ISSN: 2614-6754 (print)
ISSN: 2614-3097(online)

Halaman 9009-9016
Volume 9 Nomor 1 Tahun 2025

Implementasi Python dalam Identifikasi Subgrup dan Nonsubgrup pada Grup

Septi Agita Tarigan[1], Sri Lestari Manurung[2], Raysah Puteri Sulaiman[3], Yonata Hutapea[4]
[1,2,3,4] Pendidikan Matematika, Universitas Negeri Medan
e-mail: Septitarigan28@gmail.com

Abstrak

Identifikasi subgrup dalam struktur aljabar merupakan aspek penting dalam teori grup, namun prosesnya sering kali memerlukan perhitungan yang kompleks. Penelitian ini bertujuan untuk menilai kepraktisan penggunaan Python dalam analisis struktur aljabar, khususnya dalam identifikasi subgrup. Dengan menggunakan metode studi kasus dan eksperimen, program Python dikembangkan untuk menguji apakah suatu himpunan memenuhi syarat sebagai subgrup, yaitu tertutup terhadap operasi, memiliki elemen identitas, dan memuat invers dari setiap elemennya. Perbandingan dilakukan antara hasil analisis manual dan hasil yang diperoleh melalui program Python guna menilai kepraktisan komputasi. Hasil penelitian menunjukkan bahwa program Python yang dikembangkan mampu mengidentifikasi subgrup secara akurat dan efisien. Kesimpulannya, implementasi Python memberikan solusi yang lebih sistematis dalam menentukan subgrup dari suatu grup, serta berpotensi menjadi alat bantu dalam pembelajaran dan penelitian di bidang aljabar abstrak.

Kata kunci: *Python, Struktur Aljabar, Subgrup*

Abstract

Identifying subgroups in algebraic structures is an essential aspect of group theory, but the process often involves complex calculations. This study aimed to assess the practicality of using Python for analyzing algebraic structures, particularly in subgroup identification. A Python program was developed using case study and experimental methods to determine whether a given set forms a group and to directly generate its subgroups. Instead of explicitly verifying the group axioms, the program provides an automated approach to identifying subgroups. A comparison was made between manual analysis and the program's output to evaluate computational efficiency. The results show that the Python program accurately and efficiently identifies subgroups. These findings suggest that Python offers a more systematic approach to subgroup determination and has the potential to serve as a valuable tool for learning and research in abstract algebra.

Keywords: *Python, Algebraic Structures, Subgroup*

PENDAHULUAN

Pada matakuliah struktur aljabar, konsep grup memiliki peran yang mendasar dalam mempelajari berbagai sistem aljabar. Grup merupakan salah satu konsep dasar dalam struktur aljabar, yang terdiri dari himpunan elemen dengan operasi yang memenuhi aksioma tertentu seperti komutatif, asosiatif, memiliki elemen identitas, dan elemen invers (Isa, 2024). Dalam kajian grup, salah satu aspek yang menarik adalah pembentukan subgrup, yakni himpunan bagian dari grup yang juga memenuhi aksioma-aksioma grup (Naibaho dkk., 2025). Tantangan utama dalam memahami konsep grup adalah proses pembuktian apakah suatu subgrup merupakan bagian dari grup atau bukan, oleh karena itu diperlukan pemahaman mendalam tentang sifat-sifat grup dan struktur aljabar (Faizah, 2019).

Penentuan semua subgrup dari suatu grup dapat ditentukan dengan mudah melalui bantuan pemograman Python (Muktyas, 2018). Bahasa pemrograman Python adalah salah satu alat bantu yang ditawarkan teknologi untuk mempermudah proses pembelajaran (Lutz, 2009),

terkhususnya konsep grup dalam matakuliah struktur aljabar. Python memiliki keunggulan dalam hal sintaks yang sederhana dan mudah dipahami (Ritonga & Yahfizham, 2023), sehingga memudahkan mahasiswa untuk mengimplementasikan berbagai konsep abstrak dalam aljabar, seperti grup, subgrup, dan lainnya.

Penerapan bahasa pemrograman dalam pembelajaran matematika dapat meningkatkan pemahaman konsep secara lebih mendalam (Simarmata dkk., 2022). Selain itu, Python, sebagai bahasa pemrograman yang populer di dunia akademik, memungkinkan dilakukannya eksplorasi struktur aljabar melalui visualisasi dan simulasi, yang membuat pembelajaran lebih interaktif dan efisien (Das dkk., 2024).

Di Indonesia, penerapan bahasa pemrograman Python pada mata kuliah struktur aljabar belum banyak digunakan. Oleh karena itu, penelitian ini bertujuan untuk mengeksplorasi bagaimana Python dapat digunakan secara efektif dalam mata kuliah Struktur Aljabar untuk membuktikan apakah suatu subgrup merupakan bagian dari grup atau bukan. Penelitian ini juga diharapkan dapat memberikan kontribusi dalam pengembangan metode pembelajaran yang lebih inovatif dan interaktif dalam bidang matematika.

METODE

Penelitian ini menggunakan metode studi kasus dan eksperimen. Studi kasus merupakan pendekatan yang bertujuan untuk memahami suatu fenomena secara mendalam melalui pengumpulan dan analisis data dari berbagai sumber (Cresswell dalam Assyakurrohim, dkk: 2023). Dalam konteks penelitian ini, studi kasus dilakukan untuk memperoleh informasi mengenai penggunaan bahasa pemrograman Python dalam bidang matematika dengan merujuk pada berbagai penelitian terdahulu.

Sementara itu, metode eksperimen digunakan untuk menguji penerapan Python dalam menentukan subgrup dari suatu grup dalam struktur aljabar. Eksperimen ini melibatkan serangkaian uji coba yang dilakukan untuk mengevaluasi bagaimana Python dapat digunakan dalam proses tersebut. Setelah eksperimen selesai, hasil perhitungan yang diperoleh dengan Python akan dibandingkan dengan hasil yang diperoleh melalui metode konvensional atau manual. Perbandingan ini bertujuan untuk menilai kepraktisan penggunaan Python dalam analisis struktur aljabar, khususnya dalam identifikasi subgrup. Dengan demikian, penelitian ini tidak hanya mengeksplorasi penggunaan Python dalam matematika secara teoritis, tetapi juga menguji aplikasinya secara praktis.

HASIL DAN PEMBAHASAN

Penyelesaian Manual

Pembelajaran konsep grup, khususnya pada mata kuliah Struktur Aljabar, sering kali meminta mahasiswa untuk menentukan apakah suatu himpunan dengan operasi tertentu membentuk subgrup dari suatu grup atau tidak. Dalam menentukan suatu subgrup, mahasiswa perlu memahami tabel Cayley serta empat aksioma grup, yaitu sifat tertutup, sifat asosiatif, unsur identitas, dan unsur invers. Metode manual seperti ini tidak hanya memakan banyak waktu tetapi juga rentan terhadap kesalahan, yang dapat menjadi tantangan bagi mahasiswa. Sebagai contoh, mahasiswa diminta untuk menentukan subgrup dari Z_{30}. Adapun penyelesaian yang akan dilakukan mahasiswa adalah sebagai berikut:

Penyelesaian :

$Z_{30} = \{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30\}$

Kita ambil himpunan bagian dari Z_{30} yaitu:

a) $H_1 = \{0,15\}$

Pembuktian:

Tabel 1. Cayley 0, 15

$+_{30}$	**0**	**15**
0	0	15
15	15	0

Dengan memperhatikan tabel, maka:

- Aksioma I (sifat ketertutupan dan ketunggalan)
  $a * b \in H_1, \forall\, a, b \in H_1$ (**terpenuhi**)
- Aksioma II (sifat asosiatif)
  $a * (b * c) = (a * b) * c, \forall\, a, b, c \in H_1$(**terpenuhi**)
- Aksioma III (elemen identitas)
  Akan ditunjukkan $\exists\, e \in H_1 \ni \forall\, a \in H_1$berlaku $a * e = e * a = a$
  Pilih $e = 0 \in H_1$, ambil sembarang $a \in G$ maka
  $a * e = a + 0 = a$ dan $e * a = 0 + a = a$ sehingga
  $a * e = e * a = a$ (**terpenuhi**)
- Aksioma IV (invers)
  Akan ditunjukkan $\forall\, a \in H_1 \exists\, a^{-1} \in H_1 \ni a * a^{-1} = a^{-1} * a = e$
  Pilih $a^{-1} = -a \in H_1$, ambil sembarang $a \in H_1$ maka
  $a * a^{-1} = a + (-a) = 0$ dan $a^{-1} * a = (-a) + a = 0$ sehingga
  $a * a^{-1} = a^{-1} * a = e$ (**terpenuhi**)
  dimana 0 inversnya 15 dan 15 inversnya 0
  karena keempat aksioma terpenuhi, maka H_1 merupakan subgrup dari Z_{30}

b) $H_2 = \{0,4,8,12\}$
Pembuktian:

Tabel 2. Cayley 0, 4, 8, 12

$+_{30}$	**0**	**4**	**8**	**12**
0	0	4	8	12
4	4	8	12	16
8	8	12	16	20
12	12	16	20	24

Dengan memperhatikan tabel, maka:

- Aksioma I (sifat ketertutupan dan ketunggalan)
  $a * b \in H_2, \forall\, a, b \in H_2$ (tidak terpenuhi)
  karena aksioma pertama tidak terpenuhi, maka H_2 bukan subgrup dari Z_{30}

c) $H_3 = \{0,6,12,18,24\}$
Pembuktian:

Tabel 3. Cayley 0, 6, 12, 18, 24

$+_{30}$	**0**	**6**	**12**	**18**	**24**
0	0	6	12	18	24
6	6	12	18	24	0
12	12	18	24	0	6
18	18	24	0	6	12
24	24	0	6	12	18

Dengan memperhatikan tabel, maka:

- Aksioma I (sifat ketertutupan dan ketunggalan)
  $a * b \in H_3, \forall\, a, b \in H_3$ (**terpenuhi**)
- Aksioma II (sifat asosiatif)
  $a * (b * c) = (a * b) * c, \forall\, a, b, c \in H_3$(**terpenuhi**)
- Aksioma III (elemen identitas)
  Akan ditunjukkan $\exists\, e \in H_3 \ni \forall\, a \in H_3$berlaku $a * e = e * a = a$
  Pilih $e = 0 \in H_3$, ambil sembarang $a \in H_3$ maka
  $a * e = a + 0 = a$ dan $e * a = 0 + a = a$ sehingga
  $a * e = e * a = a$ (**terpenuhi**)
- Aksioma IV (invers)
  Akan ditunjukkan $\forall\, a \in H_1 \exists\, a^{-1} \in H_1 \ni a * a^{-1} = a^{-1} * a = e$

Pilih $a^{-1} = -a \in H_1$, ambil sembarang $a \in G$ maka
$a * a^{-1} = a + (-a) = 0$ dan $a^{-1} * a = (-a) + a = 0$ sehingga
$a * a^{-1} = a^{-1} * a = e$ (**terpenuhi**)
Dimana: $0^{-1} = 0$
$6^{-1} = 24$
$12^{-1} = 18$
$18^{-1} = 12$
$24^{-1} = 0$

Karena keempat aksioma terpenuhi, maka H_3 merupakan subgrup dari Z_{30}.

Berdasarkan penyelesaian yang dibuat oleh mahasiswa, dapat diperhatikan bahwa ketika meyelesaikan permasalahan tersebut, mahasiswa diminta untuk membuat tabel cayley terlebih dahulu. Yang dimana tabel ini digunakan untuk memudahkan mahasiswa dalam membuktikan keempat aksioma grup. Pembuatan tabel ini pastinya akan menguras banyak waktu mahasiswa dalam menyelesaikannya. Setelah mahasiswa menyelesaikan tabel cayley, mahasiswa dituntut untuk memahami 4 aksioma syarat grup dan menyesuaikannya dengan tabel cayley untuk melakukan pembuktian subgrup. Melihat hal ini, dipastikan akan banyak mahasiswa yang kesulitan dalam menentukan subgrup dari suatu grup.

Penyelesaian dengan Program Python

Dengan menggunakan bahasa pemrograman Python akan mempermudah proses verifikasi serta penentuan subgrup dari grup Z_n, serta menampilkan seluruh subgrup yang ada berdasarkan pembaginya. Penggunaan bahasa pemrograman Python dapat memberikan solusi yang lebih efisien dan akurat dibandingkan metode manual, terutama ketika berhadapan dengan grup yang memiliki banyak elemen. Dalam pembahasan ini, akan dijelaskan tahapan pembuatan program Python yang dapat digunakan untuk menentukan subgrup dalam grup Z_n.

Tahapan pembuatan Implementasi Program Python:

1. Pembuatan Grup Z_n
   Tahap pertama dalam proses ini adalah membuat grup Z_n, yaitu himpunan yang berisi elemen-elemen dari bilangan bulat non-negatif hingga $n-1$. Pada Python, hal ini direpresentasikan dalam fungsi ‘generate_group(n)’ yang menghasilkan himpunan elemen-elemen dalam Z_n.

```
def generate_group(n):
    return set(range(n))
```

Gambar 1. Kode Python untuk Membuat Himpunan Elemen dalam Grup Z_n

   Fungsi ini menghasilkan himpunan elemen-elemen $\{0, 1, 2, \ldots, n-1\}$ untuk grup Z_n.
2. Pembuatan Subgrup
   Langkah selanjutnya adalah membuat subgrup dari Z_n. Subgrup dihasilkan dengan menentukan pembentuk (generator) subgrup, yang dinotasikan dengan ‘d’. Fungsi ‘generate_subgroup(n, d)’ akan membentuk subgrup dengan menggunakan rumus $(d * i) \% n$, di mana i adalah elemen dari Z_n.

```
def generate_subgroup(n, d):
    return { (d * i) % n for i in range(n) }
```

Gambar 2. Kode Python untuk Membuat Subgrup dari Z_n

   Subgrup dalam grup dapat dibentuk oleh salah satu pembagi dari ‘n’. Sebagai contoh, untuk grup Z_6 dengan pembentuk d = 2, elemen subgrup yang terbentuk adalah $\{0, 2, 4\}$. Proses ini memastikan bahwa subgrup yang dihasilkan memenuhi sifat-sifat grup, yaitu tertutup terhadap operasi yang berlaku di dalam grup induk (Saragih, 2012).

3. Menentukan Semua Grup
   Program juga dapat menghitung seluruh subgrup yang mungkin terbentuk dalam Z_n berdasarkan pembagi dari 'n'. Fungsi 'find_subgroups(n)' mengidentifikasi pembagi dari 'n' dan menggunakannya untuk membentuk subgrup.

```
def find_subgroups(n):
    divisors = [d for d in range(1, n+1) if n % d == 0]
    subgroups = {d: generate_subgroup(n, n//d) for d in divisors}
    return subgroups
```

Gambar 3. Kode Python untuk Membentuk Subgrup

   Fungsi ini memanfaatkan konsep pembagian bilangan bulat, di mana setiap pembagi dari 'n' akan membentuk subgrup tertentu. Sebagai contoh, untuk grup Z_6, pembagi dari 6 adalah $\{1, 2, 3, 6\}$, yang akan membentuk subgrup dengan elemen $\{0\}$, $\{0, 3\}$, $\{0, 2, 4\}$, dan $\{0, 1, 2, 3, 4, 5\}$.
4. Memvalidasi Subgrup
   Fungsi 'is_valid_subgroup(n, elements)' digunakan untuk memverifikasi apakah sekumpulan elemen yang diberikan oleh pengguna membentuk subgrup yang valid. Kriteria yang digunakan adalah bahwa subgrup harus memiliki elemen identitas (0 untuk grup Z_n) dan harus tertutup terhadap operasi penjumlahan dan pengurangan modulo n.

```
def is_valid_subgroup(n, elements):
    if 0 not in elements:
        return False
    for a in elements:
        for b in elements:
            if (a + b) % n not in elements:
                return False
            if (a - b) % n not in elements:
                return False
    return True
```

Gambar 4. Kode Python untuk Memvalidasi Subgrup

   Untuk memvalidasi suatu subgrup, program pertama-tama memastikan bahwa elemen 0 (sebagai elemen identitas) ada dalam himpunan. Selanjutnya, dilakukan pengecekan apakah hasil penjumlahan dan pengurangan modulo n dari dua elemen dalam himpunan juga termasuk dalam himpunan tersebut. Jika kedua kondisi ini terpenuhi, maka himpunan elemen tersebut adalah subgrup yang valid.
5. Menggabungkan Semua Tahapan dalam Program Utama
   Seluruh fungsi di atas digabungkan ke dalam satu fungsi utama 'main()' yang akan berinteraksi dengan pengguna. Program ini memungkinkan pengguna untuk memasukkan nilai n, menentukan elemen-elemen yang ingin diuji sebagai subgrup, serta menampilkan semua subgrup yang dapat terbentuk.

```python
def main():
    n = int(input("Masukkan nilai n untuk grup Z_n: "))
    print(f"\nGrup Z_{n}: {generate_group(n)}")

    elements = set(map(int, input("Masukkan elemen-elemen yang ingin dicek (dipisahkan dengan spasi): ").split()))
    if is_valid_subgroup(n, elements):
        print("\nElemen-elemen tersebut membentuk subgrup.")
    else:
        print("\nElemen-elemen tersebut BUKAN subgrup.")

    subgroups = find_subgroups(n)
    print(f"\nJumlah subgrup: {len(subgroups)}")
    for d, subgroup in subgroups.items():
        print(f"Subgrup dengan orde {d}: {subgroup}")

if __name__ == "__main__":
    main()
```

Gambar 5. Gabungan Kode Python

Program ini meminta input dari pengguna untuk nilai n, lalu menampilkan grup Z_n yang terbentuk. Pengguna dapat memasukkan elemen-elemen untuk diuji apakah mereka membentuk subgrup yang valid. Program juga menghitung dan menampilkan seluruh subgrup yang dapat terbentuk dari grup tersebut, serta menunjukkan jumlah subgrup berdasarkan ordo.

Dengan demikian penggunaan Python, proses verifikasi dan penelusuran subgroup dalam grup Z_n dapat dilakukan secara otomatis dan efisien. Program ini memungkin pengguna untuk membentuk grup Z_n secara otomatis, menguji apakah suatu himpunan elemen membentuk subgroup yang valid atau tidak, serta menampilkan seluruh subgroup yang mungkin terbentuk dalam waktu yang cepat serta memberikan hasil yang akurat dalam menetukan struktur subgrup dari suatu grup.

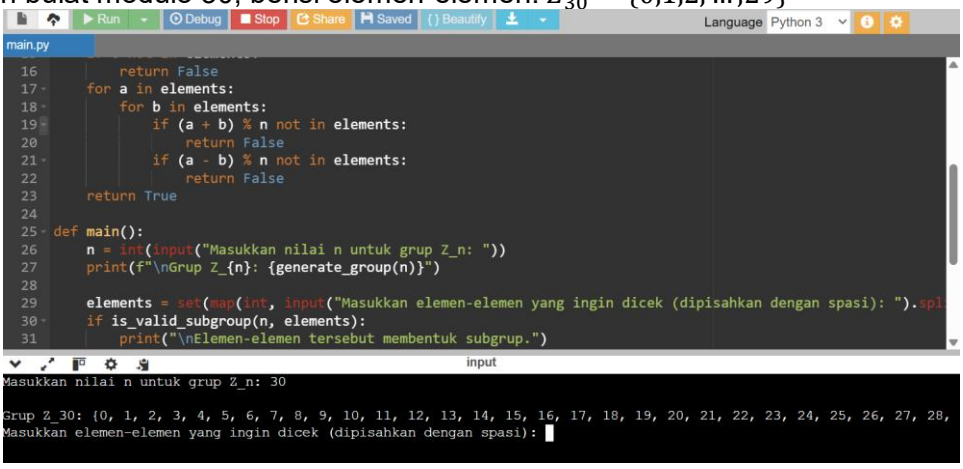
Identifikasi subgrup dari Z_{30}. Apakah $H_1 = \{0,15\}$, $H_2 = \{0,4,8,12\}$, $H_3 = \{0,6,12,18,24\}$ merupakan subrup dari Z_{30}?

Penyelesaian:

Langkah-langkah dalam menentukan subgroup dari Z_{30} menggunakan Bahasa Pemrograman Python:

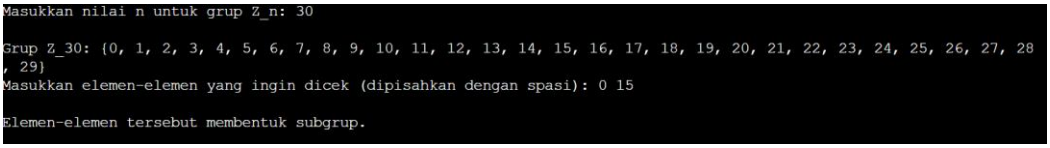
1. Input nilai n untuk grup Z_n

   Pada program ini, kita akan diminta untuk memasukkan nilai n yang di gunakan untuk membentuk grup Z_n. Berdasarkan soal di atas kita menggunakan $n = 30$. Grup Z_{30} adalah grup bilangan bulat modulo 30, berisi elemen-elemen: $Z_{30} = \{0,1,2, \dots ,29\}$



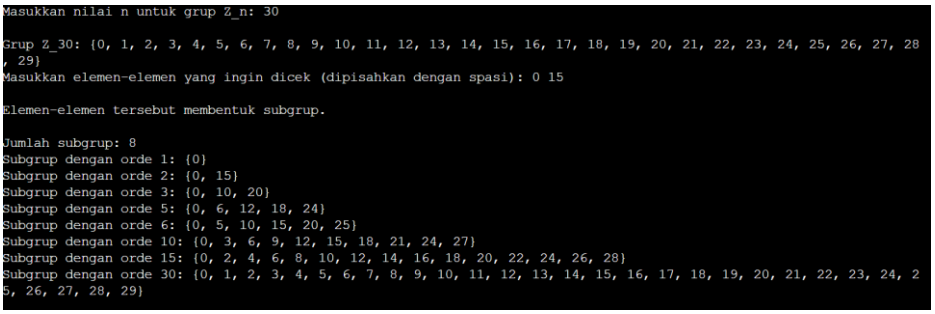
Gambar 6. Memasukkan Nilai n Grup Z_n

2. Masukkan elemen yang ingin diperiksa sebagai subgrup
   a. Program ini akan meminta pengguna untuk memasukkan elemen-elemen yang akan dicek apakah elemen-elemen tersebut membentuk subgrup dari grup Z_{30} atau tidak. Pengguna akan memasukkan elemen-elemen ini dengan dipisahkan oleh spasi. Berdasarkan soal kita akan memasukkan elemen yang di cek yaitu $H_1 = \{0,15\}$.



Gambar 7. Memasukkan Elemen Subgrup yang Dicek

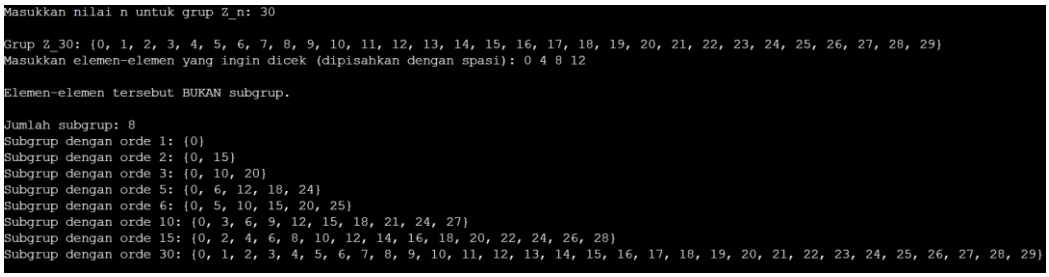
Fungsi ‘is_valid_subgroup(n, elements)’ akan memeriksa apakah kumpulan elemen yang diberikan memenuhi syarat-syarat untuk menjadi subgroup. Jika syarat-syarat ini terpenuhi, maka elemen-elemen tersebut membentuk subgrup. Jika tidak, maka mereka bukan subgrup.



Gambar 8. Validasi Elemen Subgrup

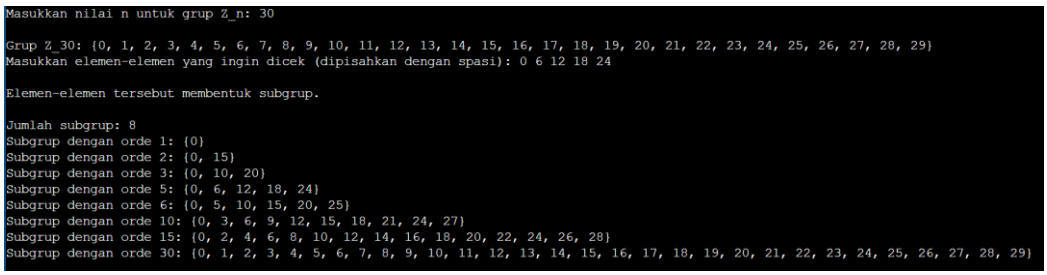
Berdasarkan gambar di atas bisa di simpulkan bahwa $H_1 = \{0,15\}$ merupkan subgrup dari Z_{30}. Program ini juga menemukan semua subgrup dari grup Z_{30}, dengan menggunakan fungsi ‘find_subgroups(n)’. Ini dilakukan dengan mengidentifikasi pembagi dari n dan menghasilkan subgrup yang sesuai dengan orde pembagi tersebut.

b. Hal yang sama akan kita lakukan pada $H_2 = \{0,4,8,12\}$. Yaitu memasukkan nilai $n = 30$. Kemudian memasukkan subgrup yaitu 0, 4, 8, 12 dengan spasi di setiap angka. Kemudian akan muncul hasil berupa pernyataan bahwa himpunan H_2 bukan subgrup.



Gambar 9. Hasil Bukan Subgrup

c. Kemudiaan subgroup yang ke tiga yaitu, $H_3 = \{0,6,12,18,24\}$ hasil berupa pernyataan bahwa himpunan tersebut adalah subgrup.



Gambar 10. Hasil Berupa Subgrup

SIMPULAN

Hasil penelitian menunjukkan bahwa penggunaan Python dalam analisis struktur aljabar, khususnya dalam identifikasi subgrup, terbukti praktis dan efisien. Program yang dikembangkan mampu mengidentifikasi subgrup secara akurat tanpa perlu melakukan verifikasi manual terhadap syarat-syarat grup. Hal ini membuktikan bahwa Python dapat menjadi alat yang sistematis dan andal dalam eksplorasi konsep aljabar abstrak. Oleh karena itu, disarankan bagi pendidik dan mahasiswa matematika untuk memanfaatkan Python sebagai alat bantu pembelajaran. Selain itu, pengembang perangkat lunak pendidikan dapat mengembangkan fitur tambahan untuk meningkatkan kemampuan program dalam menganalisis struktur aljabar yang lebih kompleks.

DAFTAR PUSTAKA

Assyakurrohim, D., Ikhram, D., Sirodj, R. A., & Afgani, M. W. (2022). Metode studi kasus dalam penelitian kualitatif. *Jurnal Pendidikan Sains Dan Komputer*. Vol: 3(01): 1-9.

Das, U., Lawson, A., Mayfield, C., & Norouzi, N. (2024). *Introduction to Python Programming*. Texas: OpenStax.

Faizah, Hanim. (2019). Pemahaman Mahasiswa Tentang Konsep Grup Pada Mata Kuliah Struktur Aljabar. *Jurnal: Journal of Mathematics Education, Science and Technology*. Vol: 4(1): 23-34.

Isa, R. A. (2024). *Struktur dan sifat-sifat K-Aljabar* (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).

Lutz, Mark. (2009). *Learning Phyton*. Sebastopol: O'Reilly Media.

Muktyas, Indra Bayu & Arifin, Samsul. (2018). Sebarang Pembangun Subgrup Siklik Dari Suatu Grup (Z_n ,+). *Jurnal Matematika "MANTIK"*.Vol: 4(2): 116-121.

Naibaho, E., Sihombing, L. P., Situmorang, C., & Manik, F. D. (2025). Model Pembelajaran Snowball Throwing Dalam Peningkatan Penalaran Belajar Struktur Aljabar. *As-Salam: Journal Islamic Social Sciences and Humanities*, *3*(1), 121-132.

Ritonga, A., & Yahfizham, Y. (2023). Studi literatur perbandingan bahasa pemrograman C++ dan bahasa pemrograman Python pada algoritma pemrograman. *Jurnal Teknik Informatika dan Teknologi Informasi*, *3*(3), 56-63.

Saragih, S. (2012). *Sruktur Aljabar 1*. Medan: LARISPA INDONESIA.

Simarmata, S. M., Sinaga, B., & Syahputra, H. (2022). Analisis Kemampuan Pemahaman Konsep Matematika Siswa Dalam Penerapan Model Discovery Learning Berbantuan Matlab. *Jurnal Cendekia*, *6*(1), 692-701.