

Aplikasi Algoritma Runut-Balik untuk Mencari *Autobiographical Number*

Jeane Mikha Erwansyah - 13519116
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13519116@std.stei.itb.ac.id

Abstrak—*Autobiographical number* merupakan bilangan bulat asli dengan n angka dalam basis n yang angka di posisi p ($0, 1, 2, \dots, n$) menyatakan jumlah kemunculan angka p pada bilangan tersebut. Algoritma runut-balik (*backtrack*) dapat digunakan untuk mencari bilangan n -angka yang memenuhi syarat-syarat sebuah bilangan untuk menjadi *autobiographical number* dengan basis n .

Kata kunci—algoritma; *autobiographical number*; basis; runut-balik;

I. PENDAHULUAN

Teka-teki merupakan sebuah soal berupa kalimat yang bertujuan untuk mengasah pikiran. Salah satu teka-teki mengenai matematika atau bilangan adalah *autobiographical number*. *Autobiographical number* merupakan bilangan yang memiliki sejumlah angka yang angka-angkanya menyatakan jumlah kemunculan angka posisi ke- i pada bilangan tersebut. Algoritma dapat digunakan untuk memudahkan pencarian bilangan tersebut.

Algoritma merupakan urutan perintah jelas untuk menyelesaikan sebuah masalah. Sebuah algoritma harus memiliki langkah yang jelas dan tidak ambigu. Sebuah masalah yang ada tidak terbatas diselesaikan oleh satu algoritma namun dapat juga diselesaikan oleh algoritma lain atau algoritma yang sama tetapi dalam representasi yang berbeda [1].

Ada berbagai macam strategi untuk mencari *autobiographical number* seperti algoritma *bruteforce*, algoritma *backtracking* atau runut-balik, algoritma *branch and bound*, dan lain-lain. Makalah ini membahas aplikasi algoritma runut-balik untuk mencari n -angka *autobiographical number* dalam basis n dan hasil aplikasi dalam bentuk program.

II. LANDASAN TEORI

A. *Autobiographical Number*

Autobiographical number atau *self-descriptive number* merupakan bilangan yang memenuhi syarat untuk angka ke- i ($0, 1, 2, \dots, n-1$) dari sebuah bilangan dengan jumlah angka n dalam basis n menyatakan jumlah angka i dalam bilangan tersebut [2][3]. Dari syarat tersebut dapat diperoleh syarat lain yaitu jumlah angka-angkanya sama dengan jumlah angka (n) dalam basis n .

Contoh bilangan *autobiographical* dengan 4 angka dalam basis 4 adalah 1210 [4]. Bilangan 1210 merupakan *autobiographical number* sebab angka pertama yaitu 1 menyatakan jumlah angka 0, angka kedua yaitu 2 menyatakan jumlah angka 1, angka ketiga yaitu 1 menyatakan jumlah angka 2, dan angka terakhir yaitu 0 menyatakan jumlah angka 3. Selain contoh tersebut ada juga bilangan *autobiographical* lain untuk bilangan 4 angka yaitu 2020.



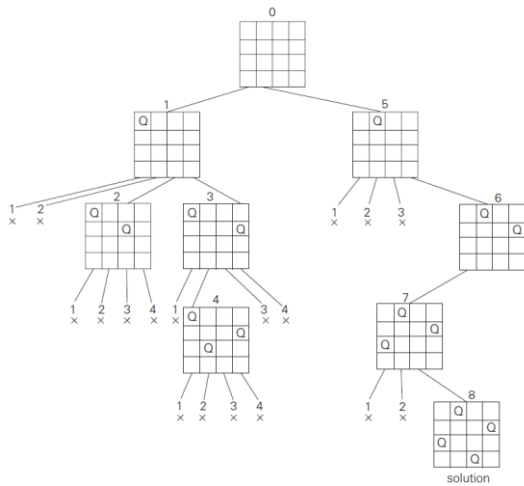
Gambar 1. Contoh *autobiographical number* (sumber: <https://brilliant.org/practice/autobiographical-numbers/>)

B. Algoritma Runut-Balik

Algoritma runut-balik memiliki prinsip yaitu membuat sebuah solusi perkomponen dan mengecek apakah seluruh komponen solusi yang telah dibuat memenuhi syarat. Jika syarat persoalan tidak dipenuhi, algoritma ini akan menggantikan kandidat komponen berikutnya dengan kandidat lain. Jika tidak ada kandidat yang tersedia, algoritma ini akan melakukan runut-balik ke komponen solusi terakhir dan mengganti komponen tersebut dengan kandidat yang lain [1].

Proses pemilihan komponen solusi dari algoritma ini dapat divisualisasikan dengan membuat pohon ruang status. Akar dari pohon ruang status ini merepresentasikan status awal sebelum komponen pertama dipilih. Simpul aras pertama pada pohon ini merepresentasikan kandidat komponen pertama dari solusi, simpul aras kedua pada pohon ini merepresentasikan kandidat komponen kedua dari solusi, dan seterusnya hingga aras terakhir merepresentasikan solusi yang utuh yang mungkin memenuhi syarat persoalan. Sebuah simpul pada pohon ruang status disebut menjanjikan menghasilkan solusi bila solusi yang sudah terbuat sejauh itu masih memungkinkan untuk menghasilkan solusi utuh dan sebaliknya simpul disebut tidak menjanjikan jika simpul tersebut tidak memungkinkan mengarah ke solusi utuh. Simpul yang tidak menjanjikan tersebut diterminasi oleh algoritma dan tidak akan dilanjutkan [1].

Algoritma ini dapat mencari hanya satu atau lebih dari satu solusi jika diinginkan. Oleh karena proses kerja algoritma runut-balik ini, algoritma ini mirip dengan proses kerja algoritma *Depth-First Search* (DFS) yaitu *depth first* [1].



Gambar 2. Contoh pohon ruang status untuk *N-Queen Problem* dengan $N = 4$ (sumber: Introduction to the Design and Analysis of Algorithms, 3rd ed.)

C. Properti Umum Algoritma Runut-Balik

Properti umum pertama adalah solusi persoalan. Solusi dinyatakan sebagai vektor dengan *n-tuple*:

$$X = (x_1, x_2, \dots, x_n), x_i \in S_i \quad (1)$$

Properti kedua adalah fungsi pembangkit nilai x_k yang dinyatakan sebagai predikat $T(x[1], x[2], \dots, x[k-1])$ yang membangkitkan nilai untuk x_k , yang merupakan komponen vektor solusi.

Properti terakhir adalah fungsi pembatas yang dinyatakan sebagai predikat $B(x_1, x_2, \dots, x_k)$. $B()$ bernilai *true* jika mengarah ke solusi dan sebaliknya jika tidak mengarah ke solusi maka akan bernilai *false*. Jika $B()$ bernilai *true*, akan dilakukan pembangkitan nilai untuk x_{k+1} dilanjutkan. Jika $B()$ bernilai *false*, nilai (x_1, x_2, \dots, x_k) dibuang atau diterminasi dan tidak dilanjutkan [5].

III. APLIKASI ALGORITMA RUNUT-BALIK UNTUK Mencari N-ANGKA *AUTOBIOGRAPHICAL NUMBER* DALAM BASIS N

Berikut adalah metode pencarian bilangan *autobiographical* dengan menggunakan algoritma runut-balik.

A. Mapping Persoalan *Autobiographical Number* ke dalam Algoritma Runut-Balik

Solusi dari persoalan ini merupakan vektor *n-tuple* yaitu

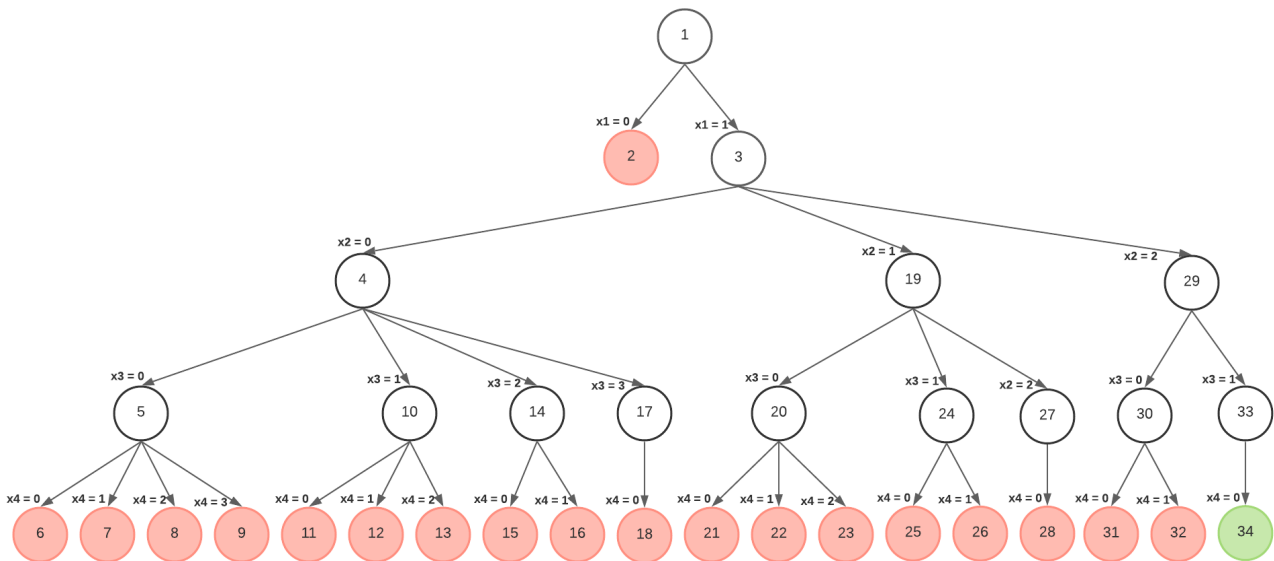
$$X = (x_1, x_2, \dots, x_n), x_i \in \{0, \dots, n-1\} \quad (2)$$

Fungsi pembatas untuk persoalan ini adalah mengecek apakah bilangan mulai dari 0, mengecek panjang dari bilangan yang telah terbentuk, mengecek jumlah angka bukan nol yang ada pada bilangan apakah kurang dari sama dengan basis n atau tidak (lihat persamaan (3)), dan mengecek apakah bilangan *autobiographical* jika panjang sama dengan basis n .

$$\sum_{i=0}^k x_i \leq n \quad (3)$$

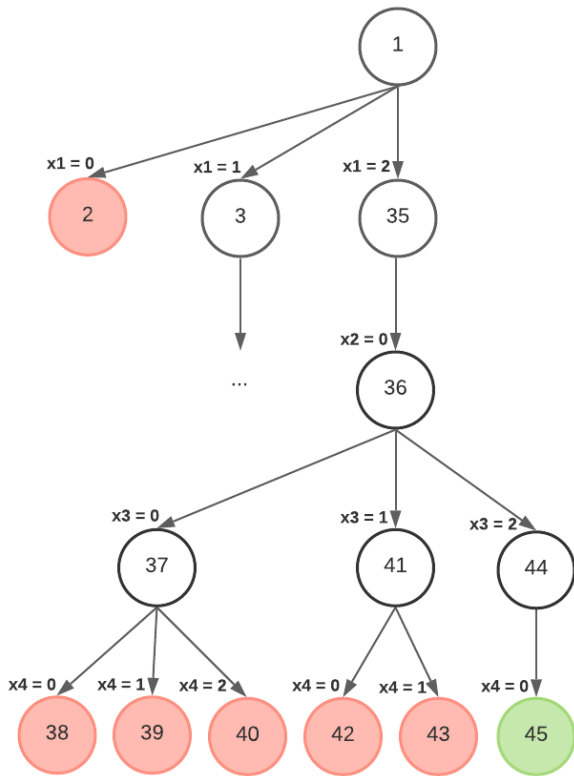
B. Pencarian dengan Algoritma Runut-Balik

Berikut adalah proses algoritma runut-balik untuk bilangan *autobiographical* dengan 4 angka dan basis 4.



Gambar 3. Pohon ruang status *autobiographical number* 4 angka dalam basis 4 (sumber: Dokumen pribadi)

Menurut [4], khusus untuk *autobiographical number* dengan 4 angka (dalam basis 4) ada solusi lain (lihat gambar 4).



Gambar 4. Lanjutan pohon ruang status *autobiographical number* 4 angka dalam basis 4 (sumber: Dokumen pribadi)

Gambar 3 dan Gambar 4 menggambarkan pohon ruang status hasil dari pencarian *autobiographical number* dengan 4 angka dan dalam basis 4. Dapat dilihat dari hasil pencarian bahwa hasil yang diperoleh adalah bilangan 1210 dan 2020 yang ditandai oleh simpul berwarna hijau. Pencarian dilakukan dengan memilih angka untuk angka pertama hingga angka keempat ditandai secara berurutan oleh aras 1 hingga aras 4. Angka pada simpul menandakan urutan pengunjungan simpul. Pohon ruang status tersebut dikunjungi secara *depth first*. Simpul 1 merupakan simpul awal sebelum pencarian dilakukan.

Simpul yang berwarna merah adalah simpul yang diterminasi karena tidak memenuhi syarat *autobiographical number* atau seperti simpul 2 tidak memenuhi syarat. Simpul 2 diterminasi karena bilangan tidak dimulai dengan angka 0.

Setiap simpul pada pohon ruang status tersebut memiliki maksimal empat simpul anak karena basis angka yang digunakan pada pencarian ini adalah basis 4. Simpul yang terletak pada aras ketiga dapat dilihat tidak semuanya memiliki 4 simpul anak karena melanggar fungsi pembatas sebab jumlah angka-angka bilangan tersebut akan melebihi jumlah angka bilangan.

Bilangan yang terbentuk oleh lintasan dari akar ke simpul berwarna hijau (simpul 34 dan simpul 45) diterima sebab memenuhi syarat dan tidak melanggar fungsi pembatas.

Bilangan 2020 merupakan *autobiographical number* sebab angka pertama yaitu 2 menyatakan jumlah angka 0, angka kedua yaitu 0 menyatakan jumlah angka 1, angka ketiga yaitu 2 menyatakan jumlah angka 2, dan angka terakhir yaitu 0 menyatakan jumlah angka 3.

IV. HASIL APLIKASI DALAM BENTUK PROGRAM

Selanjutnya, untuk memudahkan pencarian *autobiographical number* dibuatlah sebuah kode program dalam bahasa Python. Berikut adalah potongan kode bagian runut-balik.

```

1 def findAutobioNum(base, k):
2     num = list()
3     i = 0
4
5     while True:
6         while (i < base):
7             # digit terakhir
8             if base == 8:
9                 new = '{:0}'.format(int(oct(i), \
10                                     base=base))
11
12             elif base == 16:
13                 new = '{:X}'.format(int(hex(i), \
14                                     base=base))
15
16             else:
17                 new = i
18
19             temp = list(num)
20             if len(temp) == base:
21                 temp.pop()
22                 temp.append(str(new))
23
24             if isPossible(''.join(temp), base):
25                 if len(num) == base:
26                     num.pop()
27                     num.append(str(new))
28
29                 if isAutobiographical(num, base):
30                     return ''.join(num)
31                 else:
32                     if len(num) == base:
33                         if int(num[-1], base=base) == base-1:
34                             i = base
35                         else:
36                             i = int(num[-1], base=base) + 1
37                     else:
38                         i = 0
39
40             else:
41                 i += 1
42
43     num = num[:-1]
44     num = list(increment(''.join(num), base))
45     i = 0

```

Gambar 5. Kode program bagian runut-balik (sumber: <https://github.com/jerwansyah/autobiographical-number/>)

Sepotong kode program (lihat Gambar 6) tersebut menampilkan implementasi algoritma runut-balik yang dilakukan. Karena pada setiap basis ($n > 3$) memiliki setidaknya 1 *autobiographical number* kecuali bilangan dengan basis 6 [4], pencarian dapat dilakukan dalam sebuah *loop* yang hanya akan berhenti jika menemukan bilangan yang memenuhi syarat. Cara kerja dari *loop* ini adalah secara *depth first*. Pada Gambar 6 dapat dilihat pencarian secara *depth first*. Hasil cetakan atau luaran dari program ini merupakan bilangan yang masih memenuhi fungsi pembatas. Fungsi pembatas dalam kode adalah fungsi `isPossible()`. Fungsi tersebut mengecek apakah bilangan memenuhi syarat: (1) bilangan tidak dimulai dengan angka 0; (2) panjang bilangan kurang dari sama dengan basis; dan (3) jumlah angka-angka bilangan kurang dari sama dengan basis.

Cara pembangkitan nilai dalam program ini adalah dengan melakukan *while loop* dalam *loop* utama. Pertama kode akan melakukan pembangunan bilangan dengan cara *depth first* setelah itu karena panjang sudah sama dengan basis maka kode melakukan penambahan nilai untuk angka terakhir hingga sama dengan nilai maksimum untuk basis n . Nilai n merupakan hasil dari masukan pengguna dan pilihan masih terbatas antara 8, 10, atau 16. Jika pembangkitan nilai untuk angka terakhir telah mencapai batas maksimum, akan dilakukan penambahan nilai untuk angka sebelum angka terakhir dan angka terakhir dibuang. Langkah tersebut merupakan bagian dari runut-balik.

Gambar 7 menampilkan hasil cetakan dari langkah-langkah terakhir kode program sebelum mendapatkan bilangan *autobiographical*. Kode program tidak melakukan penambahan nilai (*increment*) untuk angka 1 pada bilangan 6210000001 karena total digit-digit sudah sama dengan 10 sehingga program hanya ‘memindahkan’ posisi angka 1.

```
Masukan basis (8/10/16): 10
1
10
100
1000
10000
100000
1000000
10000000
100000000
1000000000
10000000001
10000000002
10000000003
10000000004
10000000005
10000000006
10000000007
10000000008
```

Gambar 6. Cetakan langkah awal mencari *autobiographical number* dengan 10 angka dalam basis 10 (sumber: Dokumen pribadi)

```
6210
62100
621000
6210000
62100000
621000000
6210000000
62100000000
62100000001
62100000010
62100000100
62100001000
621000010000
6210000100000
62100001000000
```

Gambar 7. Cetakan langkah akhir mencari *autobiographical number* dengan 10 angka dalam basis 10 (sumber: Dokumen pribadi)

Tiga hasil percobaan yang dilakukan menggunakan kode program hanya diperoleh dua bilangan yaitu bilangan 42101000 untuk basis 8 dan 6210001000 untuk basis 10. Bilangan *autobiographical* untuk basis 16 tidak diperoleh karena keterbatasan algoritma dan fungsi pembatas sehingga menyebabkan kompleksitas waktu program meningkat secara eksponensial. Algoritma runut-balik kurang cocok untuk diaplikasikan untuk mencari *autobiographical number* dengan jumlah angka dan dalam basis yang besar. Dibandingkan dengan lama waktu eksekusi program mencari bilangan *autobiographical* berbasis 8 dan 10, pencarian *autobiographical number* berbasis 16 sangatlah besar.

Dari hasil percobaan dan menurut [4] terlihat ada sebuah pola muncul yaitu bilangan *autobiographical* memiliki 4 angka yang bukan nol yaitu, satu angka 2, dua angka 1, dan 1 angka $n - 4$. Angka $n - 4$ menandakan jumlah angka nol dan angka n dikurang 4 karena 4 merupakan jumlah angka yang bukan nol.

Karena mencari *autobiographical number* dengan 16 angka dalam basis 16 menggunakan program sangat lama, bilangan dapat diperoleh dengan pola yang muncul dan dari [4] sehingga diperoleh bilangan C21000000001000. Dibutuhkan sekitar $1,8 \times 10^{17}$ langkah untuk mendapatkan bilangan tersebut. Jumlah langkah tersebut sangat besar dibandingkan dengan langkah pencarian *autobiographical number* dengan 10 angka dalam basis 10 yang hanya membutuhkan kurang lebih 6 juta langkah.

```
kuliah/stima/autobiographical-number via 🐍 v3.8.5 took 15s
> ./autobionum.py
Masukan basis (8/10/16): 10
Autobiographical number dengan 10-angka dalam basis 10 : 6210001000
Waktu eksekusi : 0:00:03.328490
```

Gambar 8. Hasil pencarian *autobiographical number* dengan 10 angka dalam basis 10 (sumber: Dokumen pribadi)

```
kuliah/stima/autobiographical-number via 🐍 v3.8.5 took 4s
> ./autobionum.py
Masukan basis (8/10/16): 8
Autobiographical number dengan 8-angka dalam basis 8 : 42101000
Waktu eksekusi : 0:00:00.183299
```

Gambar 9. Hasil pencarian *autobiographical number* dengan 8 angka dalam basis 8 (sumber: Dokumen pribadi)

```
kuliah/stima/autobiographical-number via 🐍 v3.8.5 took 4s
> ./autobionum.py
Masukan basis (8/10/16): 16
^CTraceback (most recent call last):
  File "./autobionum.py", line 92, in <module>
    sol = findAutobioNum(basis, 1)
  File "./autobionum.py", line 15, in findAutobioNum
    new = '{:X}'.format(int(hex(i), base=base))
KeyboardInterrupt

kuliah/stima/autobiographical-number via 🐍 v3.8.5 took 59m34s
>
```

Gambar 10. Hasil pencarian *autobiographical number* dengan 16 angka dalam basis 16 (sumber: Dokumen pribadi)

V. KESIMPULAN

Aplikasi algoritma runut-balik dapat digunakan untuk mencari *autobiographical number* dengan angka n dan dalam basis n . Solusi dari persoalan ini adalah

$$X = (x_1, x_2, \dots, x_n), x_i \in \{0, \dots, n-1\}$$

dengan fungsi pembatas

$$\sum_{i=0}^k x_i \leq n$$

Pencarian *autobiographical number* semata terlihat susah namun dengan bantuan algoritma runut-balik dapat diperoleh dengan mudah. Dari hasil tiga percobaan dengan menggunakan kode program dapat dilihat terbentuk sebuah pola yaitu bilangan memiliki $n - 4$ angka 0, 4 lainnya menandakan angka 2, 1, 1, dan angka pertama yang menyatakan jumlah angka 0. Ada pola tersebut sebenarnya mempermudah pencarian *autobiographical number*. Aplikasi algoritma runut-balik menjadi kurang efisien waktu dan ruang jika basis yang digunakan besar.

VI. SARAN

Saran untuk pembaca adalah untuk mengembangkan kode program agar dapat mencari *autobiographical number* dalam basis lain tidak terbatas hanya 8, 10, atau 16. Selain itu, pembaca dapat mencoba-coba aplikasi dengan menggunakan algoritma lain sehingga pencarian *autobiographical number* dengan basis yang besar dapat dilakukan dengan lebih cepat.

PRANALA KODE DI GITHUB

<https://github.com/jerwansyah/autobiographical-number/>

APPENDIKS

TABLE I. AUTOBIOGRAPHICAL NUMBER (BASIS 4-10)

Basis	<i>Autobiographical number</i>
4	1210, 2020
5	21200
6	-
7	3211000
8	42101000
9	521001000

Basis	<i>Autobiographical number</i>
10	6210001000
11	72100001000
12	821000001000
13	9210000001000
14	A2100000001000
15	B21000000001000
16	C210000000001000

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, kepada orang tua penulis, dan teman penulis yang telah mendukung penulis dalam proses pembuatan makalah ini, serta kepada semua dosen pengampu mata kuliah IF2211 Strategi Algoritma, terutama Prof. Ir. Dwi Hendratmo Widyantoro, M.Sc., Ph.D. atas pemberian ajaran dan bimbingannya selama kuliah.

RUJUKAN

- [1] A. Levitin, Introduction to the Design and Analysis of Algorithms, 3rd ed., New Jersey: Pearson, 2012, p. 3, 423-430

- [2] E. W. Weisstein, "Self-Descriptive Number," MathWorld--A Wolfram Web Resource, <https://mathworld.wolfram.com/Self-DescriptiveNumber.html>
- [3] T. Khovanova, "Autobiographical Number," dalam A Story of Storytelling Numbers dalam Math Horizons, v.17, n.1, 14-17, 2009
- [4] N. J. A Sloane., (ed.). Sequence A138480 (Autobiographical numbers), The On-Line Encyclopedia of Integer Sequences, OEIS Foundation, <https://oeis.org/A138480>
- [5] R. Munir, Algoritma Runut-balik (Backtracking), <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-backtracking-2021-Bagian1.pdf>

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.

Jakarta, 11 Mei 2021



Jeane Mikha Erwansyah – 13519116