

Penerapan Algoritma Branch and Bound dan Greedy untuk Mengelilingi Tempat Wisata Terkenal di Bandung pada Liburan Akhir Semester

Khalilullah Al Faath – 1301204376

Naufal Abdurrahman Burhani – 1301204008

Ariqo Sukma Bahamis Ali – 1301200361

Program Studi S-1 Informatika, Fakultas Informatika, Telkom University

Jl. Telekomunikasi Terusan Buah Batu Indonesia 40257, Bandung, Indonesia

E-mail (SSO): khalilullahalfaaath@student.telkomuniversity.ac.id; zeknous@student.telkomuniversity.ac.id; arigosukma@student.telkomuniversity.ac.id

Abstrak—Masalah TSP merupakan salah satu dari masalah NP-hard yang dalam penyelesaiannya dapat dicari dengan beberapa cara. Dua di antaranya adalah Branch and Bound dan Greedy. Di sini penulis mengaplikasikan masalah TSP menjadi pencarian rute terbaik untuk perjalanan wisata menjelang liburan. Didapatkan bahwa pendekatan greedy lebih mangkus daripada dengan algoritma BnB, tetapi hasil yang didapatkan bukan merupakan solusi yang optimal.

Kata kunci — Algoritma Branch and Bound, Algoritma Greedy, TSP, tempat wisata

I. PENDAHULUAN

Waktu liburan tidak lama lagi akan tiba dan salah satu cara untuk mengisi liburan adalah dengan mengunjungi tempat-tempat wisata. Bandung dari dulu terkenal dengan berbagai macam tempat wisatanya yang menarik. Oleh karena itu, dalam makalah ini dipilihlah tempat-tempat wisata di kota Bandung untuk menjadi basis permasalahan yang akan diselesaikan dengan algoritma *Branch and Bound* dan *Greedy* untuk mencari rute perjalanan terpendek untuk setiap tempat wisata yang dipilih yang dimulai dari kampus Telkom University dan berakhir juga di kampus.

Algoritma *Branch and Bound* sering dipilih untuk menyelesaikan persoalan *Travelling Salesman Problem*. Sebagaimana masalah optimasi yang lain.

Penulis merasa bahwa perancangan rute wisata ini menarik untuk dibahas. Selain karena waktu yang relatif tepat, tetapi juga diharapkan dengan solusi yang didapatkan pada tulisan ini dapat menghemat sumber daya, seperti bensin dan waktu perjalanan.

II. TINJAUAN PUSTAKA

A. Algoritma Branch and Bound

Algoritma *Branch and Bound* atau biasa disingkat sebagai BnB adalah strategi yang biasanya digunakan untuk menyelesaikan masalah optimasi dengan menggunakan pohon ruang status untuk pencarian solusi dan juga mempertimbangkan batas tertentu dalam konstruksi pohon ruang statusnya.

Pendekatan yang paling umum digunakan pada BnB adalah *Best-First Search*, yang merupakan pengembangan dari *Breadth First Search (BFS)*.

Suatu simpul pada pohon ruang status dianggap menjanjikan jika dan hanya jika nilai batas dari simpul tersebut lebih baik dari solusi terbaik sementara dengan mempertimbangkan semua simpul pada seluruh tingkatan (Wulandari & Sa'adah, 2021).

B. Algoritma Greedy

Algoritma *greedy* adalah metode yang paling populer dan sederhana untuk memecahkan persoalan optimasi yang mana pada algoritma *greedy*, persoalan dipecahkan secara langkah per langkah sedemikian sehingga pada setiap langkah:

1. mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “take what you can get now!”)
2. dan “berharap” bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global (Munir, 2022).

Secara umum terdapat enam komponen yang ada pada algoritma *greedy*, yaitu:

1. Himpunan kandidat

Merupakan himpunan yang berisi objek-objek yang dapat dipilih pada langkah-langkah algoritma *greedy*.

2. Himpunan solusi

Merupakan himpunan yang berisi atau merepresentasikan objek-objek yang telah terpilih pada langkah-langkah algoritma *greedy*.

3. Fungsi solusi

Merupakan fungsi yang mengindikasikan bahwa kita telah menemukan solusi yang lengkap. Fungsi inilah yang menjadi penentu akhir dari suatu iterasi.

4. Fungsi seleksi

Menyatakan aturan dasar pemilihan suatu kandidat untuk dijadikan sebagai bagian dari solusi.

5. Fungsi kelayakan

Fungsi yang menetapkan apakah suatu kandidat dapat masuk ke dalam solusi berdasarkan kendala yang ada.

6. Fungsi objektif

Fungsi yang meminimalkan atau memaksimalkan suatu nilai, yang menjadi tujuan utama dari masalah terkait (Wulandari & Sa'adah, 2021).

C. Travelling Salesman Problem

Travelling Salesman Problem atau biasa disingkat sebagai TSP adalah suatu permasalahan NP-hard dalam optimasi kombinatorik. Prinsip dari TSP ini adalah pertanyaan berikut:

“Diberikan larik yang terdiri atas kota-kota dan jarak antara pasangan kota-kota yang ada. Tentukan rute terpendek yang mana setiap kota dikunjungi tepat sekali dan kembali ke kota awal?”

Karena TSP adalah masalah minimasi, maka dalam penyelesaian dengan BnB perlu didefinisikan batas bawah untuk permasalahan ini. Jika $w(a,b)$ menyatakan bobot sisi dari simpul a ke simpul b , maka total bobot yang didapatkan oleh solusi dari permasalahan ini adalah penjumlahan dari n buah sisi, yaitu:

$$w(v_1, v_2) + w(v_2, v_3) + \dots + w(v_{n-1}, v_n) + w(v_n, v_1)$$

atau dapat ditulis sebagai

$$1/2([w(v_1, v_n) + w(v_1, v_2)] + \dots + [w(v_{n-1}, v_n) + w(v_n, v_1)])$$

Yang dapat dipandang sebagai hasil bagi dua penjumlahan seluruh bobot sisi yang bersisian dengan masing-masing simpul $1, 2, \dots, n$. Misalkan V_G adalah himpunan yang berisi simpul-simpul pada graf G , maka total bobot tersebut tidak akan kurang dari $\lceil \frac{1}{2} (s_1 + s_2 + \dots + s_n) \rceil$, di mana s_i adalah penjumlahan bobot dua sisi terkecil yang bersisian dengan simpul u_i . Jika G merupakan graf berarah, maka s_i merupakan bobot *incoming edge* terkecil yang bersisian dengan simpul u_i (Wulandari & Sa'adah, 2021).

Selain itu, lower bound dapat juga didefinisikan sebagai panjang dari rute yang bisa didapatkan minimal

apabila suatu simpul tersebut dibangkitkan anak-anaknya. Sedangkan pada akar, lower bound adalah *outgoing edge* terkecil untuk setiap simpul.

Sedangkan pada pendekatan *greedy*, semua sisi yang terhubung dengan simpul akan diurutkan dan dipilih sisi terpendek jika dan hanya jika tidak menyalahi aturan berikut.

1. Tidak ada sirkuit yang terbentuk sampai tempat yang terakhir dikunjungi
2. Tidak ada tempat yang dikunjungi dua kali selain tempat awal (Daniells, 2020)

D. Tempat wisata

Objek wisata adalah suatu tempat yang menjadi kunjungan pengunjung karena mempunyai sumber daya, baik alami maupun buatan manusia, seperti keindahan alam atau pegunungan, pantai flora dan fauna, kebun binatang, dan lain-lain.

Pada makalah ini dipilih 9 tempat wisata terbaik menurut Tripcetera blog, antara lain:

1. Trans Studio Bandung

Merupakan sebuah mal sekaligus taman bermain indoor terbesar pertama di Indonesia yang terletak di Kota Bandung, Jawa Barat.



2. Dago Dream Park

wisata alam pegunungan dengan nuansa pohon pinus yang indah, asri dan sejuk. Dago Dreampark dengan luas lahan ± 13 Hektar merupakan tempat wisata alam yang menghadirkan suasana alam pegunungan dengan berbagai wahana-wahana spot foto yang Instagramable antara lain, Karpas Alladin, Sky Tree, Sky Bike, Up House, Love Seat, Flying Lounge, ada juga wisata lain seperti Flying Fox, Kora-Kora, ATV dll.



3. Floating Market Lembang

Kawasan wisata terpadu yang berlokasi di jalan Grand Hotel No 33E, Kecamatan Lembang, Kabupaten Bandung, Jawa Barat Indonesia. Tempat wisata menarik ini menjadi salah satu destinasi wisata terpopuler dan Favorit wisatawan yang sedang liburan ke kota kembang, karena menawarkan banyak sekali daya tarik wisata, fasilitas dan wahana permainan, spot berfoto dan pastinya harga atau biaya tiketnya sangat terjangkau.



4. Gedung Sate

Berupa ornamen tusuk sate pada menara sentralnya, telah lama menjadi penanda atau markah tanah Kota Bandung yang tidak saja dikenal masyarakat di Jawa Barat, tetapi juga seluruh Indonesia bahkan model bangunan itu dijadikan pertanda bagi beberapa bangunan dan tanda-tanda kota di Jawa Barat.



5. Museum Geologi Bandung

Wisata yang didirikan pada 16 Mei 1929. Museum ini telah direnovasi dengan dana bantuan dari JICA (Japan International Cooperation Agency). Setelah mengalami renovasi, Museum Geologi dibuka kembali dan diresmikan oleh Wakil Presiden RI, Megawati

Soekarnoputri pada 23 Agustus 2000.



6. Bukit Moko

Sebuah kawasan yang berada di puncak sebuah bukit di daerah Cimenyan Bandung. Ketinggian Bukit Moko hampir mencapai 1.500 meter di atas permukaan air laut, oleh karenanya objek wisata alam di Bandung ini termasuk daerah yang berhawa sangat sejuk, sehingga jangan heran jika kawasan wisata ini selalu tampak diselimuti kabut tebal, terutama di waktu pagi, menjelang sore dan di malam hari.



7. Taman Balai Kota Bandung

sebuah taman kota yang terletak di Kota Bandung. Taman ini berada di sebelah selatan Balai Kota Bandung. Saat ini, Taman Balai Kota Bandung terdiri dari dua taman yang menyatu, antara lain Taman Dewi Sartika dan Taman Badak.



8. The Great Asia Afrika

The Great Asia Afrika Lembang merupakan tempat wisata dengan konsep edukasi budaya. Pengunjung bisa melihat keragaman budaya negara-negara di benua Asia

dan Afrika melalui suasana dan arsitektur rumah-rumah tradisional.



9. Jalan Braga

Nama sebuah jalan utama di kota Bandung, Indonesia. Nama jalan ini cukup dikenal sejak masa pemerintahan Hindia Belanda. Sampai saat ini nama jalan tersebut tetap dipertahankan sebagai salah satu maskot dan objek wisata kota Bandung yang dahulu dikenal sebagai *Parijs van Java*.



III. APLIKASI ALGORITMA DALAM MENENTUKAN RUTE PERJALANAN

Setelah dilakukan pengecekan pada Gmaps maka didapatkan jarak untuk tiap tempat wisata sebagai berikut.

v\j	1	2	3	4	5	6	7	8	9	10
1	0	8.2	16.4	21.6	8.3	10.1	18.9	9.7	34	9.8
2	11.8	0	14.3	18.2	7.1	7.8	24	8.5	18.9	8.5
3	10.2	12.2	0	5.5	6.8	6.9	11.4	8.3	8.4	8.8
4	9.4	18.4	8.3	0	12.9	11	17	14.1	2.7	14.1
5	6.6	4.7	7	12.2	0	0.2	11.5	2.1	11.8	2.1
6	7.6	5.8	7.7	12.9	0.8	0	11.2	5.1	12.7	3.1
7	17.8	14	11.4	16.2	11.6	11.7	0	13.9	19.1	14.1
8	6.5	4.9	8.1	13.3	1.9	2.8	13.5	0	11.6	1.7
9	6.3	15.6	9	2.7	11.5	11.6	20.1	11.9	0	11.9
10	7.2	4	8.9	14.3	2.7	3.6	14.9	0.8	12.8	0

Dengan indeks sebagai berikut dan asumsi perjalanan dimulai dari kampus Telkom.

1. Telkom University
2. Trans Studio Bandung (Cibangkong, Batununggal)
3. Dago Dream Park (Mekarwangi, Pagerwangi)
4. Floating Market Lembang (Kabupaten Bandung Barat)
5. Gedung Sate
6. Museum Geologi Bandung (Cihaur Geulis, Kec. Cibeunying Kaler)

7. Bukit Moko
8. Taman Balai Kota Bandung (Kota Bandung)
9. The Great Asia Afrika (Gudangkahuripan, Lembang, Kabupaten Bandung Barat)
10. Jalan Braga (Kota Bandung)

Akan dicari rute perjalanan terbaik yang dimulai dari Telkom University dengan mengunjungi semua tempat wisata dan kembali lagi ke Telkom University dengan menggunakan algoritma *Branch and Bound* dan *Greedy*.

A. Algoritma Branch and Bound

Pseudocode Branch and Bound

```
PROCEDURE copyToFinal(in/out final_path: list, in
curr_path: list):
{Melakukan copy solusi sementara ke solusi final}
final_path[:N + 1]<-curr_path[:]
final_path[N]<-curr_path[0]
ENDFUNCTION
```

```
// Function to find the minimum edge cost
// having an end at the vertex i
```

```
FUNCTION firstMin(adj, i):
{Mengembalikan nilai minimal cost}
min<-maxsize
FOR k in range(N):
IF adj[i][k] < min and i != k
THEN
min<-adj[i][k]
ENDIF
return min
ENDFUNCTION
```

```
FUNCTION secondMin(adj, i):
{Mengembalikan nilai minimal cost kedua}
first, second<-maxsize, maxsize
FOR j in range(N):
IF i = j
THEN
continue
ENDIF
IF adj[i][j] <= first
THEN
second<-first
first<-adj[i][j]
ELSE IF (adj[i][j] <= second and adj[i][j] != first):
second<-adj[i][j]
return second
ENDFUNCTION
```

```
// function that takes as arguments:
// curr_bound -> lower bound of the root node
// curr_weight-> stores the weight of the path so far
// level-> current level WHILE moving ENDWHILE
// in the search space tree
// curr_path[] -> where the solution is being stored
// which would later be copied to final_path[]
```

```

Procedure TSPRec(adj, curr_bound, curr_weight,
    level, curr_path, visited):
{Melakukan procedure TSP secara rekursif}
global final_res

// Jika seluruh node sudah dikunjungi
IF level = N THEN
    IF adj[curr_path[level - 1]][curr_path[0]] != 0
        THEN
            curr_res<-curr_weight + adj[curr_path[level - 1]][
                curr_path[0]]
            IF curr_res < final_res
                THEN
                    copyToFinal(curr_path)
                    final_res<-curr_res
            ENDIF
        ENDIF
    return
ENDIF

//Pembangunan state space tree
FOR i in range(N):

    IF (adj[curr_path[level-1]][i] != 0 and
        THEN
            visited[i] = False):
            temp<-curr_bound
            curr_weight += adj[curr_path[level - 1]][i]
        ENDIF

        IF level = 1
            THEN
                curr_bound -= ((firstMin(adj, curr_path[level - 1]) +
                    firstMin(adj, i)) / 2)
            ELSE
                curr_bound -= ((secondMin(adj, curr_path[level -
1) +
                    firstMin(adj, i)) / 2)
            ENDIF

            IF curr_bound + curr_weight < final_res
                THEN
                    curr_path[level]<-i
                    visited[i]<-True

                    // level selanjutnya
                    TSPRec(adj, curr_bound, curr_weight,
                        level + 1, curr_path, visited)
                ENDIF

                curr_weight -= adj[curr_path[level - 1]][i]
                curr_bound<-temp

                visited<-[False] * len(visited)
                FOR j in range(level):
                    IF curr_path[j] != -1
                        THEN

```

```

        visited[curr_path[j]]<-True
    ENDIF
ENDFUNCTION

FUNCTION TSP(adj):
    {Mengerjakan TSP}

    curr_bound<-0
    curr_path<-[-1] * (N + 1)
    visited<-[False] * N

    //Mencari lower bound untuk setiap simpul
    FOR i in range(N):
        curr_bound += (firstMin(adj, i) +
            secondMin(adj, i))

    //Mengubah lower bound menjadi integer
    curr_bound<-math.ceil(curr_bound / 2)

    visited[0]<-True
    curr_path[0]<-0
    TSPRec(adj, curr_bound, 0, 1, curr_path, visited)
ENDFUNCTION

adj<-np.array(df)
N<-10

final_path<-[None] * (N + 1)

visited<-[False] * N

final_res<-maxsize

TSP(adj)

OUTPUT "Minimum cost :", final_res
OUTPUT "Path Taken : ", end<-' '
FOR i 0 TO N + 1
    IF (i < N)
        THEN
            OUTPUT final_path[i]+1, end<-' -> '
        ELSE
            OUTPUT final_path[i]+1
        ENDIF
    NEXT

```

B. Algoritma Greedy

Komponen-komponen greedy:

1. Himpunan kandidat: Himpunan tempat wisata yang dapat dipilih
2. Himpunan solusi: Himpunan tempat wisata secara berurut dari kota asal dan kembali ke kota asal dengan jarak terpendek.
3. Fungsi seleksi
Memilih kota dengan jarak terpendek
4. Fungsi kelayakan

Memilih tempat wisata tepat sekali kecuali untuk kampus.

5. Fungsi objektif

Meminimasi rute perjalanan dari semua tempat wisata

PROCEDURE findMinRoute(tsp):

{ Mencari nilai route minimal dengan pendekatan greedy }

```

sum<-0
counter<-0
i<-0
j<-0
mn<-999999999
visitedRouteList<-{ }
visitedRouteList[0]<-1
route<-[0] *len(tsp)
WHILE i < len(tsp) and j < len(tsp[i]) DO
  IF counter >= len(tsp[i]) - 1
    THEN
      break //This might be better as a repeat loop
    ENDIF
  IF j != i and j not in visitedRouteList
    THEN
      IF tsp[i][j] < mn
        THEN
          mn<-tsp[i][j]
          route[counter]<-j + 1
        ENDIF
      ENDIF
    j += 1
    IF j = len(tsp[i])
      THEN
        sum += mn
        mn<-999999999
        visitedRouteList[route[counter] - 1]<-1
        j<-0
        i<-route[counter] - 1
        counter += 1
      ENDIF
    ENDWHILE
  ENDFUNCTION

i<-route[counter - 1] - 1
FOR j 0 TO 0, len(tsp)
  IF i != j and tsp[i][j] < mn
    THEN
      mn<-tsp[i][j]
      route[counter]<-j + 1
    ENDIF
NEXT
sum += mn
OUTPUT sum

```

```

Implementasi Branch and Bound
-----
Minimum cost : 56.1
Path Taken : 1 -> 2 -> 5 -> 10 -> 8 -> 6 -> 7 -> 3 -> 4 -> 9 -> 1

Telkom University -> Trans Studio Bandung -> Gedung Sate -> Jalan Braga -> Taman Balai Kota Bandung -> Museum Geologi Bandung -> Bukit Noko -> Dago Dream Park -> Floating Market Lembang -> The Great Asia Afrika -> Telkom University

```

2. Greedy

```

Implementasi Greedy
-----
Minimum route: 66.9
Path Taken : 1 -> 2 -> 5 -> 6 -> 8 -> 10 -> 3 -> 4 -> 9 -> 7 -> 1

Telkom University -> Trans Studio Bandung -> Gedung Sate -> Museum Geologi Bandung -> Taman Balai Kota Bandung -> Jalan Braga -> Dago Dream Park -> Floating Market Lembang -> The Great Asia Afrika -> Bukit Noko -> Telkom University

```

IV. ANALISIS

A. Perhitungan analisis kompleksitas algoritma dalam $T(n)$

1) Strategi Branch and Bound

Pada konstruksi pohon ruang status dengan pendekatan BnB, setidaknya ada dua hal yang perlu untuk diperhatikan:

a) Level pohon

Level pohon pada kasus TSP adalah sebanyak n sesuai dengan jumlah tempat.

b) Jumlah anak

Jumlah anak simpul pada kasus TSP dengan *Branch and Bound* dalam kondisi *worst-case* sama dengan pendekatan *brute-force* yaitu $n!$. Namun, bisa berubah seiring dengan pruning state space tree tergantung dengan nilai lower boundnya.

2) Strategi Greedy

Jika pengurutan diabaikan, maka pada algoritma Greedy hanya melakukan pengecekan sebanyak $T(n) = n(n+(n-1)+\dots+1)$

B. Kelas efisiensi dalam bentuk $O(n)$

1) Branch and Bound

Pada BnB kompleksitas waktunya dalam $O(n)$ adalah $O(n \cdot n!)$.

2) Greedy

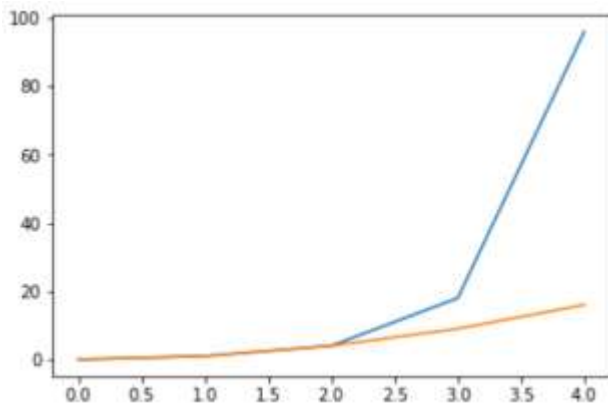
Pada *greedy*, kompleksitas waktunya dalam $O(n)$ adalah $O(n^2)$

C. Perbandingan kompleksitas

Dapat dilihat bahwa kompleksitas pada *greedy* lebih mangkus daripada *branch and bound*. Namun, penyelesaian pada *greedy* tidak selalu optimal atau global optima.

C. Output dari implementasi pemrograman dua strategi

1. Branch and Bound



Keterangan:

Biru: Kompleksitas BnB

Kuning: kompleksitas greedy

VIDEO LINK AT YOUTUBE

<https://www.youtube.com/watch?v=lougmuVrEP0>

LINK GITHUB

https://github.com/khalilullahalfaath/SA_Tubes

LINK PPT

https://www.canva.com/design/DAFEgwEC_SA/4uV6eH9Jk45v2i_O1pbkPQ/view?utm_content=DAFEgwEC_SA&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink

UCAPAN TERIMA KASIH

Segala puji dan syukur kami panjatkan ke hadirat Tuhan Yang Mahakuasa atas petunjuk yang diberikan kepada kami agar makalah ini dapat diselesaikan dengan baik. Ucapan terima kasih kami haturkan kepada Ibu Siti Sa'adah, S.T., M.T., selaku dosen pengampu mata kuliah Strategi Algoritma,

keluarga dan teman-teman penulis yang bantuannya tidak mampu kami sebutkan satu persatu.

REFERENSI

- [1] Daniells, L. (2020, April 21). *The Travelling Salesman Problem*. Retrieved from lancaster.ac.uk: <https://www.lancaster.ac.uk/stor-i-student-sites/libby-daniells/2020/04/21/the-travelling-salesman-problem/>
- [2] GeeksforGeeks admin. (2022, Maret 25). *https://www.geeksforgeeks.org/traveling-salesman-problem-using-branch-and-bound-2/*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/traveling-salesman-problem-using-branch-and-bound-2/>
- [3] GeeksforGeeks admin. (2022, January 2022). *Travelling Salesman Problem | Greedy Approach*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/travelling-salesman-problem-greedy-approach/>
- [4] Munir, R. (2022, January). *Algoritma Greedy (Bagian 1)*. Retrieved June 23, 2022, from informatika.stei.itb.ac.id: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-\(2022\)-Bag3.pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-(2022)-Bag3.pdf)
- [5] Wulandari, G. S., & Sa'adah, S. (2021). *Buku Pengantar Strategi Algoritma*. In G. S. Wulandari, & S. Sa'adah, *Buku Pengantar Strategi Algoritma* (1 ed., p. 133). Bandung, Jawa Barat, Indonesia: CV. Karya Bakti Makmur Indonesia. Retrieved June 21, 2022, from https://drive.google.com/file/d/1OIaEBWRhIkWD0HlqbBOKTk3t_26p4sfs/view

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan kami sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2022

Tim penulis Kelompok 3