

Penerapan Algoritma Brute Force dan Greedy Best First Terhadap Persoalan Dominasi Ratu untuk N sama dengan Delapan

Fakhri Muhammad Mahendra - 13521045
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13521045@std.stei.itb.ac.id

Abstract—Metode penyelesaian masalah dengan teknik *brute force* dan *greedy best first* sudah menjadi alat yang digunakan programmer. Salah satu masalah yang bisa dipecahkan adalah masalah dominasi ratu di permainan catur. Masalah ini berisi tentang berapa jumlah ratu minimal yang bisa ditempatkan pada suatu papan catur 8×8 , sehingga seluruh area nya di tutupi oleh pergerakan ratu. Makalah ini akan membahas tentang penerapan algoritma *brute force* dan *greedy best first* untuk menyelesaikan masalah tersebut. Akan dibahas langkah yang diambil ketika melakukan eksplorasi, hingga solusi yang berhasil ditemukan.

Keywords—*component; brute force, greedy best first, queen domination problem.*

I. PENDAHULUAN

Catur merupakan permainan papan yang dimainkan oleh dua orang di atas papan delapan kali delapan yang tersusun dari 64 kotak. Catur memiliki sejarah yang panjang dan diyakini berasal dari India dari sekitar abad ke-7.

Permainan catur memiliki aturan yang spesifik terkait bagaimana 'buah catur' yang dimiliki masing-masing pemain bergerak di papan permainan. Pergerakan dari tiap buah catur yang berbeda memiliki aturan yang unik. Dari aturan pergerakan buah-buah catur ini dibuat beberapa permasalahan matematika. Salah satu permasalahan yang dimaksud adalah permasalahan dominasi ratu.

Permasalahan dominasi ratu mencari himpunan mendominasi minimum ratu dari papan catur n kali n . Sebuah catur mendominasi papan apabila seluruh kotak di papan tersebut berada di daerah gerak semua buah catur yang ada di himpunan tersebut. Ratu sebagai contoh memiliki pergerakan yang paling bebas di antara bidak catur lainnya. Sebuah ratu bisa bergerak secara horizontal, vertikal, ataupun diagonal sebanyak mungkin, selama jalur tersebut tidak di halangi oleh bidak catur lainnya.

Dalam menyelesaikan persoalan ini, terdapat strategi-strategi umum yang bisa diselesaikan pemecah masalah untuk mencari solusinya. Salah satunya adalah dengan teknik brute force. Teknik brute force mencari seluruh kemungkinan yang mungkin dari persoalan dan mengecek solusi nya satu persatu-

atau. Terdapat teknik lain lagi yang berbeda dengan bruteforce adalah greedy best first. Teknik greedy best first ini memiliki heuristik sehingga program bisa menemukan solusi dengan relatif lebih cepat daripada metode bruteforce.

Makalah akan membandingkan perbedaan aplikasi metode bruteforce dengan greedy best first. Perbedaan bisa dilihat dari berapa iterasi yang diperlukan program untuk mencapai solusi yang diinginkan

II. LANDASAN TEORI

A. Himpunan

Himpunan adalah koleksi tidak berurut dari objek-objek yang berbeda, disebut elemen atau anggota dari himpunan. Ditulis

$$a \in A$$

Jika dan hanya jika a merupakan elemen dari himpunan A , dan sebaliknya, ditulis

$$a \notin A$$

Jika dan hanya jika a bukan merupakan elemen dari himpunan A .

B. Kombinasi

kombinasi- r dari sebuah himpunan adalah pemilihan r elemen tak terurut dari himpunan tersebut. Banyak kombinasi- r dari sebuah himpunan dengan n elemen berbeda dinotasikan sebagai $C(n, r) = \binom{n}{r}$ yang juga disebut sebagai koefisien binomial. Dengan memakai aturan perkalian, kita memiliki

Banyak dari kombinasi- r dari sebuah himpunan dengan n elemen adalah

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

Dimana n bilangan bulat non-negatif dan r bilangan bulat

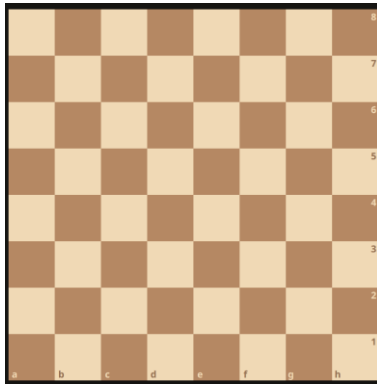
yang memenuhi $0 \leq r \leq n$.

C. Lower Bound

Sebuah elemen yang memiliki nilai kurang dari sama dengan semua elemen dari suatu himpunan

D. Papan Catur

Merupakan sebuah kotak berukuran n kali n dengan jumlah petak n^2 . Berikut merupakan contoh papan catur berukuran 8 x 8

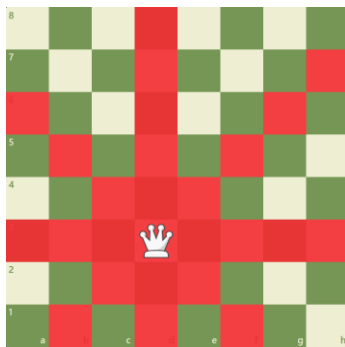


Gambar 1 Papan Catur 8 x 8
(Sumber: lichess.org)

Perlu diperhatikan bahwa papan catur pada umumnya adalah berukuran 8 x 8, namun ukuran papan catur yang dimaksud bisa berubah sesuai konteks yang diinginkan

E. Ratu

Ratu merupakan bidak catur yang memiliki arah gerak horizontal sejauh apapun, vertikal sejauh apapun, dan diagonal sejauh apapun. Arah gerak ratu bisa divisualisasikan sebagai berikut



Gambar 2 Arah Gerak Ratu
(Sumber: <https://www.chess.com/terms/chess-queen>)

F. Persoalan

Persoalan adalah pertanyaan atau tugas yang ingin kita cari permasalahannya

G. Instansi Persoalan

Instansi persoalan adalah parameter nilai yang diasosiasikan dengan persoalan

H. Algoritma

Algoritma adalah urutan langkah-langkah untuk memecahkan suatu persoalan, dengan memproses masukan menjadi luaran

I. Algoritma Brute Force

Algoritma pencarian brute force atau biasa disebut dengan exhaustive search adalah alat pemecahan masalah dengan secara sistematis meng-enumerasi semua kemungkinan solusi dan memverifikasi apakah tiap kandidat sesuai dengan yang diinginkan di persoalan [1].

Pencarian secara brute force biasanya memakan paling banyak waktu, karena semua kemungkinan solusi harus dibangkitkan satu per satu.

J. Algoritma Pencarian Greedy Best-First

Algoritma Greedy best first adalah pencarian yang berinformasi, dimana fungsi evaluasinya sama dengan fungsi heuristiknya [2]. Pada greedy best first, akan di ekspansi node yang kelihatan paling dekat dengan solusi.

K. Fungsi Pembatas

Fungsi pembatas adalah fungsi yang bisa menentukan kapan suatu node harus diekspansi lebih lanjut. atau langsung dibunuh.

L. Fungsi Heuristik

Fungsi heuristik adalah fungsi yang bisa mengestimasi nilai suatu node. Dengan heuristik diharapkan solusi bisa ditemukan secara lebih cepat tanpa mengecek semua kombinasi yang ada.

III. PERUMUSAN MASALAH

A. Deskripsi Persoalan

Pada makalah ini, persoalan yang ingin dicari solusinya adalah sebagai berikut

“Berapa jumlah ratu paling sedikit yang bisa diletakkan pada papan catur berukuran n x n sehingga dapat dipastikan setiap petak pada papan catur diserang oleh setidaknya satu ratu”

B. Instansi Persoalan

Pada makalah ini hanya akan dibahas untuk kasus nilai n = 8 saja, seperti pada papan catur pada umumnya.

IV. PEMBAHASAN

A. Pendefinisian Notasi

Pada makalah ini didefinisikan beberapa notasi atau istilah sebagai, yaitu sebagai berikut:

1) n

Menyatakan jumlah dimensi dari papan catur

2) Dominating Set

Menyatakan sebuah himpunan ratu yang diposisikan pada suatu papan catur di posisi yang berbeda sehingga semua petak yang ada di papan catur diserang oleh setidaknya satu ratu

3) $d(n)$

Merupakan banyak elemen paling sedikit yang mungkin dari suatu dominating set di papan catur $n \times n$

4) k

Jumlah ratu yang ada di papan pada suatu waktu

5) *Covered*

Merupakan istilah yang menandakan apakah semua petak di papan catur sudah diserang oleh setidaknya satu ratu atau belum

B. Penentuan Lower Bound dari $d(8)$

Sebelumnya akan ditentukan nilai paling kecil dari $d(8)$ yang mungkin. Untuk melakukan hal ini, akan diuji terlebih dahulu untuk berbagai nilai k yang kecil. Pengujian akan dilakukan secara exhaustive search, yaitu menghabiskan semua kombinasi pasangan yang mungkin.

1) Kasus $k = 3$

Dalam kasus ini akan di generasikan semua kemungkinan posisi dari 3 ratu pada papan catur 8×8 . Untuk melakukan hal tersebut, pilih 3 bilangan berbeda dari himpunan bilangan 0-63.

Jika sudah dipilih bilangan k_1 , k_2 , dan k_3 berbeda dari himpunan bilangan 0-63 maka lakukan transformasi berikut

$\text{baris}_i = k_i \text{ div } 8$

$\text{kolom}_i = k_i \text{ modulo } 8$

untuk $i = 1, 2$ dan 3

Setelah transformasi tersebut dilakukan maka tiap pasangan (baris_i , kolom_i) untuk $i = 1, 2$, dan 3 , merupakan koordinat valid untuk ratu k_1 , k_2 , dan k_3 , diletakkan pada papan catur.

Berikut merupakan pseudocode untuk membangkitkan semua kemungkinan dari kombinasi posisi 3 kartu.

```

iteration ← 0
FOR k1 = 0 to 61
  FOR k2 = k1 to 62
    FOR k3 = k2 to 63
      iteration ← iteration + 1
      Put queen k1
      Put queen k2
      Put queen k3
      IF papan is covered
        // (k1, k2, k3) is a dominating set
        → (k1, k2, k3)
      ENDIF
    ENDFOR
  ENDFOR
ENDFOR

```

```

ENDFOR

```

```

ENDFOR

```

```

Print("Solution not found")

```

```

Print(iteration)

```

Setelah program dijalankan, ternyata dominating set tidak pernah muncul sebagai keluaran. Yang keluar adalah banyak iterasi yang dilakukan program yaitu sebanyak 41664. Angka ini sama dengan nilai dari $64C3$, yaitu jumlah kombinasi memilih 3 angka dari 64 pilihan tanpa urutan diperhatikan.

Dari hasil ini bisa dipastikan bahwa nilai $d(8)$ lebih dari 3 atau bisa ditulis

$4 \leq d(8)$

Jadi didapat nilai lower bound sementara dari $d(8)$ adalah 4.

2) Kasus $k = 4$

Dari hasil percobaan selanjutnya, perlu diverifikasi kembali, apakah lower bound benar adalah 4, verifikasi bisa dilakukan dengan cara yang analog, yaitu memilih 4 bilangan bilangan berbeda dari himpunan bilangan 0-63.

Andai jika bilangan tersebut adalah k_1 , k_2 , k_3 , dan k_4 . Maka dengan cara transformasi yang sama dengan sebelumnya, ubah bilangan-bilangan tersebut menjadi bentuk koordinat sehingga bisa merepresentasikan posisi ratu pada papan catur.

Pseudocode untuk membangkitkan semua kombinasi posisi 4 kartu adalah sebagai berikut

```

iteration ← 0
FOR k1 = 0 to 60
  FOR k2 = k1 to 61
    FOR k3 = k2 to 62
      FOR k4 = k3 to 63
        iteration ← iteration + 1
        Put queen k1
        Put queen k2
        Put queen k3
        Put queen k4
        IF papan is covered
          // (k1, k2, k3, k4) is a dominating set
          → (k1, k2, k3, k4)
        ENDIF
      ENDFOR
    ENDFOR
  ENDFOR
ENDFOR
Print("Solution not found")

```

```
Print(iteration)
```

Setelah program dijalankan, yang keluar adalah hasil banyak jumlah iterasi nya karena tidak ditemukan dominating set dari 4 ratu yang ada di papan. Nilai dari jumlah iterasi tersebut adalah 635376, yaitu sejumlah 64 kombinasi 4.

Maka dari hasil percobaan tersebut didapat dengan exhaustive search bahwa $d(8) > 4$. Yaitu 4 buah ratu tidak cukup untuk menjadi dominating set dari papan catur 8×8 .

Untuk sementara ditetapkan lower bound sama dengan 5.

C. Aplikasi Algoritma Brute Force

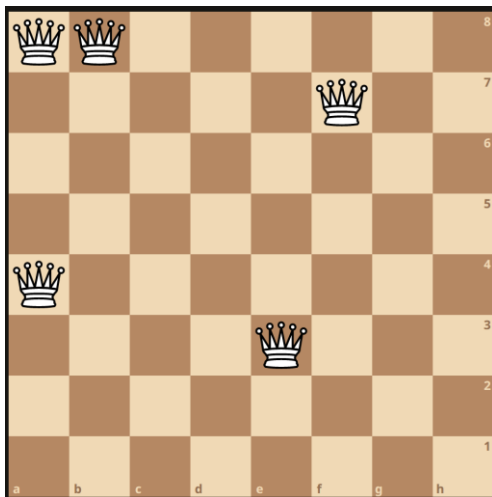
Pencarian solusi dilanjutkan dengan metode exhaustive search atau bisa juga dipanggil brute force search. Sama halnya dengan subbab sebelumnya, akan diiterasi semua kemungkinan yang mungkin untuk $k = 5$ buah ratu pada papan catur, dan akan di cek apakah k_1, k_2, k_3, k_4 , dan k_5 sudah membentuk suatu dominating set atau belum

Dengan bentuk program yang kurang lebih sama dengan pseudocode sebelumnya dengan perbedaan jumlah perulangan for yang digunakan, didapat hasil sebagai berikut

dominating set = {0, 1, 13, 32, 44}

iteration = 17736

Dengan visualisasi sebagai berikut



Gambar 3 Visualisasi Hasil Brute Force

(Sumber: https://lichess.org/editor/QQ6/5Q2/8/8/Q7/4Q3/8/8_w_--_0_1?color=white)

Karena sudah dibuktikan bahwa lower bound dari $d(8)$ adalah 5, sementara sudah didapat bahwa terdapat solusi untuk $k = 5$ maka sudah dibuktikan bahwa

$d(8) = 5$

Dengan kata lain jumlah ratu paling sedikit yang bisa diletakkan pada papan catur berukuran $n \times n$ sehingga dapat dipastikan setiap petak pada papan catur diserang oleh setidaknya satu ratu adalah sebanyak 5 ratu.

D. Pendekatan Greedy Best First

Terlihat bahwa masih terdapat sebanyak 17736 iterasi yang harus dilakukan untuk menemukan salah satu solusinya, namun akan dilakukan pendekatan dengan greedy best first untuk menemukan solusi dengan jumlah iterasi yang jauh lebih sedikit dari 17736.

1) Fungsi Heuristik

Tentu dengan pencarian greedy best first, diperlukan suatu fungsi heuristik yang memberi sinyal bahwa solusi node yang di ekspansi sudah mendekati solusi dengan paling cepat.

Pada kasus persoalan ini, fungsi heuristik yang dipakai adalah count_not_covered, yaitu menghitung banyak petak yang belum diserang oleh ratu manapun.

Berikut merupakan contoh pseudocode untuk fungsi fungsi heuristik count_not_covered

```
count ← 0
FOR i = 0 to 7
  FOR j = 0 to 7
    IF board(i, j) == 0
      count ← count
    ENDIF
  ENDFOR
ENDFOR
→ count
```

Dengan menerapkan fungsi heuristik ini, maka bisa dipastikan bahwa penempatan ratu sudah optimal pada suatu instans di node tertentu.

2) Fungsi Pembatas

Karena kita sudah mengetahui bahwa $d(8)$ adalah 5, maka apabila jumlah ratu yang ada pada suatu node sudah lebih lebih dari sama dengan 5, tetapi papan catur belum covered, maka node tersebut bisa langsung dipangkas

3) Priority Queue

Pada program utama algoritma greedy best first akan memanfaatkan Priority Queue untuk mensimulasikan pemilihan nilai fungsi heuristik yang lebih rendah terlebih dahulu. Sehingga priority queue akan mengatur nilai yang paling rendah akan diletakkan di tail untuk di pop.

4) Algoritma Utama Greedy Best First

Program utama memiliki pseudocode sebagai berikut

```
iteration ← 0
board ← empty_board
// initialize first 64 node
FOR i = 0 to 63
```

(Sumber: https://lichess.org/editor/1Q6/6Q1/8/3Q4/2Q5/4Q3/8/8_w_-_-_0_1?color=white)

Namun hal yang paling menarik dari hasil ini adalah jumlah iterasi yang diperlukan untuk mencapai solusi tersebut, yaitu hanyalah sebanyak 252 iterasi. Hal ini jauh berbeda dengan hasil jumlah iterasi yang dilakukan algoritma brute force.

E. Kasus pencarian semua solusi

Apabila yang diinginkan adalah jumlah semua solusi dominating set yang minimum dari $n = 8$, kedua pendekatan tersebut masih bisa dilakukan, tentunya dengan sedikit modifikasi pada algoritma

1) Pencarian Brute Force

Pada pendekatan secara brute force, bagian yang tadinya langsung mengembalikan nilai dominating set jika sudah ketemu, bisa diganti dengan menambahkan suatu variabel yang contoh bernama `num_of_solution`.

Dengan pencarian brute force didapat solusi sebanyak 4860 solusi.

2) Pencarian Greedy Best First

Dengan pencarian Greedy Best First perlu dilakukan modifikasi dari algoritma awal, yaitu penghidupan node yang tadinya dalam bentuk for loop

FOR $i = 0$ to 63

Diganti sehingga node yang dihidupkan hanyalah node yang memiliki tempat yang lebih 'depan' dibanding posisi ratu yang paling depan pada saat itu. Hal ini dilakukan agar tidak terdapat duplikat solusi pada dominating set, dimana isinya sama tetapi urutannya berbeda.

Hasil dari pencarian greedy best first sama dengan pencarian brute force, yaitu ditemukan sebanyak 4860 solusi. Namun hal yang menarik adalah dibutuhkan waktu setidaknya 3 kali lipat yang dibutuhkan brute force untuk pencarian greedy best first bisa selesai.

V. KESIMPULAN

Persoalan dominasi ratu bisa diselesaikan menggunakan algoritma brute force maupun greedy best first dengan instans $n = 8$. Didapat bahwa $d(n)$ atau jumlah elemen dari dominating set minimum adalah 5. Yaitu dibutuhkan minimal 5 buah ratu untuk dapat menutup semua petak di papan catur 8×8 .

Iterasi yang dibutuhkan untuk algoritma greedy best first jauh lebih sedikit dibandingkan dengan iterasi pada algoritma brute force. Hal ini terjadi karena algoritma greedy best first menggunakan heuristik yang mempercepat konvergensi ke solusi yang diinginkan

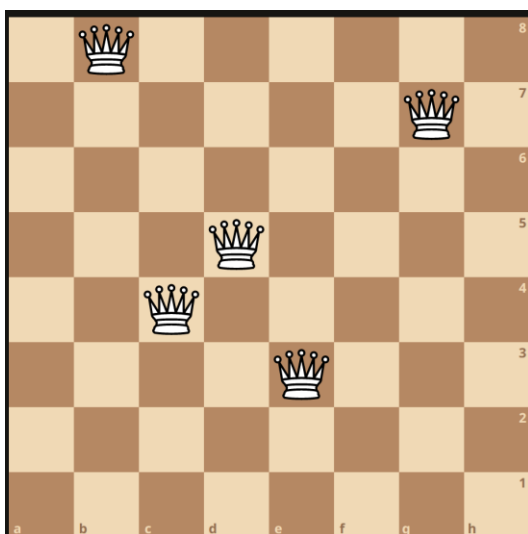
Namun pada pencarian semua solusi yang mungkin, algoritma greedy best first kalah cepat 3 kali lipat dengan algoritma brute force. Hal ini dicurigai karena harga pemeliharaan priority queue lumayan mahal dan memakan waktu yang lama untuk jumlah elemen priority queue yang besar.

```
iteration ← 1
put queen i
score ← get_heuristic()
push (score, queen_pos) to priorityQueue
ENDFOR
WHILE priorityQueue is NOT EMPTY DO
  score, queen_pos ← priorityQueue.pop()
  \\ loop to check all child node
  FOR i = - to 63
    IF i NOT IN queen_pos
      iteration ← iteration + 1
      new_queen_pos.append(i)
      score ← get_heuristic
      IF score == 0
        // solution found
        print(new_queen_pos)
        print(iteration)
      ENDIF
    ENDIF
  ENDFOR
ENDWHILE
print("Solution not Found")
```

Apabila dijalankan program tersebut, maka akan didapatkan

dominating set = {27, 14, 34, 44, 1}

Dengan visualisasi papan catur sebagai berikut



Gambar 4 Visualisasi Hasil Greedy Best First

VI. SARAN

Masih terdapat beberapa optimasi yang bisa dilakukan dalam pencarian seluruh solusi dari persoalan yang ada. Salah satunya adalah dengan memperhatikan simetri putar atau lipat yang muncul pada tiap solusi. Dengan memperhatikan simetri yang ada, jumlah kasus yang perlu dicek akan menurun drastis.

VII. APPENDIX

Link github: <https://github.com/Fakhrimm/Persoalan-Dominasi-Ratu>

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan puji syukur kepada Tuhan Yang Maha Esa, karena dengan rahmat Nya penulis bisa menyelesaikan makalah berjudul “Penerapan Algoritma Brute Force dan Greedy Best First Terhadap Persoalan Dominasi Ratu untuk N sama dengan Delapan” tepat waktu. Terima kasih kepada dosen mata kuliah IF2211 Tahun 2022/2023, Dr. Nur Ulfa Maulidevi, S. T, M. Sc., Dr. Ir. Rinaldi Munir, M. T., dan Dr. Rila Mandala. yang telah membimbing penulis selama berkuliah di mata kuliah Strategi Algoritma ini. Penulis juga ingin mengucapkan terima kasih kepada semua pihak yang terlibat. Terakhir, penulis ingin mengucapkan terima kasih kepada semua sumber yang telah dijadikan referensi pada makalah ini.

REFERENCES

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

- [1] freecodecamp, “Brute Force Algorithms Explained,” 6 Januari 2020. [Online]. Available: <https://www.freecodecamp.org/news/brute-force-algorithms-explained/>. [Diakses 22 Mei 2023].

- [2] C. Yang, “Greedy Best-First Search,” codecademy, 22 April 2023. [Online]. Available: <https://www.codecademy.com/resources/docs/ai/search-algorithms/greedy-best-first-search>. [Diakses 23 Mei 2023].
- [3] Munir, Rinaldi. 2023. “Pengantar Strategi Algoritma (Versi baru 2023)”. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/stima22-23.htm> [Diakses 22 Mei 2023].
- [4] Munir, Rinaldi. 2023. “Algoritma Brute Force (Bagian 1)”. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/stima22-23.htm> [Diakses 22 Mei 2023].
- [5] Munir, Rinaldi. 2023. “Algoritma Brute Force (Bagian 2)”. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/stima22-23.htm> [Diakses 22 Mei 2023].
- [6] Munir, Rinaldi. 2023. “Penentuan rute (Route/Path Planning) - Bagian 1”. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/stima22-23.htm> [Diakses 22 Mei 2023].
- [7] Munir, Rinaldi. 2023. “Penentuan rute (Route/Path Planning) - Bagian 2”. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/stima22-23.htm> [Diakses 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



Fakhri Muhammad Mahendra