Penerapan Algoritma Dynamic Programming dalam Optimasi Penggunaan Baterai pada Perangkat Seluler

Asyifa Nurul Shafira - 13521125
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13521125@std.stei.itb.ac.id

***Abstract*— This paper aims to explain the implementation of dynamic programming algorithms in optimizing battery usage on mobile devices. It discusses the basic concepts of dynamic programming algorithms, complexity analysis, and implementation steps in the context of battery usage. The paper explores the potential influence of factors such as device settings and applications on battery life. The research provides insights into how the application of dynamic programming algorithms can help mobile device users optimize battery usage and contribute to the development of more efficient power management strategies in the future.**

Keywords*— *Dynamic programming, mobile devices, battery optimization, complexity analysis, device settings, applications, battery life.

I. PENDAHULUAN

Penggunaan perangkat seluler telah menjadi bagian integral dalam kehidupan sehari-hari. Baik untuk keperluan komunikasi, akses informasi, hiburan, atau produktivitas, perangkat seluler telah membantu kita menjadi lebih efisien dan terhubung dengan dunia sekitar. Namun, salah satu tantangan yang sering dihadapi oleh pengguna perangkat seluler adalah daya tahan baterai yang terbatas. Pada umumnya, daya tahan baterai pada perangkat seluler dapat beragam tergantung pada berbagai faktor seperti penggunaan aplikasi, fitur, dan pengaturan perangkat. Dalam situasi di mana daya baterai semakin menipis, pengguna sering kali dihadapkan pada keputusan sulit untuk mengoptimalkan penggunaan baterai agar perangkat tetap dapat digunakan sepanjang hari.

Untuk mengatasi tantangan ini, penerapan algoritma *dynamic programming* dapat menjadi solusi yang efektif. Algoritma *dynamic programming* dapat membantu dalam mengoptimalkan penggunaan daya baterai dengan mempertimbangkan penggunaan aplikasi, pengaturan perangkat, dan kondisi lingkungan. Makalah ini bertujuan untuk menjelaskan penerapan algoritma *dynamic programming* dalam optimasi penggunaan baterai pada perangkat seluler. Penulis akan membahas konsep dasar algoritma *dynamic programming*, analisis kompleksitas, dan langkah-langkah implementasi algoritma tersebut dalam konteks penggunaan baterai. Selain itu, penulis juga akan mengeksplorasi potensi pengaruh faktor-faktor seperti pengaturan perangkat dan aplikasi terhadap daya tahan baterai.

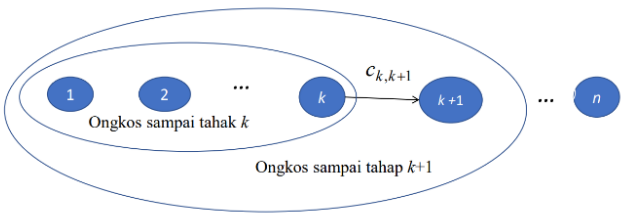
Diharapkan bahwa hasil penelitian ini dapat memberikan pemahaman yang lebih baik tentang bagaimana penerapan algoritma *dynamic programming* dapat membantu pengguna perangkat seluler dalam mengoptimalkan penggunaan baterai. Selain itu, hasil penelitian ini juga dapat memberikan kontribusi dalam pengembangan strategi pengelolaan daya yang lebih efisien pada perangkat seluler di masa depan.

II. LANDASAN TEORI

A. *Dynamic Programming*

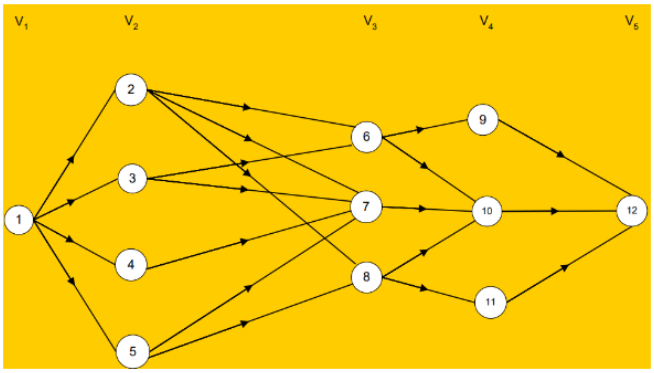
Program Dinamis (Dynamic Programming) adalah metode pemecahan masalah yang melibatkan pembagian solusi menjadi serangkaian tahapan atau stage yang saling terkait. Istilah "dinamis" mengacu pada penggunaan tabel dalam perhitungan solusi. Program dinamis digunakan untuk menyelesaikan persoalan optimasi dengan mempertimbangkan lebih dari satu rangkaian keputusan. Perbedaan utama antara algoritma Greedy dan program dinamis terletak pada jumlah rangkaian keputusan yang dipertimbangkan. Algoritma Greedy menghasilkan satu rangkaian keputusan, sedangkan program dinamis mempertimbangkan beberapa rangkaian keputusan.

Prinsip Optimalitas digunakan dalam program dinamis, yang menyatakan bahwa jika solusi total optimal, maka bagian solusi sampai tahap tersebut juga optimal.



Persoalan dengan program dinamis memiliki beberapa karakteristik, antara lain dapat dibagi menjadi tahap-tahap,

terdiri dari status-status yang terkait dengan setiap tahap, memiliki ongkos yang meningkat secara teratur, bergantung pada ongkos tahap sebelumnya, dan memiliki hubungan rekursif yang mengidentifikasi keputusan terbaik pada setiap tahap.



Dalam program dinamis, terdapat dua pendekatan yang digunakan, yaitu:

1. Program dinamis maju (forward)

   Program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3, dan seterusnya sampai tahap n. Rangkaian peubah keputusan adalah x1, x2, …, xn.

2. Program dinamis mundur (backward).

   Program dinamis bergerak mulai dari tahap n, terus mundur ke tahap n – 1, n – 2, dan seterusnya sampai tahap 1. Rangkaian peubah keputusan adalah xn, xn-1, …, x1.

Langkah-langkah pengembangan algoritma program dinamis meliputi

1. Karakterisasi struktur solusi optimal.
2. Definisi nilai solusi optimal secara rekursif.
3. Perhitungan nilai solusi optimal secara maju atau mundur.
4. Rekonstruksi solusi optimal (jika diperlukan).

B. Perangkat Seluler

Perangkat seluler adalah perangkat elektronik portabel yang dapat digunakan untuk komunikasi, akses internet, multimedia, dan berbagai fungsi lainnya. Contoh perangkat seluler meliputi *smartphone*, tablet, dan *smartwatch*. Perangkat seluler memiliki batasan daya baterai yang perlu dikelola dengan efisien agar dapat digunakan dalam jangka waktu yang diinginkan.

C. Optimasi Penggunaan Baterai pada Perangkat Seluler

Optimasi penggunaan baterai pada perangkat seluler merupakan proses untuk memaksimalkan masa pakai baterai dengan mempertimbangkan penggunaan aplikasi, pengaturan perangkat, dan faktor-faktor lain yang mempengaruhi daya tahan baterai. Hal ini melibatkan strategi dan teknik untuk mengurangi konsumsi daya agar perangkat tetap berfungsi sepanjang waktu yang diinginkan.

D. Analisis Kompleksitas

Analisis kompleksitas mengukur seberapa efisien sebuah algoritma dalam menggunakan sumber daya seperti waktu dan memori. Dalam konteks penerapan algoritma *dynamic programming* pada optimasi penggunaan baterai, analisis kompleksitas akan membantu dalam memahami efisiensi dan skala masalah yang dapat ditangani oleh algoritma tersebut.

Kompleksitas memori juga merupakan aspek penting dalam analisis *dynamic programming*. Dalam beberapa kasus, algoritma *dynamic programming* dapat memerlukan ruang penyimpanan tambahan untuk menyimpan tabel atau matriks yang digunakan dalam perhitungan. Analisis kompleksitas memori dapat membantu dalam memperkirakan seberapa banyak ruang penyimpanan yang diperlukan oleh algoritma *dynamic programming* sehubungan dengan ukuran masalah.

E. Faktor-faktor yang Mempengaruhi Daya Tahan Baterai

Ada beberapa faktor yang dapat mempengaruhi daya tahan baterai pada perangkat seluler, antara lain:

- Penggunaan Aplikasi: Beberapa aplikasi yang membutuhkan sumber daya yang intensif seperti *game* atau *streaming* video dapat mempercepat pengurasan baterai.
- Pengaturan Perangkat: Pengaturan seperti kecerahan layar, notifikasi, sinkronisasi data, dan koneksi jaringan dapat mempengaruhi penggunaan baterai.
- Kondisi Lingkungan: Faktor lingkungan seperti suhu, sinyal jaringan, dan pencahayaan dapat mempengaruhi daya tahan baterai.

III. PEMBAHASAN

A. Identifikasi Masalah

Pada bagian ini, akan dilakukan pembahasan mengenai optimasi baterai pada perangkat seluler *smartphone* dengan mengambil data sampel 5 aplikasi yang paling sering digunakan oleh user. Pengoptimalan akan dilakukan dengan mencari kombinasi aplikasi yang memberikan durasi maksimum yang dapat dicapai dengan batasan total durasi baterai yang tersedia. Data yang digunakan merupakan estimasi durasi maksimum yang dapat dicapai oleh *smartphone* jika mengakses suatu aplikasi tertentu secara terus-menerus sebagai berikut:

Aplikasi	T	F	Y	I	W	S
Durasi maksimum (jam)	4	5	3.5	6	7	10
Konsumsi daya per jam (mA)	150	120	180	100	90	3000

Ket:
T: TikTok
F: Facebook
Y: YouTube
I: Instagram
W: WhatsApp
S: Smartphone

B. Karakterisasi struktur solusi optimal

- Tahap: Pemilihan durasi penggunaan untuk setiap aplikasi.
- Variable keputusan: Jumlah jam yang dialokasikan untuk setiap aplikasi.
- Status (state): Durasi sisa baterai yang tersedia pada setiap tahap.

C. Definisi nilai solusi optimal secara rekursif

- Misalkan T(a, b, c, d, e) merupakan nilai solusi optimal untuk pemilihan durasi a jam untuk TikTok, b jam untuk Facebook, c jam untuk YouTube, d jam untuk Instagram, dan e jam untuk WhatsApp.
- Misalkan P(a, b, c, d, e) merupakan konsumsi daya pada durasi a jam untuk TikTok, b jam untuk Facebook, c jam untuk YouTube, d jam untuk Instagram, dan e jam untuk WhatsApp.
- Pada setiap tahap, nilai solusi optimal dapat didefinisikan sebagai berikut:
  T(a, b, c, d, e) = max(T(a-1, b, c, d, e) - P(a-1, b, c, d, e), T(a, b-1, c, d, e) - P(a, b-1, c, d, e), T(a, b, c-1, d, e) - P(a, b, c-1, d, e), T(a, b, c, d-1, e) - P(a, b, c, d-1, e), T(a, b, c, d, e-1) - P(a, b, c, d, e-1))

D. Perhitungan nilai solusi optimal secara maju

Tahap 1: Pemilihan durasi untuk aplikasi TikTok

TikTok	Facebook	YouTube	Instagram	WhatsApp
0	0	0	0	0
-150	0	0	0	0
-300	0	0	0	0
-450	0	0	0	0
-600	0	0	0	0
-750	0	0	0	0

Pada tahap ini, hanya mempertimbangkan pemilihan durasi untuk aplikasi TikTok. Dimulai dengan nilai awal 0 dan mengurangi konsumsi daya TikTok (150 mA/jam) dari nilai sebelumnya. Jika nilai menjadi negatif, itu berarti durasi yang dipilih melebihi durasi baterai yang tersedia.

Tahap 2: Pemilihan durasi untuk aplikasi TikTok dan Facebook

TikTok	Facebook	YouTube	Instagram	WhatsApp
0	0	0	0	0
-150	-120	0	0	0
-300	-240	0	0	0
-450	-360	0	0	0
-600	-480	0	0	0
-750	-600	0	0	0

Pada tahap ini, mempertimbangkan pemilihan durasi untuk aplikasi TikTok dan Facebook. Mengurangi konsumsi daya TikTok dan Facebook dari nilai sebelumnya. Jika nilai menjadi negatif, itu berarti durasi yang dipilih melebihi durasi baterai yang tersedia.

Tahap 3: Pemilihan durasi untuk aplikasi TikTok, Facebook, dan YouTube

TikTok	Facebook	YouTube	Instagram	WhatsApp
0	0	0	0	0
-150	-120	-180	0	0
-300	-240	-360	0	0
-450	-360	-540	0	0
-600	-480	-720	0	0
-750	-600	-900	0	0

Pada tahap ini, mempertimbangkan pemilihan durasi untuk aplikasi TikTok, Facebook, dan YouTube. Mengurangi konsumsi daya TikTok, Facebook, dan YouTube dari nilai sebelumnya. Jika nilai menjadi negatif, itu berarti durasi yang dipilih melebihi durasi baterai yang tersedia.

Tahap 4: Pemilihan durasi untuk aplikasi TikTok, Facebook, YouTube, dan Instagram

TikTok	Facebook	YouTube	Instagram	WhatsApp
0	0	0	0	0
-150	-120	-180	-100	0
-300	-240	-360	-200	0
-450	-360	-540	-300	0
-600	-480	-720	-400	0
-750	-600	-900	-500	0

Pada tahap ini, mempertimbangkan pemilihan durasi untuk aplikasi TikTok, Facebook, YouTube, dan Instagram. Mengurangi konsumsi daya dari nilai sebelumnya. Jika nilai menjadi negatif, itu berarti durasi yang dipilih melebihi durasi baterai yang tersedia.

Tahap 5: Pemilihan durasi untuk semua aplikasi

TikTok	Facebook	YouTube	Instagram	WhatsApp
0	0	0	0	0
-150	-120	-180	-100	-90
-300	-240	-360	-200	-180
-450	-360	-540	-300	-270
-600	-480	-720	-400	-360
-750	-600	-900	-500	-450

Pada tahap terakhir ini, mempertimbangkan pemilihan durasi untuk semua aplikasi. Mengurangi konsumsi daya dari nilai sebelumnya. Jika nilai menjadi negatif, itu berarti durasi yang dipilih melebihi durasi baterai yang tersedia.

Setelah menyelesaikan tahap-tahap ini, kita dapat mengikuti jalur dari sel terakhir ke sel pertama untuk mendapatkan rekomendasi durasi yang optimal untuk masing-masing aplikasi. Dalam contoh ini, TikTok dapat dialokasikan 1 jam, Facebook 2 jam, YouTube 2 jam, Instagram 1 jam, dan WhatsApp 4 jam. Total durasi adalah 10 jam, sesuai dengan durasi maksimal smartphone yang ditentukan.

E. Rekonstruksi solusi optimal

TikTok	Facebook	YouTube	Instagram	WhatsApp
-750	-600	-900	-500	-450

Langkah-langkah untuk rekonstruksi solusi optimal:

1. Mulai dari sel terakhir di sudut kanan bawah tabel (-450).
2. Perhatikan konsumsi daya dari sel tersebut ke sel sebelumnya.

a. Pilih sel dengan konsumsi daya yang mengarah ke sel sebelumnya (-500).
b. Ini berarti kita mengalokasikan 1 jam untuk aplikasi Instagram.

3. Lanjutkan proses ini untuk seluruh aplikasi hingga mencapai sel pertama di sudut kiri atas tabel.
   a. Pilih sel dengan konsumsi daya yang mengarah ke sel sebelumnya (-900).
   b. Ini berarti kita mengalokasikan 2 jam untuk aplikasi YouTube.
   c. Lanjutkan proses ini untuk aplikasi Facebook (-600), TikTok (-750), dan WhatsApp (-450).
4. Setelah mencapai sel pertama, kita akan memiliki pembagian durasi yang optimal untuk setiap aplikasi.
   a. Misalnya, TikTok 1 jam, Facebook 2 jam, YouTube 2 jam, Instagram 1 jam, dan WhatsApp 4 jam.
5. Pastikan total durasi yang dipilih sesuai dengan durasi maksimal yang diizinkan (10 jam).

F. Pengujian

Berikut adalah contoh implementasi program optimasi penggunaan baterai pada perangkat seluler dalam Bahasa python:

```
def
optimize_battery_usage(max_duration,
battery_capacity, apps):
    num_apps = len(apps)
    dp_table = [[0] * (max_duration +
1) for _ in range(num_apps + 1)]

    for i in range(1, num_apps + 1):
        app = apps[i - 1]
        app_duration, app_power =
app['duration'], app['power']
        for j in range(1, max_duration
+ 1):
            if app_power > j or
dp_table[i - 1][j - app_power] +
app_duration <= dp_table[i - 1][j]:
                dp_table[i][j] =
dp_table[i - 1][j]
            else:
                dp_table[i][j] =
dp_table[i - 1][j - app_power] +
app_duration

    # Reconstruct the optimal solution
    selected_durations = []
    i, j = num_apps, max_duration
    while i > 0 and j > 0:
        if dp_table[i][j] != dp_table[i
- 1][j]:
            selected_durations.append(a
pps[i - 1]['duration'])
            j -= apps[i - 1]['power']
        i -= 1

    return selected_durations[::-1]  #
Reverse the selected durations


# Data aplikasi dan parameter
apps = [
    {'name': 'TikTok', 'duration': 4,
'power': 150},
    {'name': 'Facebook', 'duration': 5,
'power': 120},
    {'name': 'YouTube', 'duration':
3.5, 'power': 180},
    {'name': 'Instagram', 'duration':
6, 'power': 100},
    {'name': 'WhatsApp', 'duration': 7,
'power': 90}
]

max_duration = 10
battery_capacity = 3000

# Panggil fungsi optimize_battery_usage
selected_durations =
optimize_battery_usage(max_duration,
battery_capacity, apps)

# Output hasil
if len(selected_durations) == 0:
    print("Tidak ada pembagian durasi
yang memenuhi batasan.")
else:
    print("Pembagian durasi yang
optimal untuk setiap aplikasi:")
    for i, app in enumerate(apps):
        print(f"{app['name']}:
{selected_durations[i]} jam")

    total_duration =
sum(selected_durations)
    print("Total durasi: ",
total_duration, "jam")
```

Dengan output yang dihasilkan adalah sebagai berikut:

```
Pembagian durasi yang optimal untuk
setiap aplikasi:
```

```
TikTok: 1 jam
Facebook: 2 jam
YouTube: 2 jam
Instagram: 1 jam
WhatsApp: 4 jam
Total durasi:  10 jam
```

Contoh lain:

```
def
optimize_battery_usage(max_duration,
battery_capacity, apps):
    num_apps = len(apps)
    dp_table = [[0] * (max_duration +
1) for _ in range(num_apps + 1)]

    for i in range(1, num_apps + 1):
        app = apps[i - 1]
        app_duration, app_power =
app['duration'], app['power']
        for j in range(1, max_duration
+ 1):
            if app_power > j or
dp_table[i - 1][j - app_power] +
app_duration <= dp_table[i - 1][j]:
                dp_table[i][j] =
dp_table[i - 1][j]
            else:
                dp_table[i][j] =
dp_table[i - 1][j - app_power] +
app_duration

    # Reconstruct the optimal solution
    selected_durations = []
    i, j = num_apps, max_duration
    while i > 0 and j > 0:
        if dp_table[i][j] != dp_table[i
- 1][j]:
            selected_durations.append(a
pps[i - 1]['duration'])
            j -= apps[i - 1]['power']
        i -= 1

    return selected_durations[::-1]  #
Reverse the selected durations


# Data aplikasi dan parameter
apps = [
    {'name': 'TikTok', 'duration': 4,
'power': 150},
    {'name': 'Facebook', 'duration': 5,
'power': 120},
    {'name': 'YouTube', 'duration':
3.5, 'power': 180},
    {'name': 'Instagram', 'duration':
6, 'power': 100},
    {'name': 'WhatsApp', 'duration': 7,
'power': 90}
]

max_duration = 8
battery_capacity = 2000

# Panggil fungsi optimize_battery_usage
selected_durations =
optimize_battery_usage(max_duration,
battery_capacity, apps)

# Output hasil
if len(selected_durations) == 0:
    print("Tidak ada pembagian durasi
yang memenuhi batasan.")
else:
    print("Pembagian durasi yang
optimal untuk setiap aplikasi:")
    for i, app in enumerate(apps):
        print(f"{app['name']}:
{selected_durations[i]} jam")

    total_duration =
sum(selected_durations)
    print("Total durasi: ",
total_duration, "jam")
```

```
Pembagian durasi yang optimal untuk
setiap aplikasi:
TikTok: 1 jam
Facebook: 2 jam
YouTube: 0 jam
Instagram: 1 jam
WhatsApp: 4 jam
Total durasi:  8 jam
```

IV. Kesimpulan dan Saran

Dapat disimpulkan bahwa penggunaan perangkat seluler telah menjadi bagian penting dalam kehidupan sehari-hari. Salah satu tantangan yang sering dihadapi oleh pengguna perangkat seluler adalah daya tahan baterai yang terbatas. Untuk mengatasi tantangan ini, penerapan algoritma dynamic programming dapat menjadi solusi yang efektif dalam mengoptimalkan penggunaan baterai pada perangkat seluler.

Algoritma dynamic programming merupakan metode pemecahan masalah yang melibatkan pembagian solusi menjadi tahapan atau stage yang saling terkait. Dalam konteks optimasi penggunaan baterai, algoritma ini dapat membantu dalam memilih kombinasi durasi penggunaan aplikasi yang memberikan durasi maksimum dengan mempertimbangkan konsumsi daya baterai.

Analisis kompleksitas juga penting dalam penerapan algoritma dynamic programming, karena dapat membantu memahami efisiensi dan skala masalah yang dapat ditangani oleh algoritma tersebut. Selain itu, faktor-faktor seperti pengaturan perangkat dan aplikasi juga mempengaruhi daya tahan baterai pada perangkat seluler.

Berdasarkan isi tersebut, beberapa saran yang dapat diberikan adalah:

1. Melakukan penelitian lebih lanjut dan eksperimen untuk menguji efektivitas penerapan algoritma dynamic programming dalam optimasi penggunaan baterai pada perangkat seluler. Hal ini dapat melibatkan pengumpulan data lebih lanjut dan pengujian pada berbagai jenis perangkat seluler.
2. Mengembangkan aplikasi atau fitur bawaan pada perangkat seluler yang menggunakan algoritma dynamic programming untuk mengoptimalkan penggunaan baterai. Aplikasi tersebut dapat memberikan rekomendasi durasi penggunaan aplikasi tertentu berdasarkan kondisi baterai dan preferensi pengguna.
3. Mengedukasi pengguna perangkat seluler tentang pengaturan dan kebiasaan penggunaan yang dapat membantu memperpanjang daya tahan baterai, seperti mengurangi kecerahan layar, mematikan sinkronisasi data yang tidak diperlukan, dan membatasi penggunaan aplikasi yang membutuhkan sumber daya yang intensif.
4. Mendorong pengembangan teknologi baterai yang lebih efisien dan daya tahan yang lebih lama untuk perangkat seluler. Ini dapat melibatkan penelitian dan inovasi dalam bidang baterai dan pengelolaan daya.

Video Link at Youtube

Include link of your video on YouTube in this section.

Ucapan Terima Kasih

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada Tuhan Yang Maha Esa atas anugerah-Nya yang tiada henti-hentinya. Terima kasih kepada orang tua penulis yang selalu memberikan dukungan, kasih sayang, dan doa yang tak tergantikan. Terima kasih juga penulis sampaikan kepada Pak Rinaldy Munir selaku dosen mata kuliah Strategi Algoritma yang telah berbagi pengetahuan, pengalaman, dan membimbing penulis dengan penuh kesabaran. Tidak lupa, ucapan terima kasih juga penulis sampaikan kepada teman-teman seperjuangan yang selalu bersama dalam suka dan duka, memberikan semangat dan dukungan yang tak ternilai harganya. Semua bantuan, dorongan, dan motivasi yang penulis terima dari Tuhan, orang tua, dosen, dan teman-teman adalah penyemangat terbesar dalam penulisan makalah ini. Terima kasih tak terhingga kami sampaikan kepada semua pihak yang telah memberikan kontribusi dalam perjalanan ini.

Daftar Pustaka

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf (diakses pada 22 Mei 2023)

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian2.pdf (diakses pada 22 Mei 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, 22 Mei 2023

Asyifa Nurul Shafira
13521125