

Pengembangan Generator Bilangan Prima dengan Paradigma Pemrograman Berbasis Fungsi

Cyntia Kristina Sidauruk (122450023)¹, Patricia Leondrea Diajeng Putri (122450050)², Berliana Enda Putri (122450065)³, Feryadi Yulius (122450087)⁴, Naufal Fakhri (122450089)⁵

Program Studi Sains Data Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Way Huwi, Kec. Jatiagung, Kabupaten Lampung Selatan,
Lampung 35365

Email: ¹cyntia.122450023@student.itera.ac.id, ²patricia.122450050@student.itera.ac.id,
³berliana.122450065@student.itera.ac.id, ⁴feryadi.122450087@student.itera.ac.id,
⁵naufal.122450089@student.itera.ac.id

1. PENDAHULUAN

Pemakaian bilangan prima merupakan suatu permasalahan yang serius pada perkembangan ilmu komputer dan teori bilangan, sehingga dipandang perlu untuk menciptakan suatu tool yang dapat membangkitkan urutan bilangan prima yang teratur. Generator bilangan prima merupakan suatu tool, yang dapat membangkitkan pola pola urutan bilangan prima yang teratur,

Bilangan prima sangat bermanfaat untuk diterapkan sebagai basis dari penciptaan algoritma kriptografi kunci public, pada hash table, algoritma hash table paling baik apabila diinisialisasi dengan memakai bilangan prima agar dapat meminimumkan tabrakan (*collisions*) yang akan terjadi, di dalam menentukan pola-pola urutan bilangan prima yang ukurannya amat besar bukanlah suatu pekerjaan yang mudah, sehingga hal ini menjadi permasalahan yang harus dipecahkan. Sehingga dianggap perlu bagaimana cara tercepat untuk menghasilkan deret bilangan prima yang ukurannya amat besar. Penggunaan serial prosesor untuk pencarian deret bilangan prima yang ukurannya sangat besar kurang efisien mengingat diperlukannya waktu komputasi yang cukup panjang.

2. TUJUAN

1. Menciptakan metode langsung untuk menghasilkan barisan bilangan prima.
2. Meningkatkan produktivitas dan kecepatan dalam menghasilkan data berurutan yang ekstensif atau besar.
3. Memanfaatkan paradigma pemrograman fungsional untuk implementasi program.

3. METODE

3.1 Teori Bilangan

Matematika akan selalu dilibatkan dalam berbagai kegiatan di kehidupan sehari-hari, terutama matematika diskrit. Matematika diskrit sangat diperlukan untuk menjadi dasar untuk menguasai ilmu informatika, khususnya dalam algoritma. Dalam matematika diskrit terdapat berbagai macam teori yang dapat ditemui dalam kehidupan sehari-hari sehingga dapat pula diinterpretasikan dalam kehidupan nyata. Salah satu teori tersebut adalah teori bilangan yang merupakan teori dasar pada setiap operasi perhitungan (Hadi, G, 2017).

3.2 Bilangan Prima

Dari setiap ilmu yang ada, teori bilangan menjadi dasar atas segala perhitungan. Hal ini dikarenakan dari setiap teori memiliki minimal satu jenis bilangan, termasuk bilangan prima. Bilangan prima merupakan bilangan bulat positif yang lebih besar dari satu serta hanya memiliki dua faktor, yaitu bilangan 1 (satu) serta bilangan itu sendiri (A, Muftie, 2014). Pada umumnya, bilangan prima itu sendiri berbentuk :

$$Mp = 2p - 1$$

Banyak cara dan rumus yang dapat kita lakukan untuk mencari bilangan prima dengan mudah, khususnya untuk mencari bilangan prima dengan nominal bilangan yang terbilang cukup besar. Salah satu cara tersebut adalah dengan algoritma *Sieve of Atkin* (Pendekatan Saringan Atkin).

3.3 Sieve of Atkin

Sieve of Atkin merupakan sebuah algoritma modern yang digunakan untuk mencari semua bilangan prima dan bilangan bulat dengan menandai kelipatan kuadrat bilangan prima (Atkin, A. OL; Bernstein, DJ, 2004). Algoritma ini menyaring bilangan komposit dari daftar bilangan, tetapi algoritma ini bekerja dalam sisa modulo -60.

Sieve of Atkin memiliki algoritma sebagai berikut :

1. Buat daftar hasil, isi dengan 2, 3, dan 5.
2. Buat daftar saringan dengan entri untuk setiap bilangan bulat positif; semua entri dalam daftar ini awalnya harus ditandai bukan prima gabungan.
3. Untuk setiap entri nomor n dalam daftar saringan, dengan sisa modulo-enam puluh r :
 - Jika r adalah 1, 13, 17, 29, 37, 41, 49, atau 53, balikkan entri untuk setiap solusi yang mungkin menjadi $4x^2 + y^2 = n$. Jumlah operasi pembalikan sebagai rasio terhadap rentang pengayakan untuk langkah ini semakin dekat 415×860 (8 dalam pecahan berasal dari delapan modulo yang ditangani oleh kuadrat ini dan 60 karena Atkin menghitungnya berdasarkan bilangan genap roda modulo 60), yang menghasilkan pecahan sekitar 0,1117010721276....
 - Jika r adalah 7, 19, 31, atau 43, balikkan entri untuk setiap solusi yang mungkin menjadi $3x^2 + y^2 = n$. Jumlah operasi pembalikan sebagai rasio terhadap rentang pengayakan untuk langkah ini mendekati $0,12 \times 460$ ("4" dalam pecahan berasal dari empat modulo yang ditangani oleh kuadrat ini dan 60 karena Atkin menghitungnya berdasarkan bilangan genap roda modulo 60), yang menghasilkan pecahan sekitar 0,072551974569....
 - Jika r adalah 11, 23, 47, atau 59, balikkan entri untuk setiap solusi yang mungkin menjadi $3x^2 - y^2 = n$ ketika $x > y$. Jumlah operasi pembalikan sebagai rasio terhadap rentang pengayakan untuk langkah ini mendekati $1,92 \ln(0,5 + 1,5) \times 460$ ("4" dalam pecahan berasal dari empat modulo yang ditangani oleh kuadrat ini dan 60 karena Atkin menghitungnya berdasarkan bilangan genap roda modulo 60), yang menghasilkan pecahan sekitar 0,060827679704....
 - Jika r adalah sesuatu yang lain, abaikan saja.
5. Mulailah dengan angka terendah dalam daftar saringan.
6. Ambil bilangan berikutnya pada daftar saringan yang masih bertanda prima.
7. Sertakan nomor tersebut dalam daftar hasil.
8. Kuadratkan bilangan tersebut dan tandai semua kelipatan kuadrat tersebut sebagai bukan bilangan prima. Perhatikan bahwa kelipatan yang dapat difaktorkan dengan 2, 3, atau 5 tidak perlu diberi tanda, karena kelipatan ini akan diabaikan pada pencacahan akhir bilangan prima.
9. Ulangi langkah empat sampai tujuh. Jumlah total operasi untuk pengulangan menandai kuadrat bilangan prima sebagai rasio rentang pengayakan adalah jumlah invers kuadrat bilangan prima, yang mendekati fungsi zeta prima (2) dari 0,45224752004... dikurangi 122, 132, dan 152 untuk bilangan prima yang telah dihilangkan dengan roda, dan hasilnya dikalikan dengan 1660 untuk rasio pukulan roda per jarak; ini menghasilkan rasio sekitar 0,01363637571....

Dengan menambahkan rasio operasi di atas, algoritme di atas mengambil rasio operasi membalik/menandai yang konstan ke kisaran pengayakan sekitar 0,2587171021...; Dari implementasi algoritma yang sebenarnya, rasionya sekitar 0,25 untuk rentang pengayakan serendah 67.

4. PEMBAHASAN

Penelitian ini menghasilkan Python Script Generator yang dapat digunakan untuk mencari sebuah bilangan bulat dalam batas tertentu sebesar mungkin dengan paradigma pemrograman berbasis fungsi. Teknologi yang digunakan untuk membangun python Script Generator adalah bahasa pemrograman Python dengan menggunakan algoritma *Sieve of Atkins*. Kemudian script yang telah diprogram diharapkan bisa dipanggil sebagai modul pada command terminal ataupun program python lain.

4.1 PSEUDOCODE

Dari algoritma pencari bilangan prima Sieve of Atkins, bisa ditulis dalam bentuk pseudocode sebagai berikut:

```
PROGRAM: Sieve of Atkin
INPUT: limit (positive integer)
OUTPUT: P (list of prime numbers up to limit)
INITIALIZATION
- Create an empty list P to store prime numbers.
- Create a boolean sieve array of size limit + 1, initialized to all False.
- Add 2 and 3 to the P list (base cases for primes).
SIEVING
1. LOOP through x from 1 to the square root of limit (inclusive):
  a. LOOP through y from 1 to the square root of limit (inclusive):
    i. Calculate  $n1 = 4 * x^2 + y^2$ .
      - IF  $n1 \leq \text{limit}$  and  $(n1 \% 12 == 1 \text{ or } n1 \% 12 == 5)$ :
        - Invert the value of sieve[n1].
    ii. Calculate  $n2 = 3 * x^2 + y^2$ .
      - IF  $n2 \leq \text{limit}$  and  $n2 \% 12 == 7$ :
        - Invert the value of sieve[n2].
    iii. Calculate  $n3 = 3 * x^2 - y^2$  (only if  $x > y$ ).
      - IF  $n3 \leq \text{limit}$  and  $n3 \% 12 == 11$ :
        - Invert the value of sieve[n3].
2. LOOP through n from 5 to the square root of limit (inclusive):
  - IF sieve[n] is True (n is a potential prime):
    - LOOP through k from  $n^2$  to limit with a step of  $n^2$ :
      - SET sieve[k] to False (mark multiples of n as non-prime).
PRIME COLLECTION
- LOOP through p from 5 to limit (inclusive):
  - IF sieve[p] is True (p is prime):
    - APPEND p to the P list.
RETURN P (list of prime numbers)
```

Langkah selanjutnya adalah mengaplikasikan algoritma *Sieve of Atkins* pada pseudocode di atas menjadi kode fungsi dalam bahasa pemrograman Python.

Kode Program 1. Fungsi Generator Prime Number

```
def generate(limit):
    P = [2, 3]
    sieve = [False] * (limit + 1)
    for x in range(1, int(limit**0.5) + 1):
        for y in range(1, int(limit**0.5) + 1):
            n = 4*x**2 + y**2
            if n <= limit and (n % 12 == 1 or n % 12 == 5):
                sieve[n] = not sieve[n]
            n = 3*x**2 + y**2
            if n <= limit and n % 12 == 7:
                sieve[n] = not sieve[n]
            n = 3*x**2 - y**2
            if x > y and n <= limit and n % 12 == 11:
                sieve[n] = not sieve[n]
    for n in range(5, int(limit**0.5) + 1):
        if sieve[n]:
            for k in range(n**2, limit + 1, n**2):
                sieve[k] = False
    for p in range(5, limit + 1):
        if sieve[p]:
            P.append(p)
    return P
```

Pada Kode Program 1. merupakan fungsi generator yang meminta masukan berupa angka yang menjadi batas pencarian dan akan mengembalikan hasil berupa list bilangan dari 1 sampai angka batas pencarian yang merupakan bilangan prima. Fungsi bekerja dengan melakukan iterasi sebanyak batas yang ditentukan untuk mencari sebanyak mungkin bilangan yang tidak memenuhi satu pun kondisi yang diberikan. Ketika bilangan tersebut tidak memenuhinya, maka bilangan tersebut disimpan dalam list P dan merupakan bagian dari bilangan prima.

Kode Program 2. Fungsi Menentukan Bilangan Prima paling Besar

```
def biggest_prime(limit):  
    max_prime = max(generate(limit))  
    return max_prime
```

Setelah menentukan fungsi untuk menghasilkan list bilangan prima kita membuat fungsi untuk menentukan mana bilangan prima paling besar dalam list yang dihasilkan tersebut menggunakan fungsi nested (fungsi di dalam fungsi). Fungsi `biggest_prime` di atas bekerja dengan parameter input sama dengan fungsi `generate()` sebelumnya, kemudian dilakukan pemanggilan fungsi generator sebagai parameter fungsi built-in Python, yaitu `max()`. Fungsi `max` berguna untuk mencari nilai terbesar dalam sebuah iterable seperti list, tuple, dictionary, dll. Fungsi `biggest_prime()` mengembalikan bilangan terbesar yang pada batas tersebut.

Penggunaannya dalam Terminal Desktop Windows

Setelah mendapatkan fungsi mencari bilangan prima dan fungsi yang mencari bilangan prima terbesar pada rentang tertentu, selanjutnya kita akan menggunakannya sebagai modul Python yang dapat dipanggil kembali. Setelah menuliskan kodenya dalam bahasa python dan disimpan dalam format .py dengan nama `PrimeNumberGenerator.py` bisa di panggil lewat terminal desktop.

```
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\HP\14\Downloads\COLLAGE\SEMESTER 4\PPF\PRAKTIKUM 3>python  
Python 3.11.8 (tags/v3.11.8:db85d51, Feb 6 2024, 22:03:32) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> |
```

Gambar 1.

Pada Gambar. 1 di atas, pertama jalankan terminal pada directory file `PrimeNumberGenerator.py` berada, kemudian jalankan `python.exe`. Setelah terminal dijalankan dalam bahasa Python, lakukan import modul `PrimeNumberGenerator`.

```
import PrimeNumberGenerator as prime
```

Kemudian panggil fungsi `generate()` dengan nilai batas 100 sehingga didapat output berupa list bilangan prima dari 2 sampai 97.

```
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\HP\14\Downloads\COLLAGE\SEMESTER 4\PPF\PRAKTIKUM 3>python  
Python 3.11.8 (tags/v3.11.8:db85d51, Feb 6 2024, 22:03:32) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import PrimeNumberGenerator as prime  
File "<stdin>", line 1  
    import PrimeNumberGenerator as prime  
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  
SyntaxError: invalid syntax  
>>> import PrimeNumberGenerator as prime  
>>> prime.generate(100)  
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]  
>>> |
```

Gambar 2.

Pada Gambar 2. diatas, setelah modul diimport dan memanggil `prime.generate(100)`, yang memberi argumen 100 artinya pemanggilan bilangan prima tidak lewat dari 100. Kita juga bisa mencari bilangan prima sampai batas yang lebih besar, misalnya 1.000 sampai 100.000.000

```

Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP 14\Downloads\COLLAGE\SEMESTER 4\DPB\PRAKTIKUM 3>python
Python 3.11.8 (tags/v3.11.8:db853f3, Feb. 6 2024, 22:03:32) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import PrimeNumberGenerator as prime
>>> File "testdine", line 1
>>>     Import PrimeNumberGenerator as prime
>>> SyntaxError: invalid syntax
>>> import PrimeNumberGenerator as prime
>>> prime.generate(100)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
>>> prime.generate(1000)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 713, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]

```

Gambar 3.

Pada Gambar 3. diatas, terlihat bahwa argumen diberikan yaitu 1000 yang artinya bilangan prima dipanggil dengan batas 1000 dan didapat output 2 sampai dengan 997.

```

83, 991, 997]
>>> prime.generate(1000000)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 713, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997, 1009, 1013, 1019, 1021, 1031, 1033, 1039, 1043, 1049, 1051, 1057, 1061, 1063, 1067, 1069, 1073, 1079, 1081, 1087, 1091, 1093, 1097, 1103, 1107, 1111, 1117, 1123, 1127, 1129, 1133, 1139, 1143, 1147, 1151, 1153, 1157, 1163, 1167, 1171, 1181, 1187, 1193, 1201, 1213, 1217, 1223, 1229, 1231, 1237, 1243, 1249, 1253, 1259, 1261, 1267, 1271, 1273, 1277, 1283, 1289, 1291, 1297, 1301, 1303, 1307, 1319, 1321, 1327, 1331, 1337, 1343, 1349, 1351, 1357, 1361, 1367, 1373, 1381, 1387, 1391, 1397, 1403, 1409, 1411, 1417, 1423, 1429, 1433, 1439, 1443, 1449, 1451, 1457, 1463, 1469, 1471, 1481, 1483, 1487, 1489, 1493, 1499, 1511, 1523, 1531, 1543, 1549, 1553, 1559, 1567, 1571, 1579, 1583, 1597, 1601, 1607, 1609, 1613, 1619, 1621, 1627, 1631, 1637, 1643, 1649, 1653, 1657, 1661, 1669, 1673, 1679, 1681, 1687, 1691, 1693, 1697, 1699, 1703, 1709, 1711, 1717, 1721, 1723, 1729, 1733, 1741, 1747, 1753, 1759, 1763, 1769, 1771, 1777, 1783, 1789, 1793, 1799, 1801, 1811, 1813, 1817, 1823, 1829, 1831, 1837, 1841, 1843, 1847, 1853, 1859, 1861, 1867, 1871, 1873, 1877, 1879, 1883, 1889, 1891, 1897, 1901, 1903, 1907, 1913, 1931, 1933, 1949, 1951, 1957, 1973, 1979, 1987, 1993, 1997, 1999, 2003, 2011, 2017, 2027, 2029, 2039, 2043, 2053, 2059, 2063, 2069, 2071, 2081, 2083, 2087, 2089, 2099, 2111, 2113, 2129, 2131, 2137, 2141, 2143, 2149, 2153, 2161, 2179, 2203, 2207, 2213, 2237, 2239, 2243, 2251, 2267, 2269, 2273, 2281, 2287, 2293, 2297, 2309, 2311, 2333, 2339, 2341, 2347, 2351, 2357, 2371, 2377, 2381, 2383, 2389, 2393, 2399, 2411, 2417, 2423, 2437, 2441, 2447, 2459, 2467, 2473, 2477, 2503, 2521, 2531, 2539, 2543, 2549, 2551, 2557, 2579, 2591, 2593, 2609, 2617, 2621, 2633, 2647, 2657, 2659, 2663, 2671, 2677, 2683, 2687, 2699, 2701, 2711, 2713, 2719, 2729, 2731, 2741, 2749, 2753, 2767, 2777, 2789, 2791, 2797, 2801, 2803, 2807, 2819, 2833, 2837, 2839, 2843, 2857, 2859, 2861, 2867, 2869, 2873, 2879, 2881, 2887, 2889, 2893, 2897, 2903, 2909, 2917, 2927, 2939, 2953, 2957, 2963, 2969, 2971, 2999, 3001, 3011, 3013, 3019, 3023, 3027, 3031, 3037, 3041, 3043, 3049, 3053, 3059, 3061, 3067, 3071, 3073, 3079, 3083, 3089, 3091, 3097, 3103, 3107, 3109, 3113, 3121, 3137, 3139, 3143, 3147, 3151, 3157, 3163, 3167, 3169, 3181, 3187, 3191, 3193, 3197, 3203, 3207, 3211, 3217, 3221, 3229, 3231, 3237, 3259, 3271, 3299, 3301, 3307, 3313, 3319, 3323, 3329, 3331, 3337, 3343, 3347, 3359, 3361, 3371, 3373, 3379, 3381, 3383, 3389, 3391, 3397, 3401, 3403, 3407, 3409, 3413, 3419, 3421, 3427, 3431, 3433, 3437, 3439, 3443, 3449, 3451, 3457, 3461, 3463, 3467, 3469, 3473, 3479, 3481, 3487, 3491, 3493, 3497, 3503, 3509, 3511, 3517, 3523, 3527, 3529, 3533, 3539, 3541, 3547, 3551, 3553, 3559, 3563, 3569, 3571, 3577, 3581, 3583, 3589, 3593, 3599, 3607, 3613, 3617, 3623, 3631, 3637, 3643, 3659, 3671, 3673, 3677, 3691, 3697, 3701, 3709, 3719, 3727, 3733, 3739, 3743, 3749, 3751, 3757, 3761, 3763, 3769, 3773, 3779, 3781, 3787, 3791, 3793, 3799, 3803, 3809, 3811, 3817, 3821, 3823, 3829, 3833, 3839, 3841, 3847, 3851, 3853, 3859, 3863, 3869, 3871, 3877, 3881, 3883, 3889, 3893, 3899, 3901, 3907, 3911, 3913, 3919, 3923, 3929, 3931, 3937, 3941, 3943, 3947, 3953, 3959, 3961, 3967, 3971, 3973, 3979, 3983, 3989, 3991, 3997, 4001, 4003, 4009, 4013, 4019, 4021, 4027, 4031, 4033, 4039, 4043, 4049, 4051, 4057, 4061, 4063, 4069, 4073, 4079, 4081, 4087, 4091, 4093, 4099, 4103, 4109, 4111, 4117, 4121, 4123, 4129, 4133, 4139, 4143, 4149, 4153, 4157, 4159, 4177, 4201, 4211, 4217, 4219, 4229, 4231, 4241, 4243, 4253, 4259, 4261, 4271, 4273, 4283, 4289, 4297, 4307, 4317, 4327, 4339, 4349, 4357, 4363, 4369, 4373, 4381, 4387, 4393, 4397, 4403, 4409, 4411, 4417, 4421, 4423, 4429, 4433, 4439, 4441, 4447, 4451, 4453, 4459, 4463, 4469, 4471, 4477, 4481, 4483, 4489, 4493, 4499, 4501, 4503, 4507, 4511, 4513, 4519, 4523, 4529, 4531, 4537, 4543, 4549, 4553, 4559, 4561, 4567, 4569, 4573, 4577, 4581, 4583, 4589, 4591, 4597, 4603, 4607, 4609, 4613, 4617, 4621, 4623, 4629, 4633, 4637, 4639, 4643, 4649, 4651, 4657, 4661, 4663, 4669, 4673, 4679, 4681, 4687, 4691, 4693, 4697, 4703, 4709, 4711, 4717, 4721, 4723, 4729, 4733, 4739, 4743, 4749, 4753, 4759, 4763, 4769, 4771, 4777, 4781, 4783, 4789, 4793, 4799, 4801, 4803, 4807, 4811, 4813, 4817, 4821, 4823, 4829, 4831, 4837, 4841, 4843, 4849, 4851, 4853, 4859, 4861, 4867, 4871, 4873, 4879, 4881, 4883, 4889, 4891, 4897, 4901, 4903, 4909, 4911, 4913, 4919, 4921, 4927, 4931, 4933, 4937, 4943, 4949, 4951, 4957, 4963, 4969, 4973, 4987, 4993, 4999, 5003, 5009, 5011, 5017, 5021, 5023, 5029, 5033, 5039, 5041, 5047, 5051, 5053, 5059, 5063, 5069, 5071, 5077, 5081, 5083, 5089, 5093, 5099, 5101, 5107, 5111, 5113, 5119, 5123, 5129, 5131, 5137, 5141, 5143, 5147, 5153, 5159, 5161, 5167, 5171, 5173, 5179, 5183, 5189, 5197, 5203, 5209, 5211, 5217, 5221, 5223, 5229, 5231, 5233, 5237, 5241, 5243, 5249, 5251, 5257, 5259, 5263, 5269, 5271, 5273, 5279, 5281, 5283, 5289, 5293, 5299, 5303, 5309, 5311, 5317, 5321, 5323, 5329, 5331, 5337, 5341, 5343, 5347, 5351, 5353, 5359, 5363, 5369, 5371, 5377, 5381, 5383, 5389, 5393, 5399, 5401, 5403, 5407, 5413, 5417, 5419, 5423, 5427, 5431, 5433, 5437, 5441, 5443, 5449, 5451, 5457, 5459, 5463, 5469, 5471, 5473, 5479, 5481, 5483, 5489, 5493, 5499, 5501, 5507, 5511, 5513, 5519, 5521, 5527, 5531, 5537, 5541, 5543, 5549, 5551, 5557, 5561, 5563, 5569, 5571, 5577, 5581, 5583, 5589, 5593, 5599, 5601, 5607, 5611, 5613, 5619, 5621, 5623, 5629, 5631, 5637, 5641, 5643, 5649, 5651, 5653, 5657, 5659, 5669, 5683, 5689, 5693, 5701, 5711, 5717, 5737, 5741, 5743, 5749, 5779, 5783, 5791, 5801, 5807, 5813, 5821, 5

```

Gambar 4.

Pada Gambar 4. diatas, dilakukan pemanggilan berikutnya dengan memberikan argumen 1.000.000 maka akan mengeluarkan output bilangan prima dari 2 sampai batas yang telah ditentukan dan tidak akan melewati 1.000.000 di dalam terminal.

Kemudian untuk menentukan mana bilangan prima paling besar dalam batas tersebut kita bisa memanfaatkan fungsi `biggest_prime()`.

```

>>> prime.biggest_prime(1000000)
999983
>>>

```

Gambar 5.

Pada Gambar 5. diatas, dengan menggunakan fungsi `biggest_prime`, akan digunakan untuk mencari bilangan prima terbesar dengan batas akhir sesuai argumen yang dimasukkan sebesar 1.000.000 dan didapat output yaitu 999983.

5. KESIMPULAN

Tujuan dari jurnal pratikum ini adalah untuk mengembangkan pembangkit bilangan prima dengan menggunakan paradigma pemrograman berbasis fungsi. Metode yang digunakan adalah algoritma saringan Atkin, yang merupakan algoritma modern yang efisien untuk mencari bilangan prima. Dalam bidang pemrograman fungsional, penelitian ini berhasil menciptakan script generator Python yang dapat menghasilkan barisan bilangan prima dalam batasan tertentu. Pendekatan ini memanfaatkan paradigma pemrograman fungsional untuk meningkatkan produktivitas dan kecepatan dalam menghasilkan data berkelanjutan berskala besar. Penelitian ini memperkenalkan dua fungsi utama dengan mengimplementasikan algoritma saringan Atkin dengan Python.

Fungsi `generate()` menghasilkan daftar bilangan prima dalam rentang tertentu, dan fungsi `most_prime()` menentukan nilai maksimum. Bilangan prima dalam batas ini Modul Python yang dibuat memungkinkan anda dengan mudah memperoleh barisan bilangan prima dalam rentang yang berbeda dan juga menemukan bilangan prima terbesar dalam batas tertentu. Oleh karena itu, jurnal pratikum ini berkontribusi pada pengembangan alat yang dapat digunakan dalam berbagai aplikasi, termasuk kriptografi, hashing, dan bidang komputasi numerik lainnya.

REFERENSI

- G. Hadi, "Apa yang dimaksud dengan penelitian eksperimental," 8 Agustus 2017. [Online]. Available:<https://www.dictio.id/t/apa-yang-dimaksud-dengan-penelitian-eksperimental/8896>. [Accessed 3 November 2018].
- Muftic A. Matematika Alam Semesta Kodetifikasi Bilangan Prima dalam Al-Qur'an. Bandung: PT. Kiblat Utama; 2014
- A.OL Atkin, DJ Bernstein, Saringan prima menggunakan bentuk kuadrat biner, Matematika. Komp. 73(2004), 1023-1030.
- HIDAYAT, I. R. (2010). Generator bilangan prima dengan menggunakan struktur bit-array (Doctoral dissertation, Universitas Gadjah Mada).
- William, Fu. (2019). Analisis Perbandingan Kompleksitas Waktu Algoritma Sieve of Sundaram dengan Algoritma Pencarian Bilangan Prima Lainnya. Institut Teknologi Bandung