Volume. 1 No 2 Oktober (2024)

e-ISSN: 3062-7737
p-ISSN: 3063-2137

Semeton Mathematics Journal

Homepage jurnal: http://eigen.unram.ac.id/semeton

Penyelesaian Polinomial *Irreducible* pada $\mathbb{Z}_p$ dan FPB, KPK Dua Polinomial pada $\mathbb{F}_n$ Menggunakan Python

(*Solving Irreducible Polynomials over* $\mathbb{Z}_p$ *and GCD, LCM of Two Polynomials over* $\mathbb{F}_n$ *Using Python*)

***Yudha Sakti Nusantara*[a*]*, Wahyu Maulana*[b]**

[a.] Program Studi Matematika, Universitas Mataram, Indonesia. Email: yudha.saktil30@gmail.com

[b.] Program Studi Matematika, Universitas Mataram, Indonesia. Email: wahyu.maulaaana@gmail.com

ABSTRACT

This research aims to solve the irreducible polynomial problem over finite field $\mathbb{Z}_\mathrm{p}$ and determine the Greatest Common Divisor (GCD) and Least Common Multiple (LCM) of two polynomials over finite field $\mathbb{F}_\mathrm{n}$ using Python programming language. In the digital age, programming plays an important role in various disciplines. Python, with its simple syntax and computational libraries like SymPy, has become a top choice among the various programming languages available. Polynomials appear frequently in the computer field, especially in cryptographic algorithms, data compression, and error coding. This research utilizes Euclid's Algorithm to determine the GCD and LCM of two polynomials over a finite field $\mathbb{F}_\mathrm{n}$, as well as evaluate the irreducibility of polynomials over a finite field $\mathbb{Z}_\mathrm{p}$. Determining irreducibility is very important in polynomial theory and is a challenging task if done manually. With the help of Python, this research produces a script or syntax that is able to automate the process, thus saving time and reducing complexity. The final result of this research is an effective Python script or syntax to determine the GCD and LCM of two polynomials over $\mathbb{F}_\mathrm{n}$, as well as evaluate whether a polynomial is reducible or irreducible over $\mathbb{Z}_\mathrm{p}$.

Keywords: Reducible Polynomial, GCD, LCM, Python

ABSTRAK

Penelitian ini bertujuan untuk menyelesaikan masalah polinomial *irreducible* atas lapangan hingga $\mathbb{Z}_p$ dan menentukan Faktor Persekutuan Terbesar (FPB) serta Kelipatan Persekutuan Terkecil (KPK) dari dua polinomial atas lapangan hingga $\mathbb{F}_n$ menggunakan bahasa pemrograman Python. Dalam era digital, pemrograman memainkan peran penting dalam berbagai disiplin ilmu. Python, dengan sintaksis sederhana dan *library* komputasi seperti SymPy, telah menjadi pilihan utama di antara berbagai bahasa pemrograman yang ada. Polinomial sering muncul dalam bidang komputer, terutama dalam algoritma kriptografi, kompresi data, dan pengkodean kesalahan. Penelitian ini memanfaatkan Algoritma Euclid untuk menentukan FPB dan KPK dua polinomial atas lapangan hingga $\mathbb{F}_n$, serta mengevaluasi *irreducibility* polinomial atas lapangan hingga $\mathbb{Z}_p$. Penentuan *irreducibility* adalah hal yang sangat penting dalam teori polinomial dan merupakan tugas yang menantang jika dilakukan secara manual. Dengan bantuan Python, penelitian ini menghasilkan skrip atau syntax yang mampu mengotomatisasi proses tersebut, sehingga menghemat waktu dan mengurangi kompleksitas. Hasil akhir penelitian ini

* Corresponding author
e-mail: yudha.saktil30@gmail.com

adalah sebuah skrip atau syntax Python yang efektif untuk menentukan FPB dan KPK dua polinomial atas $\mathbb{F}_n$, serta mengevaluasi apakah suatu polinomial adalah *reducible* atau *irreducible* atas $\mathbb{Z}_p$.

Keywords: Polinomial *Irreducible*, FPB, KPK, Python

Diterima: 22-06-2024; Doi: https://doi.org/10.29303/semeton.v1i2.227
Disetujui: 07-10-2024;

1. Pendahuluan

Dalam era digital saat ini, pemrograman mengambil peran penting dalam berbagai bidang ilmu pengetahuan dan teknologi. Ada beberapa bahasa pemrograman yang terkenal yaitu JavaScript, C++, Julia, *Python*, dan lainnya [1]. Di antara banyaknya Bahasa Pemrograman tersebut, *Python* menjadi bahasa pemrograman yang paling populer dan banyak digunakan. Hal tersebut dikarenakan *Python* memiliki sintaksis yang paling sederhana dan lebih mudah dipahami dibandingkan bahasa yang lainnya [2]. Selain itu, *Python* juga memiliki berbagai pustaka *(library)* yang dapat mendukung kebutuhan komputasi salah satunya adalah SymPy.

Polinomial merupakan salah satu objek dalam matematika yang sering muncul dalam bidang komputer. Dalam bidang komputer, polinomial mengambil peran dalam pembentukan algoritma kriptografi dan pengkodean kesalahan [3]. Salah satu masalah yang menarik dari polinomial dalam bidang matematika adalah penentuan Faktor Persekutuan Terbesar (FPB) dan Kelipatan Persekutuan Terkecil (KPK) dari dua polinomial atas lapangan $\mathbb{F}_n$ (bidang dengan elemen bilangan bulat modulo bilangan n dengan $n = p^k$ dengan p bilangan prima). Algoritma yang paling sering digunakan untuk menyelesaikan permasalahan tersebut adalah Algoritma Euclid [4].

Penentuan FPB dan KPK dari dua buah polinom tak terlepas dari konsep polinomial *reducible* (yang dapat direduksi) dan *irreducible* (tidak dapat diredukasi). Polinomial *reducible* dan *irreducible* merupakan konsep dasar dalam teori polinomial. Polinomial *reducible* adalah polinomial yang dapat diuraikan menjadi faktor-faktor polinomial yang lebih rendah, sedangkan polinomial *irreducible* tidak dapat diuraikan lebih lanjut [5]. Dalam konteks bidang $\mathbb{Z}_p$, menentukan apakah suatu polinomial dapat direduksi menjadi pertanyaan yang menarik dan penting.

Penyelesaian masalah penentuan FPB dan KPK atas lapangan hingga $\mathbb{F}_n$, serta menentukan apakah suatu polinom adalah *reducible* atau tidak atas lapangan hingga $\mathbb{Z}_p$, sering kali memerlukan waktu yang lama dan kompleksitas yang tinggi jika dilakukan secara manual. Dalam menyelesaikan masalah matematika, *python* telah banyak digunakan untuk menyelesaikan kasus matematika yang kompleks dan waktu komputasi yang lama. Seperti pada tahun 2020 Nascimento, Fricke, dan Viana menggunakan python untuk menyelesaikan masalah Persamaan Diferensial Biasa (PDB) [6]. Kemudian pada tahun 2023 Rackauckas membandingkan beberapa bahasa pemrograman seperti *python,* r, Julia, C untuk menyelesaikan masalah Persamaan Diferensial [7]. Oleh karena itu, penelitian ini bertujuan untuk menyelesaikan masalah penentuan FPB dan KPK dari dua polinom atas lapangan hingga $\mathbb{F}_n$ dengan menggunakan *Python*. Selain itu, penelitian ini juga akan membahas penentuan apakah suatu polinom adalah *reducible* atau tidak atas lapangan hingga $\mathbb{Z}_p$dengan bantuan Python. Hasil akhir dari penelitian ini adalah sebuah skrip *Python* yang dapat menyelesaikan masalah FPB dan KPK dari dua polinom atas lapangan hingga $\mathbb{F}_n$, serta untuk menentukan apakah suatu polinom adalah reducible atau tidak atas lapangan hingga $\mathbb{Z}_p$.

1.1. Python

Python pertama kali diperkenalkan oleh Guido Van Rossum pada 1991. Nama *Python* sendiri terinspirasi dari nama sebuah acara di BBC pada era 1980an dengan judul "*Monty Python's Flying Circus*" [8]. *Python* menjadi bahasa pemrograman yang sangat popular di era sekarang karena kemudahan sintaksisnya serta dapat diperoleh secara gratis [9]. Selain itu, *Python* juga telah dilengkapi dengan berbagai macam *Library* (modul) siap pakai sehingga memudahkan pengguna untuk menggunakannya. SymPy merupakan salah satu *Library* dalam *Python* yang berguna untuk melakukan penyelesaian masalah Teori Bilangan seperti penentuan FPB dan KPK [10]. Selain itu, ketika kita ingin membuat sebuah aplikasi menggunakan *Python* terdapat *Library* PyQt yang sangat berguna untuk mendesain tampilan suatu aplikasi yang dibuat [11].

1.2. Polinomial

Polinomial $p(x)$ berderajat n, didefinisikan sebagai suatu fungsi berbentuk:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \quad (1)$$

dengan a_i adalah konstanta real, $i = 0, 1, 2, \dots, n$ dan $a_n \neq 0$.

di mana:

x : Peubah

$a_0, a_1, a_2, \dots, a_n$: Nilai koefisien

x : Orde atau derajat tertinggi persamaan

[12].

1.3. Polinomial atas lapangan hingga $\mathbb{Z}_p$ dan $\mathbb{F}_n$

Lapangan $\mathbb{Z}_p$ merupakan kasus khusus dari lapangan $\mathbb{F}_n$ ketika $n = p$, dengan p bilangan prima. Polinomial atas lapangan $\mathbb{Z}_p$ adalah polinomial yang koefisien suku-sukunya merupakan himpunan terurut *infinite* dari ring $\mathbb{Z}_p$ dengan sebagian besar bilangan terbatas pada elemen bukan nol [13]. Dan polinomial atas lapangan $\mathbb{F}_n$ adalah polinomial yang koefisiennya berasal dari lapangan $\mathbb{F}_n$ [14].

1.4. Polinomial reducible

Sebuah polinomial $f(x)$ dengan derajat positif dikatakan *reducible* (atas lapangan $\mathbb{Z}_p$) jika terdapat dua polinomial $h(x)$ dan $r(x)$ dengan koefisien dalam $\mathbb{Z}_p$ sehingga $\deg(h(x)) < \deg(g(x))$, $\deg(r(x)) < \deg(g(x))$ dan $g(x) = h(x).r(x)$. Sebaliknya, polinomial $g(x)$ dengan derajat positif dikatakan *irreducible* (atas lapangan $\mathbb{Z}_p$) [15]. Terdapat sebuah teorema yang sangat berguna untuk menentukan apakah polinom *reducible* atau *irreducible* atas lapangan $\mathbb{Z}_p$.

Teorema 1. Misalkan $\mathbb{F}$ lapangan hingga, $h(x) \in \mathbb{F}[x]$, $\alpha \in \mathbb{F}$. $(x - \alpha)$ faktor dari $h(x)$ jika $h(\alpha) = 0$.

Contoh kasus polinomial *irreducible*:

Misalnya kita memiliki polinomial $x^4 + x + 1$ atas $\mathbb{Z}_2$, selanjutnya kita akan selidiki apakah polinomial $x^4 + x + 1$ atas $\mathbb{Z}_2$ *irreducible* atau tidak. Dengan memanfaatkan Teorema 1, didapatkan hasil sebagai berikut.

$f(x) = x^4 + x + 1$

$\mathbb{Z}_2 = \{0, 1\}$, jadi nilai α diambil dari $\mathbb{Z}_2$

Tabel 1. Pensubstitusian nilai α ke $f(\alpha)$

$\boldsymbol{\alpha}$	$\boldsymbol{f(\alpha)}$
0	1
1	1

Karena tidak terdapat α yang membuat $f(\alpha) = 0$, maka polinom $x^4 + x + 1$ *irreducible* atas $\mathbb{Z}_2$.

1.5. Algoritma Euclid

Algoritma Euclid adalah sebuah metode yang efisien untuk menghitung Faktor Persekutuan Terbesar (FPB). FPB dari dua polinomial adalah polinomial yang membagi kedua polinomial tersebut tanpa menyisakan sisa yang lebih rendah daripada polinomial itu sendiri [16].

Langkah-langkah Algoritma Euclid:

1. Misalkan $f(x)$ dan $g(x)$ adalah dua polinomial dengan koefisien dalam suatu lapangan, dan $\deg(f(x)) \geq \deg(g(x))$.
2. Jika $g(x) = 0$, maka $FPB(f(x), g(x)) = f(x)$.

3. Jika $g(x) \neq 0$, hitung $r(x) = f(x) \; mod \; g(x)$, yang berarti $f(x) = q(x).g(x) + r(x)$, di mana $q(x)$ adalah polinom hasil bagi dan $\deg(r(x)) < \deg(g(x))$.
4. Ganti $f(x)$ dengan $g(x)$ dan $g(x)$ dengan $r(x)$.
5. Ulangi langkah 3 dan 4 sampai $g(x) = 0$. Pada saat ini, $FPB(f(x), g(x))$ adalah polinom terakhir yang bukan nol, yaitu $g(x)$

Untuk menentukan KPK dari 2 polinom dapat digunakan rumus berikut:

$$KPK(f(x), g(x)) = \frac{f(x).g(x)}{FPB(f(x), g(x))} \quad (2)$$

Contoh soal penggunaan Algoritma Euclid:

Misalnya kita memiliki polinomial $x^2 + 7x + 6$ dan $x^2 - 5x - 6$ atas lapangan $\mathbb{F}_{20}$, selanjutnya kita akan selidiki FPB dan KPK dari kedua polynomial tersebut. Dengan menggunakan Algoritma Euclid didapatkan hasil sebagai berikut:

Misalkan $a^2 + 7a + 6 = f(a)$ dan $a^2 - 5a - 6 = g(a)$, $q(a) = faktor\ dari\ f(a)$, dan $r(a) =$ sisa bagi.

- Pembagian pertama

$f(a) = q(a).g(a) + r(a)$
$r(a) = f(a) - q(a).g(a)$
$r(a) = a^2 + 7a + 6 - (1).(a^2 - 5a - 6)$
$r(a) = a^2 + 7a + 6 - a^2 + 5a + 6$
$r(a) = 12a + 12$

- Pembagian kedua

$g(a) = q_2(a).r(a) + r_2(a)$
$r_2(a) = g(a) - q_2(a).r(a)$
$r_2(a) = (a^2 - 5a - 6) - \left(\frac{a}{12}\right).(12a + 12)$
$r_2(a) = a^2 - 5a - 6 - 12a - 12$
$r_2(a) = -6a - 6 = -6(a + 1)$

- Pembagian ketiga

Karena $r_2(a)$ adalah faktor dari $r(a)$, kita dapatkan sisanya adalah nol.

Jadi berdasarkan perhitungan yang telah dilakukan, didapatkan FPB dari $a^2 + 7a + 6$ dan $a^2 - 5a - 6$ atas lapangan $\mathbb{F}_{20}$ adalah $(a + 1)$.

Dan KPK dari $a^2 + 7a + 6$ dan $a^2 - 5a - 6$ atas lapangan $\mathbb{F}_{20}$ adalah:

$$KPK((a^2 + 7a + 6),\ (a^2 - 5a - 6)) = \frac{(a^2 + 7a + 6).(a^2 - 5a - 6)}{a + 1}$$
$$KPK((a^2 + 7a + 6),\ (a^2 - 5a - 6)) = \frac{(a + 6)(a + 1).(a - 6)(a + 1)}{a + 1}$$
$$KPK((a^2 + 7a + 6),\ (a^2 - 5a - 6)) = (a + 6)(a + 1)(a - 6)$$
$$KPK((a^2 + 7a + 6),\ (a^2 - 5a - 6)) = a^3 + a^2 - 36a - 36$$
$$KPK((a^2 + 7a + 6),\ (a^2 - 5a - 6)) = a^3 + a^2 + 4a + 4$$

Jadi didapatkan KPK dari $a^2 + 7a + 6$ dan $a^2 - 5a - 6$ atas lapangan $\mathbb{F}_{20}$ adalah $a^3 + a^2 + 4a + 4$.

Diperoleh hasil akhirnya:

- FPB dari $a^2 + 7a + 6$ dan $a^2 - 5a - 6$ atas lapangan $\mathbb{F}_{20}$ adalah $(a + 1)$.
- KPK dari $a^2 + 7a + 6$ dan $a^2 - 5a - 6$ atas lapangan $\mathbb{F}_{20}$ adalah $a^3 + a^2 + 4a + 4$.

2. Metode

Penelitian kali ini dilakukan dengan melakukan studi literatur mengenai reduksi polinom atas lapangan hingga $\mathbb{Z}_p$, serta FPB, dan KPK dari 2 polinom atas lapangan hingga $\mathbb{Z}_p$. Membuat algoritma untuk menyelesaikan masalah tersebut kemudian diimplementasikan dalam Python. Selanjutnya dilakukan pengujian terhadap algoritma yang telah dibuat, langkah selengkapnya sebagai berikut:

1. Melakukan studi literatur studi literatur mengenai reduksi polinom atas lapangan hingga $\mathbb{Z}_p$, serta FPB, dan KPK dari 2 polinom atas lapangan hingga $\mathbb{Z}_p$.
2. Membuat algoritma penyelesaian reduksi polinom atas lapangan hingga $\mathbb{Z}_p$ dengan memanfaatkan Teorema Berikut pseudocode dari algoritmanya:

```
FUNCTION isFactor(F, f, alpha)
    # Input:
    # F: Lapangan hingga
    # f: Polinom dengan koefisien dalam F
    # alpha: Elemen dalam F

    # Evaluasi polinom f di titik alpha
    result = EVALUATE_POLYNOMIAL(f, alpha)

    # Jika hasil evaluasi adalah 0, maka (x – alpha) faktor dari f(x)
    IF result == 0 THEN
        RETURN True  # (x - alpha) adalah faktor dari f(x)
    ELSE
        RETURN False  # (x - alpha) bukan faktor dari f(x)
    END IF
END FUNCTION
```

3. Membuat algoritma penyelesaian masalah penentuan FPB dan KPK dari 2 polinom atas lapangan hingga $\mathbb{Z}_p$. Berikut pseudocode dari algoritmanya:

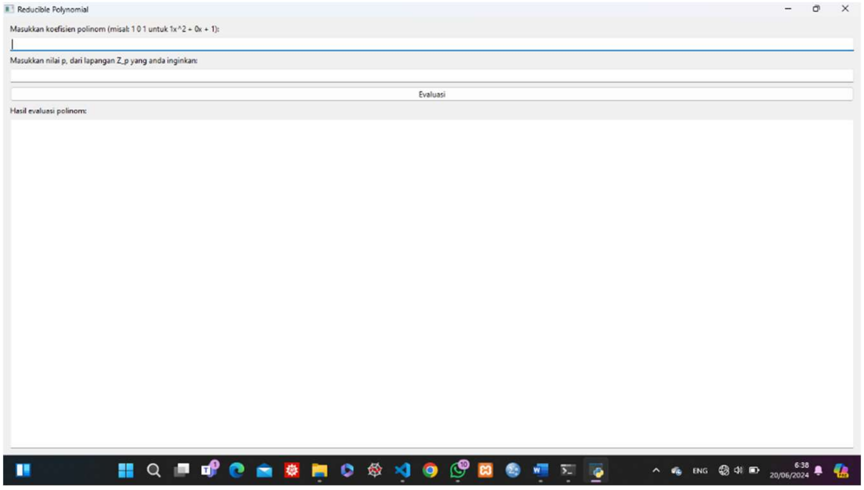
```
function FPB_polynomial(h(a), l(a)):
  if l(a) == 0:
      return h(a)

  while l(a) != 0:
      r(a) := h(a) mod l(a)
      h(a) := l(a)
      l(a) := r(a)

  return h(a)
```

4. Implementasikan pseudocode atau algoritma-algoritma tersebut ke dalam Python. Pada kali ini akan digunakan *Library* SymPy dan PyQt6 dari Python. *Library* SymPy berfungsi untuk menyelesaikan masalah polinomial dan *Library* PyQt6 berfungsi untuk mengatur tampilan ketika kode Python dijalankan
5. Melakukan uji coba terhadap kode Python yang telah dibuat. Pengujian deilakukan dengan menggunakan laptop Asus A416MA dengan spesifikasi Processor Intel Celeron.

3. Hasil dan Pembahasan

3.1. Penentuan polinomial reducible atas lapangan $\mathbb{Z}_p$

Ketika kode atau syntax dijalankan, maka akan muncul tampilan seperti berikut.

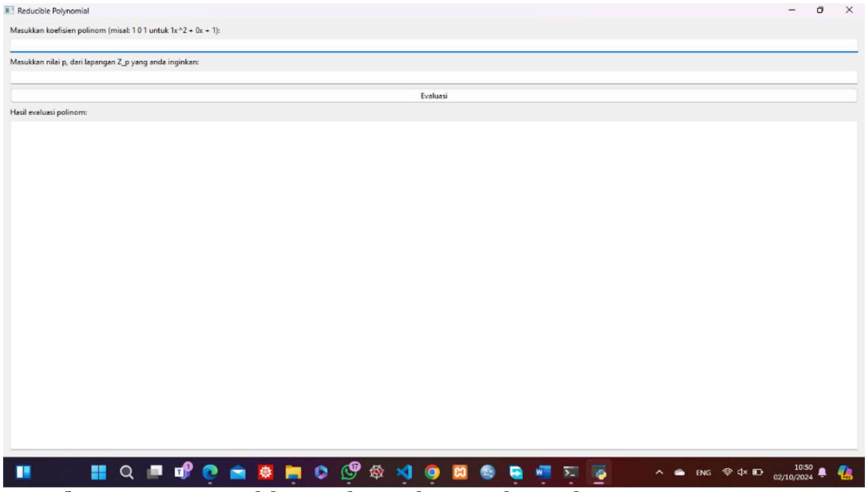


Gambar 1. tampilan awal

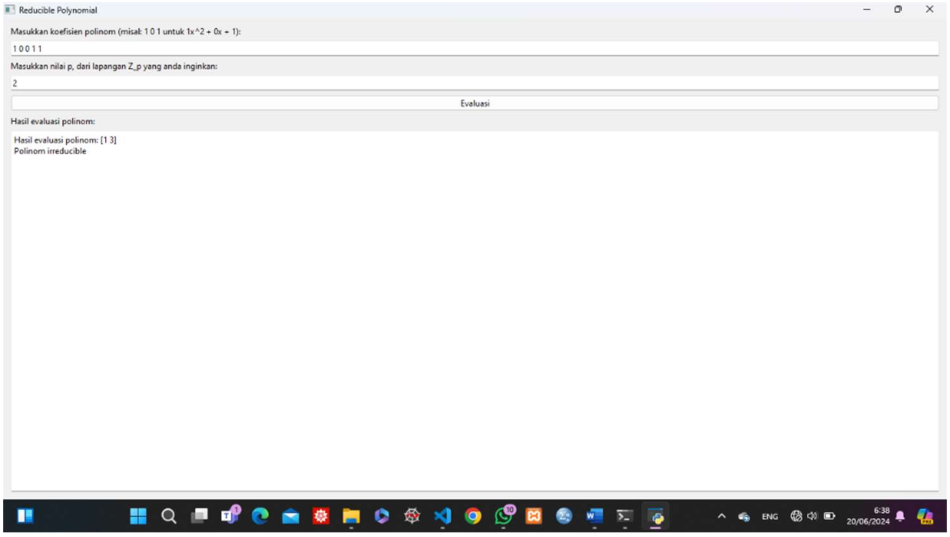
Pada baris pertama terdapat perintah untuk memasukkan polinomial yang ingin dievalusi *Reducible* atau tidak. Polinom yang ingin dievaluasi hanya ditulis koefisiennya, dipisah dengan tanda spasi serta dimulai dari pangkat tertinggi, misalnya pada kali ini kita ambil contoh $x^4 + x + 1$ diubah menjadi [1 0 0 1 1].

Kemudian pada baris kedua terdapat perintah untuk memasukkan nilai p dari $\mathbb{Z}_p$ yang diinginkan. Untuk bagian ini, kita langsung bisa memasukkan nilai atau angka dari p yang kita inginkan. Pada kali ini kita ambil contoh untuk nilai p = 2 ($\mathbb{Z}_2$).

Setelah kedua baris terisi, kita bisa mengevalusi polinom dengan menekan bagian evaluasi. Berikut implementasinya dan hasil evaluasinya.



Gambar 2. memasukkan nilai polinom dan nilai p

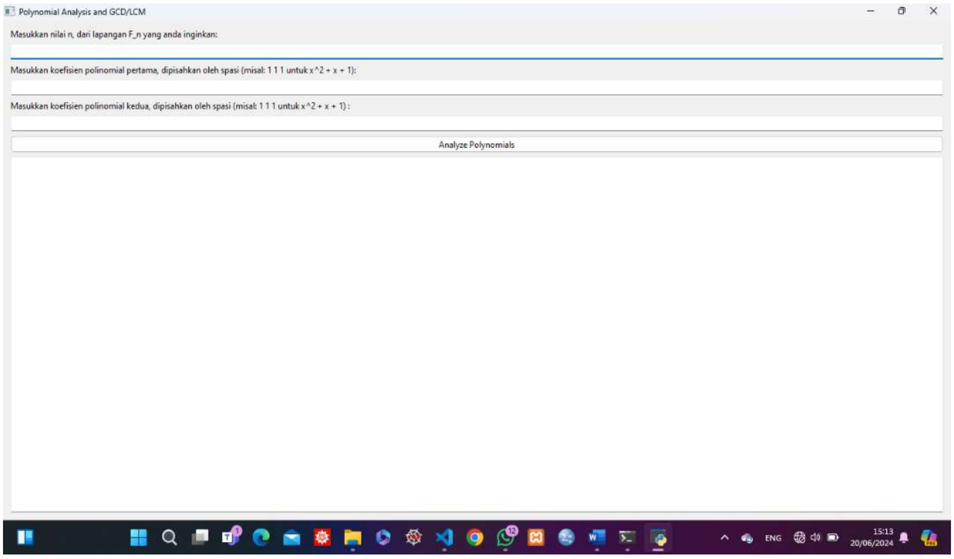


Gambar 3. hasil evaluasi

Dari gambar 3 kita dapatkan bahwa polinomial $x^4 + x + 1$ atas $\mathbb{Z}_2$ merupakan polinomial *irreducible*. Jawaban tersebut merupakan jawaban yang benar, karena dengan melakukan perhitungan secara manual benar bahwa $x^4 + x + 1$ atas $\mathbb{Z}_2$ merupakan polinomial *irreducible*.

3.2. Penentuan FPB dan KPK dua polynomial atas lapangan $\mathbb{F}_n$

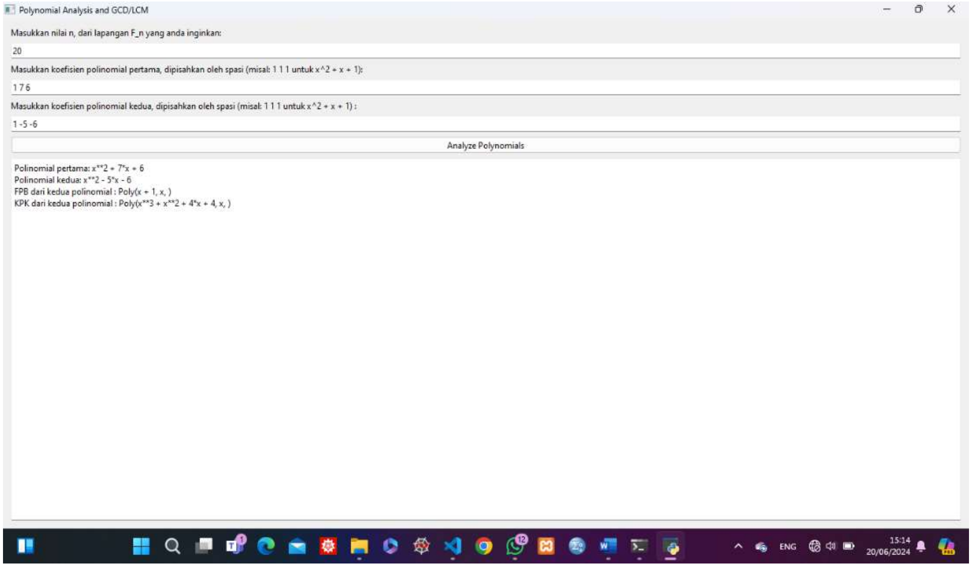
Ketika kode atau syntax dijalankan diperoleh tampilan sebagai berikut.



Gambar 4. tampilan awal

Pada baris pertama baris perintah untuk menentukan nilai n yang diinginkan dari lapangan $\mathbb{F}_n$, kemudian pada baris kedua dan ketiga berisi perintah untuk memasukkan nilai koefisien dari polinom yang ingin dievaluasi. Sama seperti kode untuk *reducible* polinom, polinom yang ingin dievaluasi hanya ditulis koefisiennya dan dipisah dengan tanda spasi serta dimulai dari pangkat tertinggi, misalnya $x^4 + x + 1$ diubah menjadi [1 0 0 1 1].

Misal kita ambil contoh, $x^2 + 7x + 6$ dan $x^2 - 5x - 6$ atas lapangan $\mathbb{F}_{20}$. Berikut implementasinya ke dalam kode yang dibuat.



Gambar 5. hasil evaluasi

Dari gambar 6 didapatkan FPB dari polinom yang ingin dievaluasi adalah $(x + 1)$, serta KPK dari polinom yang ingin dievaluasi adalah $x^3 + x^2 + 4x + 4$. Hasil tersebut merupakan hasil yang tepat dari evaluasi polinom-polinom $x^4 + x + 1$ dan $x^4 + x + 1$ atas lapangan $\mathbb{F}_{20}$.

4. Kesimpulan

Beradasarkan hasil yang didapatkan, dengan memanfaatkan Bahasa Pemrograman Python penelitian ini berhasil mengembangkan algoritma Python untuk menentukan apakah suatu polinomial atas lapangan hingga $\mathbb{Z}_p$ *irreducible* atau tidak. Serta didapatkan algoritma Python untuk menentukan FPB dan KPK dari dua polinomial atas lapangan hingga $\mathbb{F}_n$. Sehingga, untuk menentukan suatu polinom *irreducible* atau tidak atas lapangan hingga $\mathbb{Z}_p$ dapat dilakukan tanpa menggunakan perhitungan manual yang memakan waktu yang banyak. Begitu juga dalam penentuan FPB dan KPK dari dua polynomial atas lapangan hingga $\mathbb{F}_n$ dapat dilakukan tanpa menggunakan perhitungan manual, sehingga membutuhkan waktu komputasi yang lebih cepat.

Ucapan Terima Kasih

Kami berterima kasih kepada semua pihak yang telah membantu menyelesaikan penelitian ini, kami berterima kasih atas seluruh kontribusinya

DAFTAR PUSTAKA

[1] D. N. Zuraidah et al., "Menelisik Platform Digital dalam Teknologi Bahasa Pemrograman," *Teknois Journal: Jurnal Ilmiah Teknologi-Informasi & Sains*, vol. 11, no. 2, pp. 1-6, 2021. http://repository.uinsa.ac.id/id/eprint/1941

[2] S. Tribethran et al., "Pelatihan Pemrograman Dasar Python Dengan Memanfaatkan ChatGPT pada SMK Methodist 2 Palembang: Pelatihan Pemrograman Dasar Menggunakan Bahasa Python Kepada Para Siswa Kelas 10 SMK Methodist 2 Palembang," *Jumat Informatika: Jurnal Pengabdian Masyarakat*, vol. 4, no. 2, pp. 71-79, 2023. https://doi.org/10.32764/abdimasif.v4i2.3709

[3] H. Q. Dinh, B. T. Nguyen, A. K. Singh, and S. “Sriboonchitta, Hamming and Symbol-Pair Distances of Repeated-Root Constacyclic Codes of Prime Power Lengths Over $\mathbb{F}_{p^m} + u\mathbb{F}_{p^m}$,” *IEEE Communications Letters*, vol. 22, no. 12, pp. 2400-2403, Dec. 2018. https://doi.org/10.1109/LCOMM.2018.2868637

[4] Z. Zakirman, W. Gusta, and S. Dewimarni, "The Effectiveness of Euclid's Algorithm Method in Solving Problems about FPB," *Jurnal Cendekia: Jurnal Pendidikan Matematika*, vol. 6, no. 2, pp. 1603-1613, 2022. https://doi.org/10.31004/cendekia.v6i2.1331

[5] J. L. Butar-butar and F. Sinuhaji, “Faktorisasi Polinomial Square-Free dan bukan Square-Free atas Lapangan Hingga Zp,” *JTAM (Jurnal Teori dan Aplikasi Matematika)*, vol. 3, no. 2, pp. 132-142, 2019. https://doi.org/10.31764/jtam.v3i2.1079

[6] R. G. Nascimento, K. Fricke, and F. A. Viana, "A Tutorial on Solving Ordinary Differential Equations Using Python and Hybrid Physics-Informed Neural Network," *Engineering Applications of Artificial Intelligence*, vol. 96, no. 103996, 2020. https://doi.org/10.1016/j.engappai.2020.103996

[7] C. Rackauckas, "A Comparison Between Differential Equation Solver Suites in Matlab, R, Julia, Python, C, Mathematica, Maple, and Fortran," *Authorea Preprints*, 2023.

[8] M. Jamaluddin, *Algoritma dan Pemrograman Komputer dengan Python*, 2021.

[9] M. B. Tamam, "The Introduction to Python Programming Language for Students at Mtsn 4 Pandeglang School," *Journal of Community Service and Engagement*, vol. 2, no. 6, pp. 35-42, 2022. https://doi.org/10.9999/jocosae.v2i6.57

[10] A. Meurer et al., "SymPy: Symbolic Computing in Python," *PeerJ Computer Science*, vol. 3, no. e103, 2017.

[11] S. B. Nabijonovich and G. Najmiddin, "OPTIMIZING PYQT5 DEVELOPMENT WITH QT DESIGNER," *Web of Teachers: Inderscience Research*, vol. 2, no. 4, pp. 254-259, 2024. http://webofjournals.com/index.php/1/article/view/1224

[12] R. Munir, *Metode Numerik Revisi Kedua*. Bandung, Indonesia: Informatika Bandung, 2008.

[13] M. D. Raisinghania and R. S. Aggarwal, *Modern Algebra*. Raw Negar, New Delhi, India: Chad & Company LTD, 1980.

[14] J. B. Fraleigh and V. Katz, *A First Course in Abstract Algebra*. Pearson, 2004.

[15] S. Ling and C. Xing, *Coding Theory: A First Course*. Cambridge University Press, 2004.

[16] K. H. Rosen, *Elementary Number Theory*. London, U.K.: Pearson Education, 2011.