

# Implementasi Algoritma Greedy dalam Menentukan Rute Kabel Internet di Perumahan

Muhammad Zaki - 13522136

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): 13522136@std.stei.itb.ac.id

**Abstract**— Dalam era digital, kebutuhan akan konektivitas internet yang andal dan efisien menjadi sangat krusial, terutama di kawasan perumahan. Namun, seringkali pemilihan rute untuk menyambungkan kabel dari rumah ke rumah menimbulkan masalah berupa penumpukan kabel yang berlebihan dan tidak enak dipandang. Dalam makalah ini, penulis mencoba menentukan rute yang optimal dengan menggunakan algoritma Greedy untuk mengurangi penumpukan kabel tersebut. Algoritma Greedy dipilih karena kemampuannya dalam menghasilkan solusi yang efisien melalui pemilihan langkah terbaik secara iteratif. Hasil penelitian menunjukkan bahwa penerapan algoritma Greedy mampu menghasilkan rute kabel yang lebih teratur dan efisien, mengurangi jumlah kabel yang terlihat dan meningkatkan estetika lingkungan perumahan. Kesimpulannya, penggunaan algoritma Greedy dalam perencanaan rute kabel internet di perumahan tidak hanya mengurangi penumpukan kabel tetapi juga meningkatkan kualitas dan efisiensi instalasi.

**Keywords**— *Internet, Kabel, Algoritma Greedy, Perencanaan Rute, Perumahan*

## I. PENDAHULUAN

### A. Latar Belakang

Dalam era digital saat ini, konektivitas internet telah menjadi kebutuhan dasar yang mendukung berbagai aktivitas sehari-hari, mulai dari komunikasi, pendidikan, hingga pekerjaan. Di kawasan perumahan, permintaan akan layanan internet yang cepat dan stabil semakin meningkat seiring dengan perkembangan teknologi dan kebutuhan masyarakat. Penyedia layanan internet menghadapi tantangan dalam memasang jaringan kabel yang tidak hanya efisien tetapi juga estetis dan tidak mengganggu pemandangan lingkungan.

Salah satu masalah utama yang sering dihadapi adalah penumpukan kabel yang berlebihan ketika mencoba menyambungkan setiap rumah ke jaringan utama. Penumpukan ini tidak hanya mengurangi keindahan lingkungan, tetapi juga dapat menimbulkan masalah teknis seperti gangguan sinyal dan kesulitan dalam perawatan jaringan. Oleh karena itu, diperlukan sebuah metode yang dapat menentukan rute pemasangan kabel yang optimal untuk mengurangi

penumpukan kabel sekaligus menjaga efisiensi biaya dan waktu pemasangan.

Penulis merasa bahwa kondisi penarikan kabel internet atau listrik di Indonesia, khususnya di tempat tinggal penulis, sudah sangat meresahkan. Penarikan kabel yang dilakukan secara sembarangan tanpa mempertimbangkan tata kelola yang baik atau perencanaan jangka panjang menjadi sumber utama kekhawatiran ini. Kabel-kabel sering kali ditarik secara acak dan tidak teratur, mengakibatkan tampilan yang semrawut dan potensi bahaya di kemudian hari.



Gambar 1 Ilustrasi Kabel Di Perumahan  
Sumber: Dokumentasi Pribadi

Selain aspek keamanan dan perawatan, estetika lingkungan juga terganggu. Kabel-kabel yang menjuntai sembarangan membuat pemandangan di sekitar menjadi tidak sedap dipandang. Lingkungan yang tertata rapi dengan penarikan kabel yang teratur tidak hanya meningkatkan nilai estetika, tetapi juga mencerminkan tata kelola yang baik dan kepedulian terhadap masyarakat.

Oleh karena itu, penulis berharap adanya perhatian lebih dari pihak terkait, baik pemerintah maupun perusahaan penyedia jasa internet dan listrik, untuk memperbaiki tata

kelola penarikan kabel. Dengan perencanaan yang baik dan penarikan kabel yang teratur, tidak hanya keamanan dan efisiensi yang dapat ditingkatkan, tetapi juga kenyamanan dan keindahan lingkungan tempat tinggal akan terjaga. Penulis percaya bahwa dengan langkah-langkah yang tepat, masalah ini dapat diatasi demi kebaikan bersama.

Algoritma Greedy, yang dikenal karena kesederhanaan dan efisiensinya dalam menyelesaikan masalah optimasi, menawarkan solusi potensial untuk masalah ini. Dengan prinsip memilih solusi terbaik secara iteratif, algoritma ini dapat membantu menentukan jalur pemasangan kabel yang paling efisien dari segi biaya dan penampilan.

### B. Tujuan

Penelitian ini bertujuan untuk menerapkan algoritma Greedy dalam menentukan rute pemasangan kabel internet di kawasan perumahan. Diharapkan, implementasi algoritma ini dapat menghasilkan desain jaringan yang lebih rapi, efisien, dan estetis, sehingga meningkatkan kualitas layanan internet yang disediakan kepada penghuni perumahan.

### C. Manfaat

Penelitian ini diharapkan memiliki berbagai manfaat signifikan dalam konteks perencanaan dan pemasangan jaringan kabel internet di kawasan perumahan. Pertama, implementasi algoritma Greedy dapat mengurangi biaya pemasangan kabel dengan menentukan rute yang optimal dan ekonomis, memungkinkan penyedia layanan internet untuk menghemat sumber daya dan mengalokasikannya untuk peningkatan layanan lainnya. Selain itu, algoritma ini membantu mengurangi penumpukan kabel yang berlebihan, menciptakan lingkungan perumahan yang lebih rapi dan estetis, serta mengurangi potensi gangguan dan kerusakan pada kabel.

## II. STUDI LITERATUR

### A. Internet

Internet merupakan singkatan dari Interconnection Networking. Kata "Internet" berasal dari bahasa lain, dengan "Inter" yang berarti antara. Secara harfiah, Internet dapat diartikan sebagai jaringan penghubung. Definisi ini merujuk pada koneksi antara berbagai jenis komputer dan jaringan di seluruh dunia yang memiliki sistem operasi dan aplikasi yang berbeda, di mana koneksi tersebut memanfaatkan teknologi komunikasi seperti telepon dan satelit, serta menggunakan protokol standar untuk komunikasi, yaitu TCP/IP (Transmission Control Protocol/Internet Protocol). Definisi ini dijelaskan oleh Supriyanto, Dodit (2008:60).

Secara sederhana, menurut Jogiyanto Hartono (2009:1), Internet adalah kumpulan dari sejumlah komputer, bahkan

hingga jutaan komputer di seluruh dunia yang saling terhubung. Media yang digunakan untuk menghubungkan komputer-komputer ini bisa berupa kabel atau serat optik, satelit, atau sambungan telepon. Menurut pendapat ini, Internet adalah media komunikasi dan informasi modern yang dapat diakses secara global oleh pengguna di seluruh dunia melalui interkoneksi antar jaringan komputer yang difasilitasi oleh penyedia layanan Internet (provider). Dengan demikian, Internet sebagai media informasi menjadi sarana yang efektif dan efisien untuk pertukaran dan penyebaran informasi tanpa dibatasi oleh jarak, perbedaan waktu, atau faktor geografis bagi siapa saja yang ingin mengakses informasi.

### B. Algoritma Dijkstra



Gambar 2 Edsger W. Dijkstra  
Sumber: Wikipedia.com

Algoritma Dijkstra, ditemukan oleh Edsger Dijkstra pada tahun 1959, merupakan algoritma yang digunakan untuk menyelesaikan permasalahan jarak terpendek pada sebuah graf berarah. Algoritma ini dianggap sebagai algoritma greedy karena pada setiap langkahnya, ia memilih simpul dengan jarak terpendek dari simpul awal tanpa mempertimbangkan jarak keseluruhan ke tujuan.

Permasalahan rute terpendek dari satu titik ke titik lainnya adalah masalah klasik dalam optimisasi dan sering digunakan untuk menguji efektivitas algoritma baru. Meskipun permasalahannya sederhana untuk dimengerti - hanya dengan menjumlahkan seluruh panjang edge yang dilalui - namun memiliki banyak kemungkinan solusi.

Alasan mengapa para peneliti terus mempelajari masalah pencarian jalur terpendek sangat beragam. Masalah ini relevan dalam berbagai aplikasi seperti routing jaringan, game, desain sirkuit, dan pemetaan.

Secara matematis, grafik dapat diwakili sebagai  $G = \{V, E\}$ , yang berarti grafik ( $G$ ) terdiri dari sebuah set simpul (Vertex =  $V$ ) dan kumpulan edge ( $E$ ).

### C. Algoritma Greedy

Algoritma greedy adalah jenis algoritma yang membuat pilihan optimal secara lokal pada setiap langkah

dengan harapan menemukan solusi global yang optimal. Dalam kata lain, algoritma ini membuat pilihan terbaik pada setiap langkah tanpa mempertimbangkan konsekuensi jangka panjang. Algoritma greedy bekerja dengan memilih opsi terbaik secara berulang pada setiap langkah tanpa melihat ke depan. Pendekatan ini sering digunakan ketika masalah dapat dipecah menjadi sub-masalah yang lebih kecil, dan solusi optimal untuk sub-masalah tersebut juga optimal untuk keseluruhan masalah.

Algoritma greedy memiliki beberapa kelebihan. Pertama, algoritma ini mudah diimplementasikan karena hanya perlu mempertimbangkan opsi terbaik pada setiap langkah. Kedua, algoritma greedy bisa sangat cepat karena tidak perlu mempertimbangkan semua kemungkinan solusi. Ketiga, algoritma ini sederhana untuk dianalisis karena hanya perlu mempertimbangkan opsi terbaik pada setiap langkah. Namun, algoritma greedy juga memiliki beberapa kelemahan.

Pertama, algoritma ini mungkin tidak selalu menemukan solusi optimal karena hanya mempertimbangkan opsi terbaik pada setiap langkah. Kedua, algoritma greedy mungkin tidak bekerja untuk semua masalah karena membutuhkan masalah untuk dipecah menjadi sub-masalah yang lebih kecil.

Sebagai gambaran bagaimana algoritma greedy ini bekerja, diberikan contoh sederhana sebagai berikut. Bayangkan kita memiliki sebuah peta kecil dengan beberapa tiang listrik atau internet yang tersebar di sekitar lingkungan kita. Setiap tiang dihubungkan dengan kabel yang memiliki panjang tertentu, yang direpresentasikan sebagai bobot (weight) pada graf. Misalnya, kita ingin menemukan jalur terpendek dari rumah kita (node awal) ke sebuah tiang tertentu (node tujuan).

Algoritma Dijkstra, yang merupakan contoh dari algoritma greedy, memulai pencarian dari node awal dan selalu memilih node tetangga dengan jarak terpendek yang belum dikunjungi. Ini dilakukan dengan menggunakan struktur data heap minimum untuk memastikan bahwa setiap langkah selalu mempertimbangkan node dengan jarak terpendek saat ini. Algoritma ini terus memperbarui jarak terpendek dari node awal ke setiap node lain yang bisa dicapai melalui node yang telah dikunjungi, sambil melacak jalur yang diambil.

Sebagai ilustrasi, jika kita memulai dari node rumah kita dan bergerak ke node tiang terdekat, algoritma ini akan terus mencari jalur dengan jarak terpendek ke node tujuan dengan cara iteratif. Pada akhirnya, kita akan mendapatkan jalur terpendek beserta total jaraknya. Dengan menggunakan pendekatan greedy ini, Algoritma Dijkstra memastikan bahwa solusi yang ditemukan adalah jalur terpendek antara dua titik dalam graf, yang dalam contoh ini merepresentasikan penarikan kabel terpendek dan paling efisien antara dua tiang listrik atau internet di sekitar lingkungan kita.

### III. IMPLEMENTASI

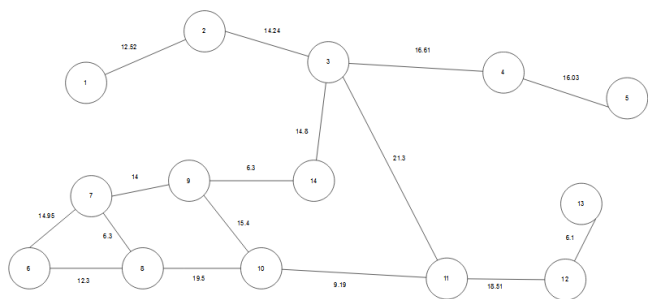
Implementasi yang saya gunakan untuk makalah ini adalah denah lokasi dari tiang tiang internet atau listrik di sekitar lingkungan tempat tinggal saya. Pertama tama saya memetakan tiang – tiang internet ataupun listrik, lalu setelah saya memetakan itu saya pun mengubah nya menjadi sebuah node node yang merepresentasikan keterhubungan antara satu tiang dengan tiang yang lain.

Untuk perhitungan jarak antara tiang, saya menggunakan bantuan dari Google Maps. Aplikasi ini sangat berguna karena memungkinkan pengukuran jarak yang akurat antara dua titik dengan cepat dan mudah. Dengan menggunakan fitur pengukuran jarak di Google Maps, saya dapat memastikan bahwa perhitungan yang saya lakukan tepat dan dapat diandalkan. Selanjutnya, saya memasukkan data jarak tersebut ke node node sebagai acuan jarak.



Gambar 3 Ilustrasi Pengukuran Menggunakan Google Maps

Algoritma Greedy yang saya gunakan adalah Algoritma Dijkstra. Algoritma ini memilih tiang internet dengan jarak terdekat dari posisi awal pada setiap langkahnya, tanpa mempertimbangkan jarak keseluruhan ke tujuan (misalnya, ke rumah Anda). Proses ini berlanjut hingga semua tiang internet dieksplorasi. Algoritma Dijkstra memastikan bahwa setiap tiang internet dijelajahi, dan jarak terpendek dari posisi awal ke setiap tiang internet ditemukan. Meskipun algoritma ini dapat menemukan tiang internet terdekat dari posisi awal, ia cenderung menjadi kurang efisien ketika berhadapan dengan banyaknya tiang internet dalam area yang luas karena harus memeriksa setiap kemungkinan jalur.



Gambar 4 Ilustrasi Graf yang Digunakan

Gambar di atas memperlihatkan representasi visual dari sejumlah node yang merepresentasikan tiang listrik atau internet yang tersebar di sekitar rumah penulis. Setiap node dihubungkan dengan garis-garis yang mewakili jarak antara satu node dengan node lainnya. Meskipun representasi ini tidak sepenuhnya akurat atau presisi secara skala, namun memberikan gambaran tentang keterhubungan antara node-node tersebut atau keterhubungan antara tiang satu dengan yang lain.

Dalam konteks ini, setiap node mungkin mewakili sebuah tiang listrik atau internet, sementara garis-garis antara node merepresentasikan jarak antara tiang-tiang tersebut dalam satuan meter. Meskipun demikian, perlu diingat bahwa skala pada gambar tersebut mungkin tidak mencerminkan skala sebenarnya di lapangan.

Pemahaman visual dari keterhubungan ini sangat berguna dalam merencanakan atau menganalisis jaringan listrik atau internet di sekitar rumah penulis. Dengan melihat gambar ini, dapat diperoleh pemahaman tentang bagaimana tiang-tiang tersebut terhubung satu sama lain dan bagaimana rute terpendek dari satu tiang ke tiang lainnya dapat ditentukan. Hal ini dapat membantu dalam perencanaan perawatan atau pengembangan jaringan, serta dalam mengevaluasi efisiensi jaringan yang ada.

```

g.add_edge(1, 2, 12.52)
g.add_edge(2, 3, 14.24)
g.add_edge(3, 4, 16.61)
g.add_edge(3, 11, 21.3)
g.add_edge(4, 5, 16.03)
g.add_edge(3, 11, 21.3)
g.add_edge(11, 12, 18.51)
g.add_edge(12, 13, 6.1)
g.add_edge(11, 10, 9.19)
g.add_edge(10, 9, 15.4)
g.add_edge(9, 7, 14)
g.add_edge(7, 6, 14.95)
g.add_edge(6, 8, 12.3)
g.add_edge(8, 10, 19.5)
g.add_edge(14, 3, 14.8)
g.add_edge(14, 9, 6.3)

```

Gambar 5 Translasi Graf Ke Kode

Kode ini merupakan implementasi dari sebuah graf yang menggunakan Algoritma Dijkstra untuk mencari jalur terpendek antara dua titik (node) dalam graf tersebut. Pada bagian awal, kelas `Graph` didefinisikan dengan metode untuk menambahkan edge dan mengimplementasikan Algoritma Dijkstra. Metode `add\_edge` digunakan untuk menambahkan edge antara dua node dengan bobot tertentu. Algoritma Dijkstra diimplementasikan dalam metode `dijkstra`, yang menerima node awal dan node tujuan sebagai parameter. Algoritma ini menggunakan heap minimum (min-heap) untuk secara efisien menemukan jalur terpendek dari node awal ke node tujuan, memperbarui jarak minimum ke setiap node, dan melacak node sebelumnya untuk membangun jalur terpendek.

Pada bagian contoh penggunaan, sebuah graf dengan 12 node dibuat, dan sejumlah edge ditambahkan antara node-node tersebut dengan bobot tertentu yang merepresentasikan jarak dalam meter. Setelah mendefinisikan graf dan menambahkan edge, metode `dijkstra` dipanggil dengan node awal dan akhir tertentu untuk menemukan dan mencetak jalur terpendek beserta total jarak dari node awal ke node tujuan. Contoh ini menggambarkan bagaimana Algoritma Dijkstra dapat digunakan untuk mencari jalur terpendek dalam jaringan yang kompleks, seperti jaringan tiang internet atau listrik di sekitar lingkungan penulis.

#### IV. ANALISIS

Dari data yang sudah kita dapatkan, kita bisa memeriksa rute mana yang seharusnya dipakai oleh sebuah kabel internet. Hasil perhitungan ini kemudian dapat dibandingkan dengan kondisi di dunia nyata untuk melihat apakah rute kabel yang ada saat ini sudah optimal atau belum.

Dengan menggunakan Algoritma Dijkstra, kita dapat menemukan jalur terpendek dan paling efisien antara dua titik, misalnya antara dua tiang internet atau listrik.

Dengan membandingkan jalur optimal yang dihitung oleh algoritma dengan jalur yang sebenarnya digunakan di lapangan, kita dapat mengidentifikasi apakah ada penarikan kabel yang tidak efisien atau terlalu panjang. Jika ternyata jalur yang ada di lapangan lebih panjang atau lebih kompleks dibandingkan dengan jalur optimal yang dihitung, ini bisa menjadi indikasi bahwa penarikan kabel yang ada perlu dioptimalkan.

Selain itu, perbandingan ini juga dapat membantu dalam perencanaan jaringan yang lebih baik di masa depan. Misalnya, jika kita menemukan bahwa jalur optimal melintasi area yang lebih aman atau lebih mudah diakses dibandingkan dengan jalur yang ada saat ini, kita bisa merekomendasikan penarikan ulang kabel untuk meningkatkan efisiensi dan mengurangi risiko gangguan atau kecelakaan.

Perbandingan yang kita ambil pertama adalah penarikan kabel dari ke tiang 8, berikut adalah data jarak tiap tiang ke tiang 8 menggunakan algoritma djikstra

Tabel 1: Jarak Tiap Tiang ke Tiang 8

Tiang	Jarak (M)
1	68.16
2	5.64
3	41.40
4	58.01
5	74.04
6	12.3
7	6.3
8	0
9	20.3
10	19.5
11	26.69
12	47.2
13	53.3
14	26.6

Dari data diatas kita bisa bandingkan apakah rute kabel di lingkungan saya sudah optimal atau belum menerut algoritma djikstra. Untuk perbandingan awal kita lihat dari tiang 1.

Pada Algoritma Dijkstra yang digunakan rute yang optimal adalah:

1 -> 2 -> 3 - > 14 -> 9 -> 7 -> 8

Dengan total cost yaitu:

$$12.52 + 14.24 + 14.8 + 6.3 + 14 + 6.3 = 68.16$$

```
Shortest path from 1 to 8: [1, 2, 3, 14, 9, 7, 8]
Total distance: 68.16
Edges and their costs:
1 -> 2: 12.52
2 -> 3: 14.24
3 -> 14: 14.8
14 -> 9: 6.3
9 -> 7: 14
9 -> 7: 14
7 -> 8: 6.3
```

Gambar 6 Hasil Algoritma

Sedangkan pada kenyataan yang penulis telusuri rute yang digunakan untuk menyambung kabel 1 ke 8 adalah rute : 1 -> 2 -> 3 -> 11 -> 10 -> 8 , dengan total cost yang digunakan adalah  $12,52 + 14,24 -> 21,3 -> 9.19 + 19.5 = 76.75$ . Dari kasus diatas kita dapat mengetahui bahwa rute kabel yang ada pada lingkungan penulis belum optimal yang dapat menyebabkan kabel yang berantakan dan mengganggu estetika lingkungan.



Gambar 7 Rute kabel dalam kenyataan

Untuk selanjutnya kita akan data dari tiap tiang ke tiang 13 datanya sebagai berikut :

Tabel 2: Jarak tiap tiang ke tiang 13

Tiang	Jarak (M)
1	72.67
2	60.15
3	45.91
4	62.52
5	78.55
6	65.6
7	59.6
8	53.3
9	49.9
10	33.8
11	24.61
12	6.1
13	0
14	55.5

Sebagai perbandingan kita akan melihat apakah rute dari tiang 14 ke 13 sudah optimal menurut algoritma djikstra atau greedy ini. Pada Algoritma Dijkstra yang digunakan rute yang optimal adalah:

14- > 9 -> 10 -> 11 -> 12 -> 13

Dengan total cost yaitu =  $6.3 + 15.4 + 9.19 + 18.51 + 6.1 = 55.5$

```
Shortest path from 14 to 13: [14, 9, 10, 11, 12, 13]
Total distance: 55.50000000000001
Edges and their costs:
14 -> 9: 6.3
9 -> 10: 15.4
10 -> 11: 9.19
11 -> 12: 18.51
12 -> 13: 6.1
```

Gambar 8 Hasil Algoritma

Sedangkan pada kenyataan yang penulis temukan, rute yang digunakan untuk menyambung kabel dari 14 ke 13 adalah 14 -> 3 -> 11 -> 12 -> 13, dengan total cost yang dibutuhkan adalah  $14.8 + 21.3 + 18.51 + 6.1 = 60.71$ . Dari kasus kedua tadi kita dapat melihat bahwa rute kabel yang dilalui pada kasus dunia nyata tidak optimal atau memiliki selisih 5.21 meter.

Dari kedua kasus yang sudah kita teliti tadi, ternyata keduanya masih tidak optimal dalam penentuan rute. Kasus pertama memiliki selisih 8.34 meter, sementara kasus kedua memiliki selisih 5.21 meter. Hal ini menunjukkan bahwa rute yang dipilih dalam kenyataan tidak selalu mengikuti hasil optimal yang ditentukan oleh algoritma, sehingga menyebabkan inefisiensi dalam penggunaan kabel.

### V. KESIMPULAN

Dari penelitian ini mengkaji penggunaan Algoritma Dijkstra untuk menentukan rute optimal pemasangan kabel internet di kawasan perumahan. Dengan membandingkan hasil perhitungan algoritma dengan kondisi nyata di lapangan, ditemukan bahwa rute yang ada belum optimal. Contoh kasus menunjukkan bahwa rute yang digunakan di lapangan memiliki panjang yang lebih besar dibandingkan rute optimal yang dihitung oleh algoritma, dengan selisih 8.34 meter pada kasus pertama dan 5.21 meter pada kasus kedua. Ketidakakuratan ini menunjukkan perlunya perbaikan dalam

perencanaan dan penarikan kabel untuk meningkatkan efisiensi dan estetika lingkungan.

Temuan ini menekankan pentingnya penggunaan algoritma optimasi dalam perencanaan infrastruktur untuk mengurangi inefisiensi dan meningkatkan kualitas layanan serta estetika lingkungan perumahan.

### REFERENCES

- [1] Alabi, T.H. (2024) What is a greedy algorithm? examples of greedy algorithms, freeCodeCamp.org. Available at: <https://www.freecodecamp.org/news/greedy-algorithms/> (Accessed: 09 June 2024).
- [2] GeeksforGeeks (2024) *What is Dijkstra's algorithm?: Introduction to dijkstra's shortest path algorithm*, GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/introduction-to-dijkstras-shortest-path-algorithm/> (Accessed: 09 June 2024).
- [3] *Algoritma greedy*. Available at: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf) (Accessed: 10 June 2024).

### VIDEO YOUTUBE :

<https://youtu.be/ccDqd3gtxzE>

### 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, 12 Juni 2024



Ttd  
Muhammad Zaki 13522136