan menjadi konsep kombinatorial yang dikenal secara umum.

### Rumus Menghitung Kombinasi

Sebagai perluasan dari kaidah penjumlahan dan perkalian, telah dirumuskan suatu formula untuk mencari kombinasi dari sebuah himpunan. Caranya adalah sebagai berikut.

$$C(n, r) = \frac{n!}{r!(n-r)!}$$

Kombinasi  $r$  dari  $n$  elemen dinotasikan sebagai  $C(n, r)$ . Notasi tersebut artinya adalah banyaknya cara memiliki  $r$  buah elemen dari himpunan  $n$  buah elemen. Pengkalkulasian banyaknya kombinasi memanfaatkan konsep matematika yaitu faktorial.

### E. Terminologi Mobile Legends

Istilah penting dalam permainan Mobile Legends yang perlu diketahui untuk memahami makalah ini adalah sebagai berikut.

*Map*, adalah arena permainan dimana kedua tim melakukan pertarungan. Pada *map* terdapat area khusus yang disebut *jungle* dimana terdapat banyak monster yang dapat dibunuh oleh para pemain untuk meningkatkan kekuatan mereka, adapun pemain yang bertugas untuk menghabisi monster di *jungle* disebut sebagai *jungler*, sedangkan *jungling* adalah istilah untuk kegiatan seorang *jungler* membunuh monster *jungle*.

Pada makalah ini akan dibahas pola perpindahan *jungler* dari satu monster ke monster lainnya mulai dari awal permainan sampai menit ke-dua yaitu saat monster *turtle* pertama muncul, *turtle* adalah sebuah monster yang lebih kuat yang jika dibunuh dapat memberikan keuntungan lebih pada tim yang membunuhnya.

Analisis pada makalah ini hanya akan dilakukan sampai *turtle* pertama muncul karena menit-menit awal permainan memiliki pengaruh besar kepada kemenangan, dan juga setelah membunuh *turtle*, pola *path jungling* seorang *jungler* akan menyesuaikan dengan kondisi permainan dan bukan lagi hal yang cocok untuk dianalisis dengan algoritma Prim.

## III. METODE

### A. Persiapan Perencana Rute (Mission Planner)



**Gambar 5.** Map Mobile Legends

Sumber : Mobile Legends v1.8.33.905.4

Pada gambar 5 didefinisikan 9 buah simpul yang mewakili lokasi monster pada *map* Mobile Legends, dalam pembentukan *minimum spanning tree* nantinya, akar pohon harus berada pada simpul 1 (posisi *start*) dan simpul 9 (monster *turtle*) merupakan daun pohon. Terdapat pula simplifikasi yaitu tidak setiap monster pada *jungle* dijadikan simpul karena ada

beberapa monster yang pasti sudah diambil oleh *jungler* musuh dan tidak mungkin kita lalui lagi.

Jika pada *minimum spanning tree* gagal membentuk pohon dengan ketentuan berikut, maka diperlukan evaluasi lebih lanjut tentang algoritma apa yang lebih cocok untuk menyelesaikan permasalahan ini yang tidak akan dibahas pada makalah ini.

### B. Pembentukan Graf Berbobot (Weighted Graph)



**Gambar 5.** Graf Map Mobile Legends

Sumber : Mobile Legends v1.8.33.905.4

Pada permainan Mobile Legends, seorang *jungler* membutuhkan waktu untuk berjalan dan waktu untuk membunuh monster *jungle*. Maka dari itu, bobot pada graf akan mewakili total waktu berjalan dari node awal ke node tujuan dan waktu membunuh monster pada node tujuan dalam satuan detik.

Pada gambar dapat diamati bahwa sebuah simpul tidak terhubung ke semua simpul, walaupun hal tersebut adalah hal yang mungkin saja terjadi namun seorang *jungler* umumnya hanya akan melalui jalur seperti pada graf yang digambarkan pada gambar 5.

Data bobot pada graf akan diperoleh dengan cara melakukan eksperimen

**Tabel 1.** Bobot graf pada gambar 5

Sisi	Bobot	Sisi	Bobot
1—2	42	5—7	15
1—5	42	5—8	12
2—3	17	6—7	16
2—4	10	6—8	12
2—5	24	7—8	19
2—6	26	7—9	21
3—4	16	7—12	17
3—9	20	8—9	32
3—10	10	8—12	12
4—5	25	9—10	20

4-7	10	9-11	20
4-9	22	9-12	29
5-6	12	10-11	13

### C. Data Path Jungling Onic Kairi

Data *path jungling* Onic Kairi merupakan data yang diperoleh dari ronde 1-6 pada babak final MPL ID S12, data kemudian disajikan pada tabel.

*Path* yang terdapat pada tabel dapat berupa hasil isomorf *path* sebenarnya yang dilalui Kairi pada permainan karena *map* Mobile Legends terdapat 2 jenis yaitu seperti yang disajikan pada gambar 4 ataupun hasil dari pencerminan terhadap diagonalnya (diagonal bawah kiri-atas kanan).

**Tabel 2.** *Path Jungling* Onic Kairi

Babak	<i>Path Jungling</i> (Simpul)	Total Bobot
1	1-5-6-7-8-12-9	130
2	1-5-6-2-3-10-9	127
3	1-5-6-7-4-2-3-9	127
4	1-5-2-3-4-7-9	130
5	1-2-4-5-6-8-9	133
6	1-2-4-7-5-8-9	121

Bisa diperhatikan pada Tabel 2 bahwa tidak semua simpul yang terdapat pada gambar 5 digunakan, disinilah peran teorema kombinasi untuk menghasilkan subgraf yang hanya memuat node sesuai seperti yang tertera pada Tabel 2.

### D. Perhitungan Kombinasi Sub-Graf

Syarat kombinasi dari suatu graf (subgraf) yang valid pada konteks ini adalah sub-graf yang memiliki simpul 1 dan simpul 9. Maka menurut persamaan pada teorema kombinasi, dapat dihasilkan graf sebanyak pada perhitungan berikut.

$$\text{Banyak Sub-Graf} = \sum_{i=0}^{10} C(10, i)$$

Namun, diantara banyaknya sub-graf yang mungkin terbentuk berdasarkan perhitungan di atas. Sub-graf yang akan digunakan hanyalah yang cocok dengan *path* yang dilalui oleh Onic Kairi. Sub-graf yang digunakan disajikan pada tabel berikut.

**Tabel 3.** *Sub-Graf yang digunakan pada Analisis*

Babak	Banyak Simpul	Himpunan Simpul
1	7	{1, 5, 6, 7, 8, 9, 12}

2	7	{1, 2, 3, 5, 6, 9, 10}
3	8	{1, 2, 3, 4, 5, 6, 7, 9}
4	7	{1, 2, 3, 4, 5, 7, 9}
5	7	{1, 2, 4, 5, 6, 8, 9}
6	7	{1, 2, 4, 5, 7, 8, 9}

### D. Penentuan Minimum Spanning Tree dari Sub-Graf

Pada bagian ini akan ditentukan *minimum spanning tree* dari tiap sub-graf yang tertera pada Tabel 3 menggunakan Algoritma Prim berupa urutan simpul yang dilalui mulai dari akar sampai daun, jika terdapat langkah yang tidak mungkin maka akan melakukan *backtracking*. Data *minimum spanning tree* disajikan pada tabel berikut.

**Tabel 4.** Data *Minimum Spanning Tree (MST)*

Babak	MST	Total Bobot
1	1-5-6-8-12-7-9	116
2	1-5-6-2-3-10-9	127
3	1-5-6-7-4-2-3-9	127
4	1-5-2-4-7-3-9	122
5	1-2-4-5-6-8-9	133
6	1-2-4-7-5-8-9	121

## IV. HASIL DAN PEMBAHASAN

### A. Hasil

Setelah diperoleh data pada Tabel 2 dan Tabel 4, untuk melakukan analisis lebih lanjut akan dihitung persentase kemiripan bobot *minimum spanning tree* dengan *jungling path* Onic Kairi pada tiap babak. Persentase diperoleh dengan cara membagi bobot MST dengan bobot *jungling path* lalu mengalikannya dengan 100%. Setelah dilakukan perhitungan, diperoleh data pada tabel berikut.

**Tabel 5.** Data Kemiripan *MST* dengan *Jungling Path*

Babak	Bobot MST	Bobot <i>Jungling Path</i>	Persentase Kemiripan
1	116	130	89.2%
2	127	127	100%
3	127	127	100%

4	122	130	93.8%
5	133	133	100%
6	121	121	100%

## B. Pembahasan

Melalui analisis yang mendalam terhadap data yang terdapat dalam Tabel 5, dapat ditarik kesimpulan bahwa pencapaian yang dicapai oleh Onic Kairi dalam menjalani jalur jungling sangat luar biasa. Dari enam babak yang diperhatikan, empat di antaranya menunjukkan bahwa kemiripan *minimum spanning tree* dengan jalur *jungling*-nya mencapai puncak 100%. Pencapaian ini tidak hanya mencerminkan kehebatan Kairi dalam mengambil keputusan strategis, tetapi juga menggambarkan keefisienan optimal jalur *jungling* yang ditempuhnya, terutama ketika diukur menggunakan algoritma Prim. Babak 2, 3, 5, dan 6 menjadi bukti konkret bahwa Kairi mampu memilih jalur yang tidak hanya efisien tetapi juga optimal untuk membunuh monster *jungle*, meraih keuntungan maksimal untuk timnya.

Meskipun demikian, pada babak 1, walaupun Kairi berhasil mencapai kemiripan jalur sebesar 89.2%, dan pada babak 4 sebesar 93.8%, terdapat peluang untuk meningkatkan efisiensi lebih lanjut. Meskipun pemilihan jalur Kairi sudah mencapai tingkat yang baik dalam persentase kemiripan, masih ada ruang untuk penyesuaian guna menyelaraskan jalur dengan jalur yang ditentukan oleh algoritma Prim.

Secara menyeluruh, analisis menyimpulkan bahwa algoritma Prim berhasil menggambarkan dengan jelas bahwa Kairi adalah salah satu pemain jungler terbaik. Keunggulan ini tidak hanya terletak pada kemampuannya dalam memilih jalur yang efisien sesuai dengan algoritma Prim, tetapi juga pada kebijaksanaannya dalam mengikuti prinsip-prinsip matematis yang terbukti efektif dalam meraih kemenangan. Dengan kata lain, Kairi tidak hanya mengandalkan insting dan pengalaman semata, tetapi juga menerapkan pendekatan yang terukur dan strategis untuk memastikan kesuksesan timnya. Dengan demikian, implementasi algoritma Prim dalam menganalisis permainan Kairi memberikan pandangan yang sangat mendalam tentang keunggulan strategisnya sebagai seorang *jungler* yang ulung.

Terlebih lagi, analisis yang mendalam terhadap kinerja Kairi membuktikan bahwa algoritma Prim bukan hanya sekadar alat analisis, tetapi juga menjadi penanda keunggulan strategisnya dalam tekanan permainan yang intens. Pada babak-babak krusial seperti 2, 3, 5, dan 6, Kairi mampu mempertahankan tingkat konsistensi yang luar biasa dalam pemilihan jalur *jungling*-nya, memberikan dampak positif yang signifikan pada progres timnya.

Penting untuk dicatat bahwa, meskipun terdapat sedikit ruang untuk peningkatan efisiensi pada beberapa babak, hal ini tidak mengurangi nilai dari keberhasilan Kairi dalam menerapkan algoritma Prim sebagai panduan strategisnya. Dalam dunia permainan profesional, di mana setiap keputusan dan langkah

memiliki dampak besar, kualitas inisiatif yang diambil Kairi dalam mengikuti jalur yang dihasilkan oleh algoritma Prim memberikan bukti konkret bahwa keputusan berdasarkan analisis matematis dapat menjadi kunci untuk meraih kemenangan.

Selain itu, penggunaan algoritma Prim dalam analisis permainan Kairi tidak hanya menunjukkan keahliannya dalam memilih jalur *jungling*, tetapi juga membuka pintu bagi pengembangan strategi lebih lanjut. Pemain-pemain yang terinspirasi oleh pendekatan matematis Kairi dapat menggali lebih dalam ke dalam prinsip-prinsip algoritma dan mengadopsinya ke dalam gaya bermain mereka sendiri. Dengan demikian, penerapan algoritma Prim dalam ranah permainan Mobile Legends tidak hanya memberikan keunggulan kompetitif bagi individu seperti Kairi, tetapi juga dapat memengaruhi evolusi strategi dalam komunitas pemain secara lebih luas.

## V. KESIMPULAN

Penggunaan Algoritma Prim dalam menentukan jalur *jungling* dalam permainan Mobile Legends telah terbukti sangat tepat. Jalur *jungling* yang dihasilkan oleh algoritma ini menonjol dengan tingkat kemiripan yang sangat tinggi dengan jalur *jungling* yang biasa ditempuh oleh seorang pemain profesional yang berhasil meraih kemenangan dalam MPL ID S12, sebuah turnamen bergengsi di tingkat nasional. Algoritma Prim tidak hanya memberikan pandangan yang mendalam dan analitis terhadap keputusan pemain profesional tersebut, tetapi juga menunjukkan keefisienan dan keunggulan strategis yang dapat diadopsi oleh pemain lainnya.

Ketepatan algoritma ini tergambar jelas melalui kesesuaian jalur *jungling* dengan gaya bermain pemain berkelas seperti yang terlihat pada turnamen MPL ID S12. Adanya kesesuaian tersebut memperkuat keyakinan bahwa Algoritma Prim dapat menjadi pedoman yang berharga bagi pemain Mobile Legends, terutama para jungler, untuk meningkatkan efisiensi dan konsistensi mereka dalam menjelajahi hutan dan membunuh monster *jungle*. Dengan merinci jalur yang optimal, algoritma ini membantu para pemain mengambil keputusan yang cerdas, berdasarkan analisis matematis yang terbukti efektif dalam menghadapi berbagai situasi permainan.

Penting untuk dicatat bahwa tingkat kemiripan yang tinggi antara jalur *jungling* algoritma Prim dengan jalur yang ditempuh oleh pemain profesional mencerminkan bukan hanya kualitas algoritma, tetapi juga kemampuannya untuk menangkap esensi strategi dan taktik yang diterapkan oleh pemain berpengalaman. Oleh karena itu, implementasi Algoritma Prim tidak hanya memberikan pandangan objektif terhadap jalur *jungling*, tetapi juga menjadi alat yang berharga dalam memahami dan mengadopsi strategi yang telah teruji oleh pemain-pemain terbaik di level nasional.

## VI. SARAN

Saran untuk peneliti selanjutnya adalah dapat membandingkan pola *jungling* suatu tim dengan tim lawan baik saat kondisi menang atau kalah, selain itu faktor seperti diserang oleh tim lawan saat melakukan *jungling* juga dapat diperhitungkan sebagai faktor yang mempengaruhi bobot graf.

## VII. UCAPAN TERIMA KASIH

Terima kasih kepada Tuhan Yang Maha Esa karena dengan rahmat dan hidayahnya makalah berjudul "Analisis Pola Jungling Onic Kairi pada Babak Final MPL ID S12 Menggunakan Algoritma Prim dan Teorema Kombinasi" dapat terselesaikan tepat waktu. Selanjutnya saya juga berterima kasih kepada Bpk Dr. Ir. Rinaldi Munir, S.T, M.T dan Bpk Monterico Adrian, S.T, M.T yang telah membimbing mata kuliah IF2120 Matematika Diskrit selama satu semester ini.

## DAFTAR PUSTAKA

- [1] Wilson, R. J (1996), Introduction to Graph Theory Fourth Edition, Longman.
- [2] <https://www.javatpoint.com/prim-algorithm> (diakses pada 09-12-2023).
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf> (diakses pada 09-12-2023).
- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/22-Pohon-Bag1-2023.pdf> (diakses pada 09-12-2023).
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/17-Kombinatorial-Bagian1-2023.pdf> (diakses pada 09-12-2023).

## 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, 11 Desember 2023



Ahmad Rafi Maliki - 13522137