

LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN
“POST TEST PRAKTIKUM 4: PERULANGAN
(ITERATIF-REKURSIF)”

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Algoritma Pemrograman yang
di ampu oleh:

Dr. Ardiansyah S.T., M.Cs



Disusun Oleh:

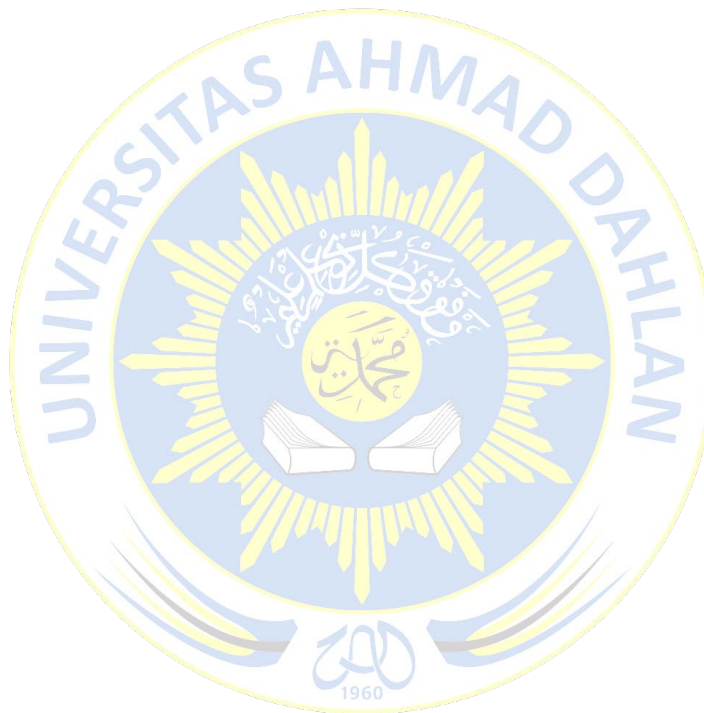
Mohammad Farid Hendianto 2200018401

A / Jumat 13.30 Lab. Jaringan

PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

DAFTAR SOAL

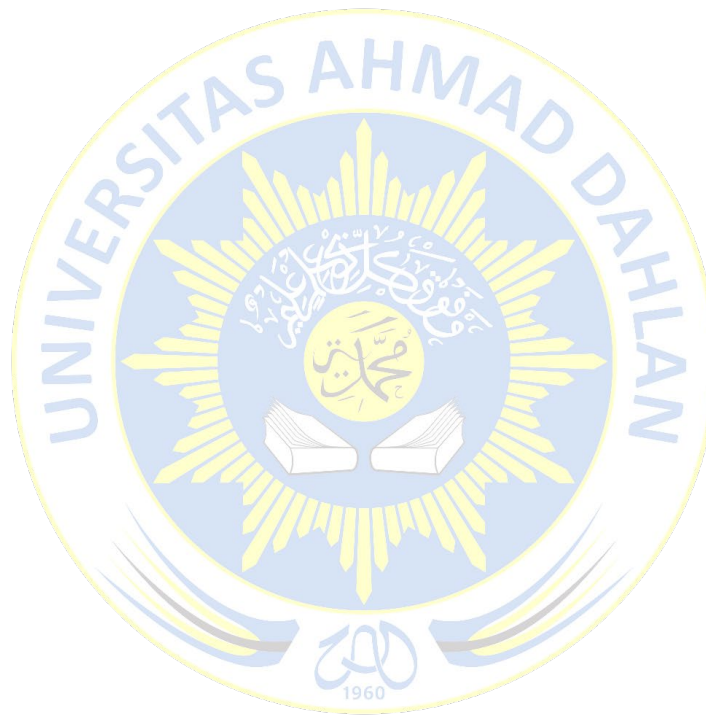
1. Buat lah flowchart untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 secara iteratif kemudian dirubah ke rekursif. 3
2. Seperti nomor 1, gunakan subprogam dalam flowchart untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 secara iteratif kemudian dirubah ke rekursif.
10
3. Konversikan hasil dari flowchart nomor 1 dan 2 menjadi progam C++. 23

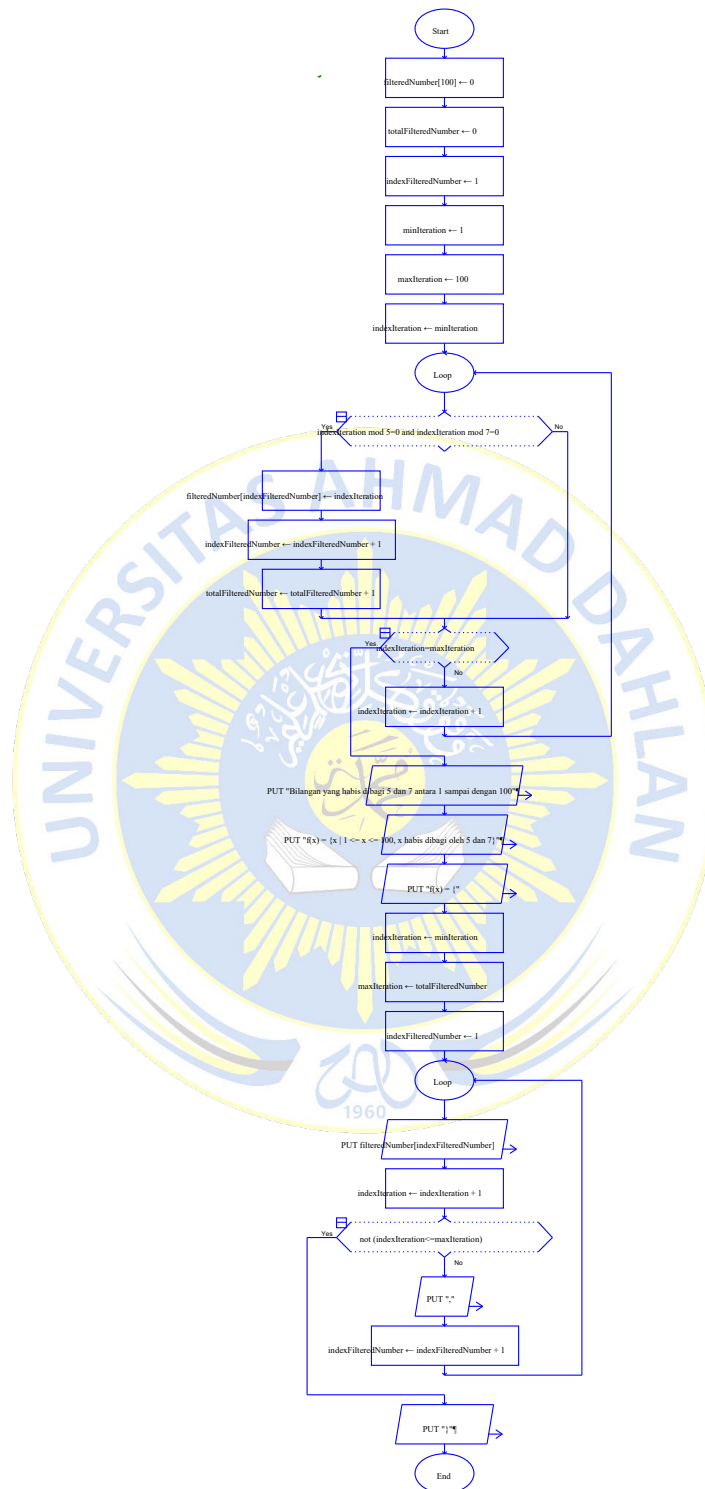


1. Buat lah flowchart untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 secara iteratif kemudian dirubah ke rekursif.

Jawab:

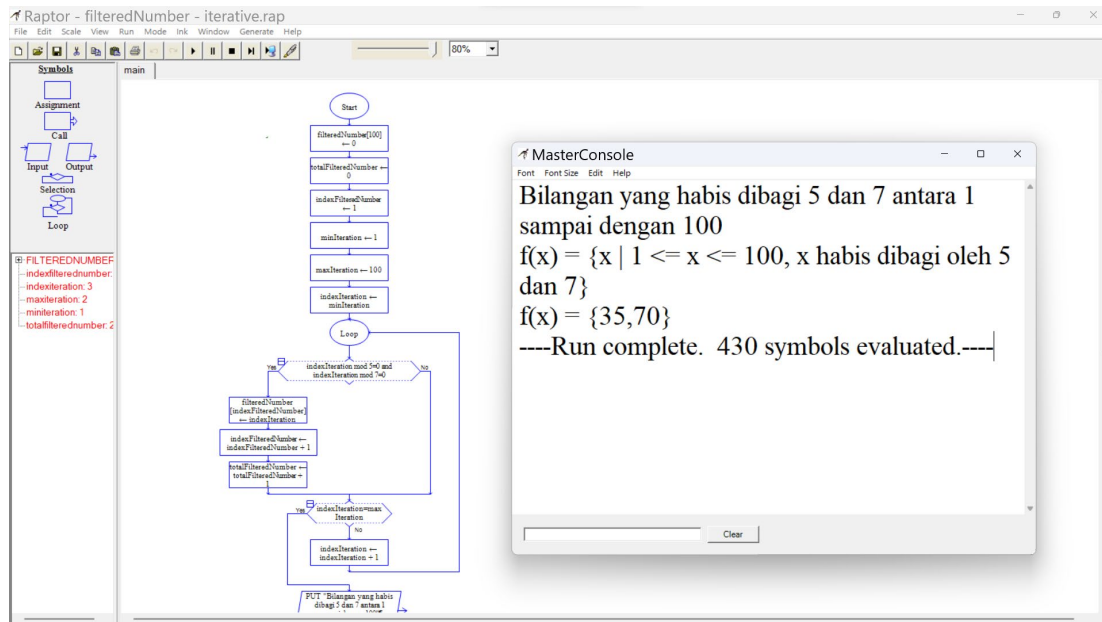
Berikut adalah flowchart untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 secara iteratif.





Gambar 1 Iteratif. (Sumber: Penulis)

Berikut adalah outputnya:



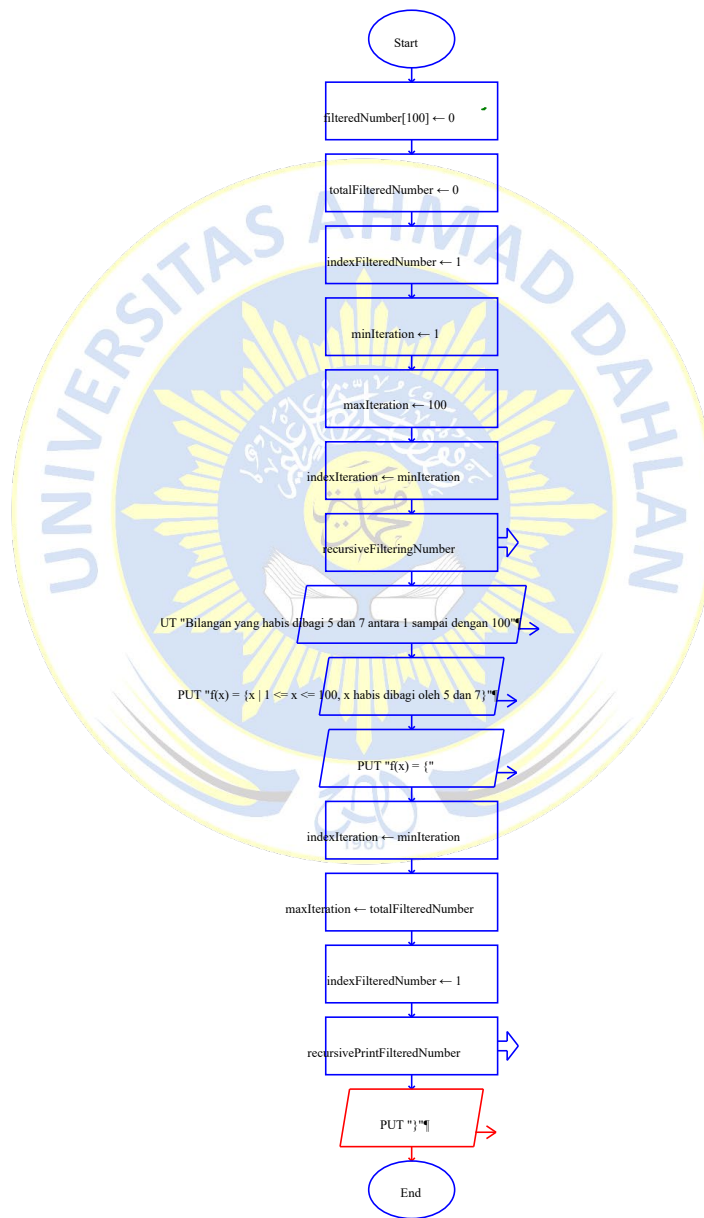
Gambar 2 Output iteratif. (Sumber: Penulis)

Flowchart yang disajikan di atas adalah program untuk mencari bilangan bulat antara 1 sampai dengan 100 yang habis dibagi oleh 5 dan 7. Berikut adalah penjelasan alur flowchart tersebut secara lengkap:

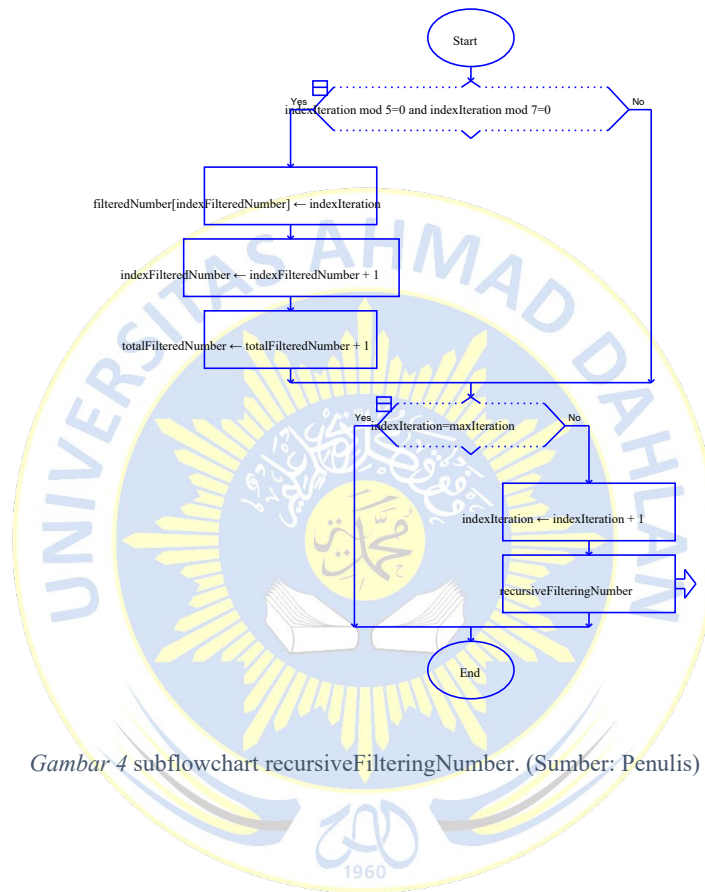
- 1) Program dimulai dengan menentukan variabel-variabel yang akan digunakan dalam program, seperti miniteration, indexfilterednumber, maxiteration, indexiteration, totalfilterednumber, dan filterednumber.
- 2) Kemudian, program menginisialisasi array filterednumber dengan nilai 0 dan variabel totalFilteredNumber dengan nilai 0.
- 3) Program akan memulai iterasi dari minIteration (nilai awal = 1) hingga maxIteration (nilai awal = 100).
- 4) Pada setiap iterasi, program akan mengecek apakah nilai indexIteration habis dibagi oleh 5 dan 7 (dengan menggunakan operator mod). Jika ya, maka nilai indexIteration akan ditambahkan ke dalam array filterednumber dan indexFilteredNumber serta totalFilteredNumber akan di-increment.
- 5) Setelah iterasi selesai, program akan menampilkan pesan "Bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100" dan " $f(x) = \{x \mid 1 \leq x \leq 100, x \text{ habis dibagi oleh } 5 \text{ dan } 7\}$ ".
- 6) Selanjutnya, program akan menampilkan kumpulan bilangan yang ditemukan dengan format $f(x) = \{\text{bilangan1}, \text{bilangan2}, \dots, \text{bilanganN}\}$.
- 7) Program selesai.

Demikianlah penjelasan alur flowchart untuk program filteredNumber - iterative dengan menggunakan RAPTOR. Flowchart ini bertujuan untuk membantu pengguna memahami bagaimana program menghasilkan kumpulan bilangan bulat antara 1 sampai dengan 100 yang habis dibagi oleh 5 dan 7.

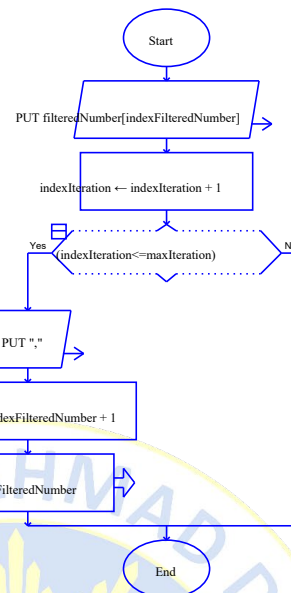
Berikut adalah flowchart untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 secara rekursif.



Gambar 3 Rekursif flowchart utama. (Sumber: Penulis)



Gambar 4 subflowchart recursiveFilteringNumber. (Sumber: Penulis)



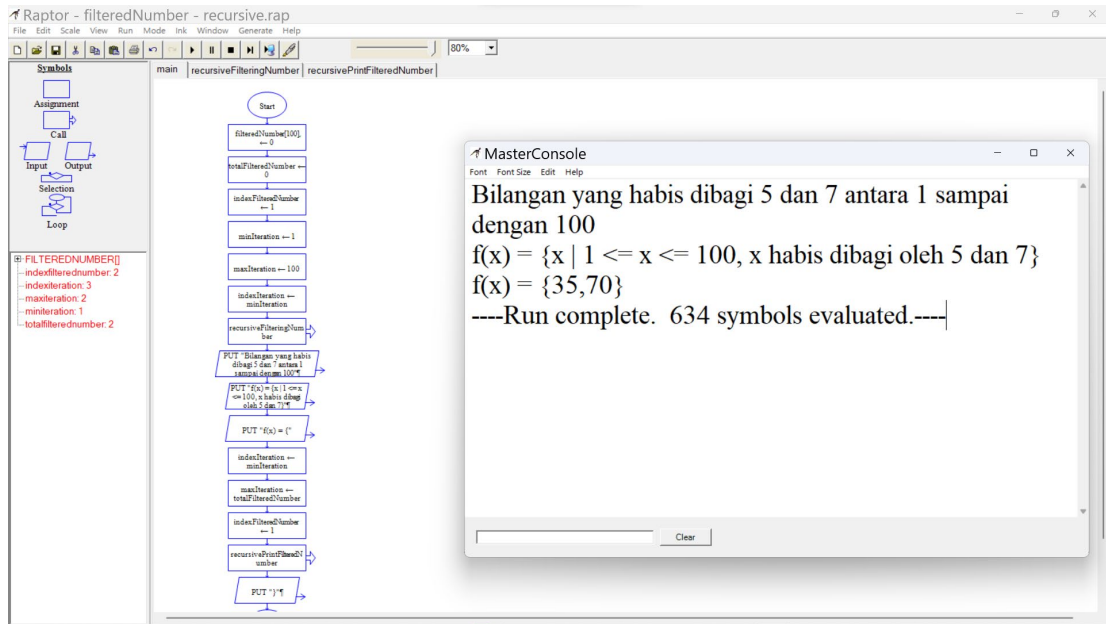
Gambar 5 subflowchart recursivePrintFilteredNumber. (Sumber: Penulis)

Flowchart rekursif yang disajikan dalam Raptor tersebut adalah untuk melakukan filtering terhadap bilangan antara 1 hingga 100 yang habis dibagi 5 dan 7, lalu mencetak hasilnya. Untuk flowchart recursive, harus memakai prosedur/fungsi agar bisa memanggil diri sendirinya. Berikut penjelasan mengenai flowchart tersebut:

- Algoritma dimulai dengan inisialisasi array "filteredNumber" dengan nilai 0 dan variabel "indexFilteredNumber" dengan nilai 1, yang akan digunakan pada proses filtering bilangan.
- Kemudian variabel "minIteration" diinisialisasi dengan nilai 1 dan "maxIteration" dengan nilai 100, yang akan digunakan sebagai batas iterasi.
- Selanjutnya variabel "indexIteration" diinisialisasi dengan nilai "minIteration" dan prosedur "recursiveFilteringNumber" dipanggil dengan menggunakan perintah "call recursiveFilteringNumber".
- Pada prosedur "recursiveFilteringNumber", dilakukan pengecekan apakah bilangan tersebut habis dibagi 5 dan 7, jika iya maka bilangan tersebut disimpan pada array "filteredNumber" pada indeks ke-"indexFilteredNumber" dan nilai variabel "totalFilteredNumber" akan ditambah 1. Variabel "indexFilteredNumber" juga akan bertambah 1.
- Setelah itu, prosedur "recursivePrintFilteredNumber" dipanggil dengan menggunakan perintah "call recursivePrintFilteredNumber".

- Pada prosedur "recursivePrintFilteredNumber", pertama-tama akan dicetak bilangan pada indeks ke-"indexFilteredNumber" dari array "filteredNumber". Kemudian nilai variabel "indexFilteredNumber" akan bertambah 1.
- Selanjutnya dilakukan pengecekan apakah nilai variabel "indexFilteredNumber" sudah melebihi nilai variabel "totalFilteredNumber". Jika iya, maka akan dicetak tanda "," dan nilai variabel "indexFilteredNumber" akan bertambah 1 lagi.
- Kemudian dilakukan pemanggilan rekursif terhadap prosedur "recursivePrintFilteredNumber", sehingga proses pencetakan bilangan yang habis dibagi 5 dan 7 akan dilakukan secara berulang-ulang sampai semua bilangan pada array "filteredNumber" tercetak.
- Selanjutnya dilakukan pencetakan string "Bilangan yang habis dibagi 5 dan 7 antara 1 sampai 100 ", " $f(x) = \{x | 1 \leq x \leq 100, x \text{ habis dibagi } 5 \text{ dan } 7\}$ ", dan " $f(x) = \{\}$ ".
- Dilakukan looping untuk mencetak setiap bilangan pada array "filteredNumber", dengan menggunakan perintah "do".
- Pada setiap perulangan, dicetak nilai pada indeks ke-"indexFilteredNumber" dari array "filteredNumber", kemudian nilai variabel "indexFilteredNumber" bertambah 1.
- Kemudian dilakukan pengecekan apakah nilai variabel "indexIteration" sudah melebihi nilai variabel "maxIteration". Jika iya, maka akan dicetak tanda "," dan nilai variabel "indexFilteredNumber" akan bertambah 1 lagi.
- Setelah looping selesai, nilai variabel "indexIteration" diinisialisasi kembali dengan nilai 1, dan nilai variabel "maxIteration" diinisialisasi dengan nilai "totalFilteredNumber".
- Nilai variabel "indexFilteredNumber" juga diinisialisasi kembali dengan nilai 1, dan prosedur "recursiveFilteringNumber" dipanggil kembali.
- Setelah itu, prosedur "recursivePrintFilteredNumber" dipanggil kembali, sehingga proses pencetakan bilangan yang habis dibagi 5 dan 7 dilakukan lagi dengan bilangan hasil filtering yang baru.
- Terakhir, dicetak tanda "}" untuk menutup daftar bilangan hasil filtering.

Berikut adalah outputnya:

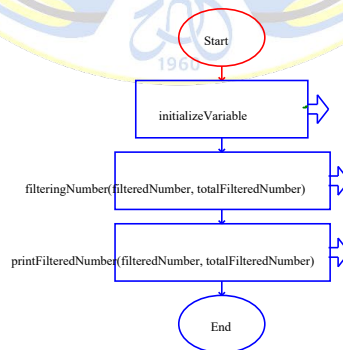


Gambar 6 Output rekursif. (Sumber: Penulis)

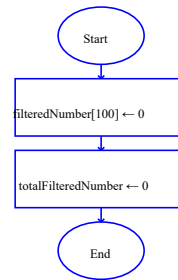
2. Seperti nomor 1, gunakan subprogram dalam flowchart untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 secara iteratif kemudian dirubah ke rekursif.

Jawab:

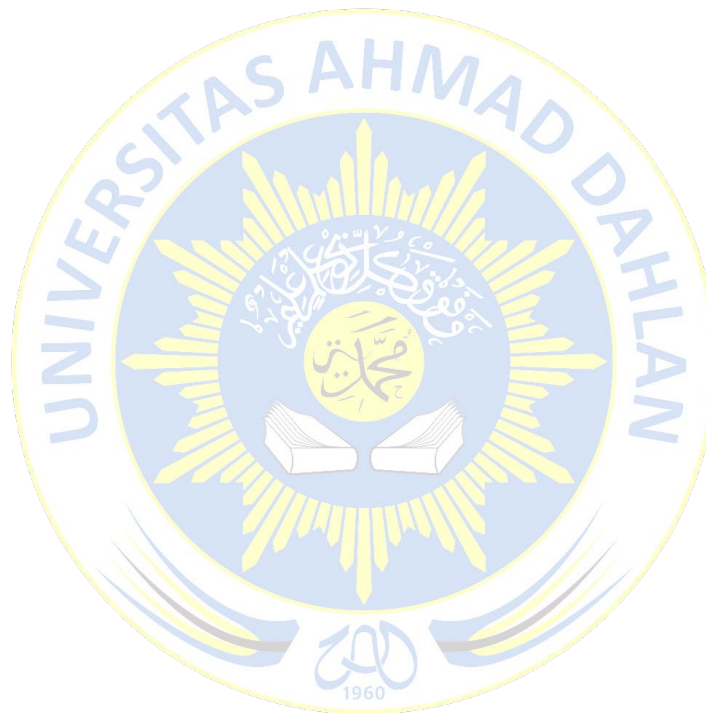
Berikut adalah flowchart untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 secara iteratif dengan subprogram.

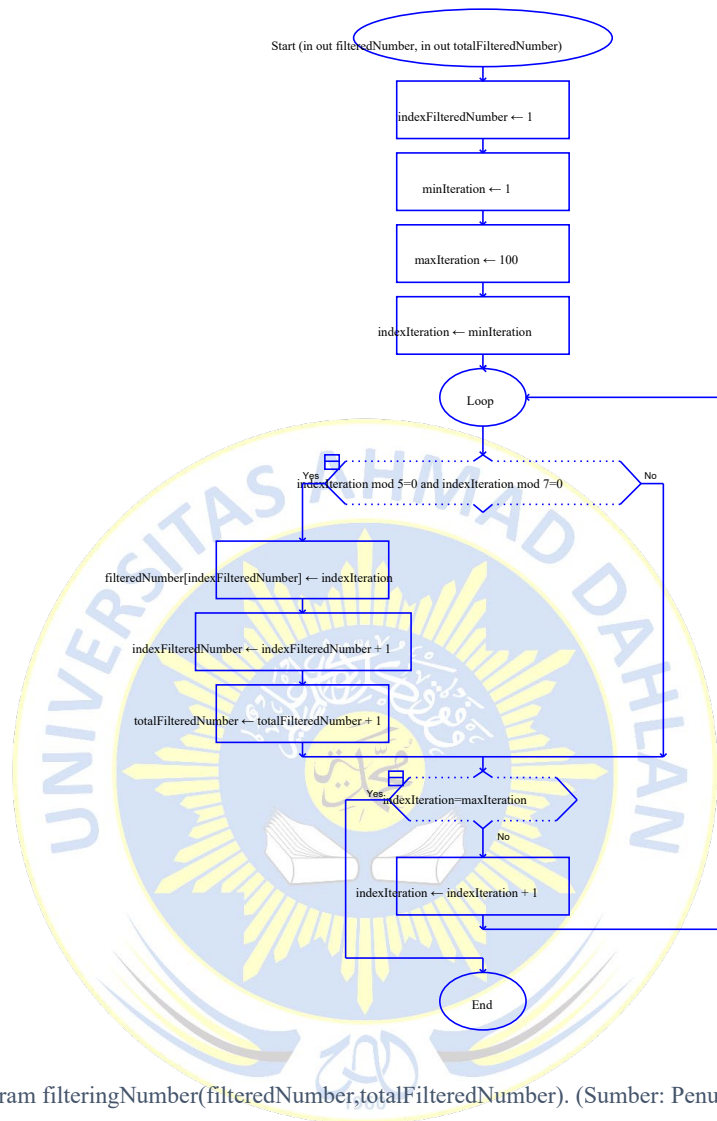


Gambar 7 Fungsi Main (Utama) Iteratif dengan subProgram. (Sumber: Penulis)

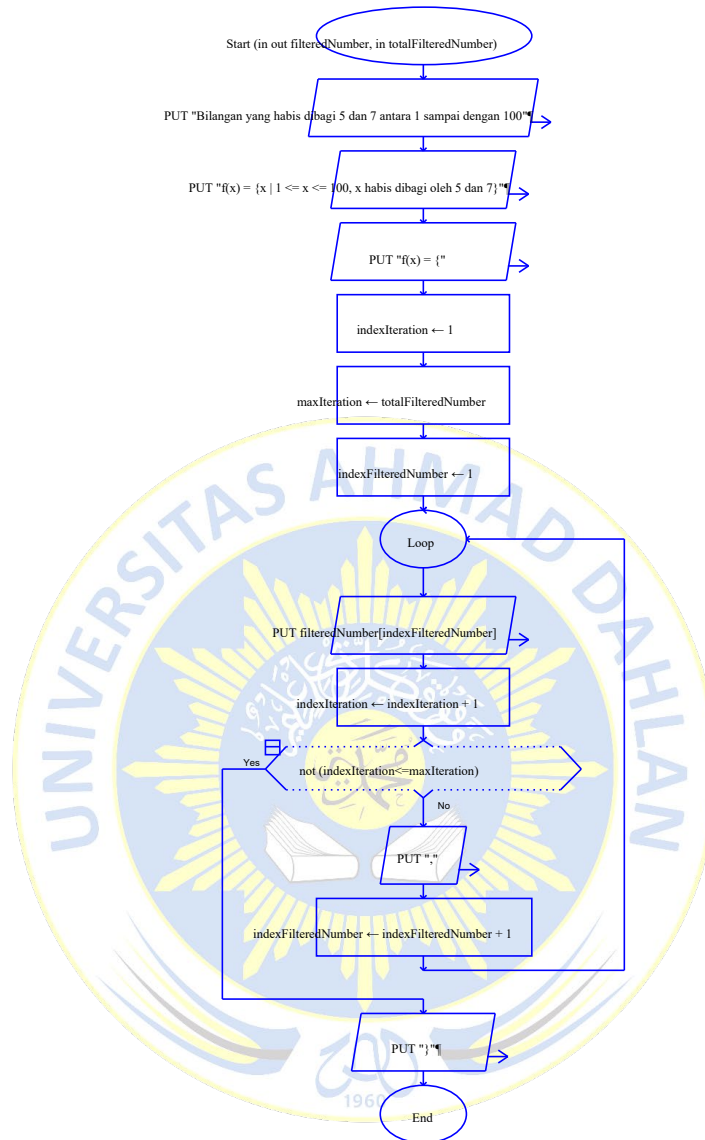


Gambar 8 SubChart initializeVariable (Sumber: Penulis)





Gambar 9 Subprogram filteringNumber(filteredNumber,totalFilteredNumber). (Sumber: Penulis)



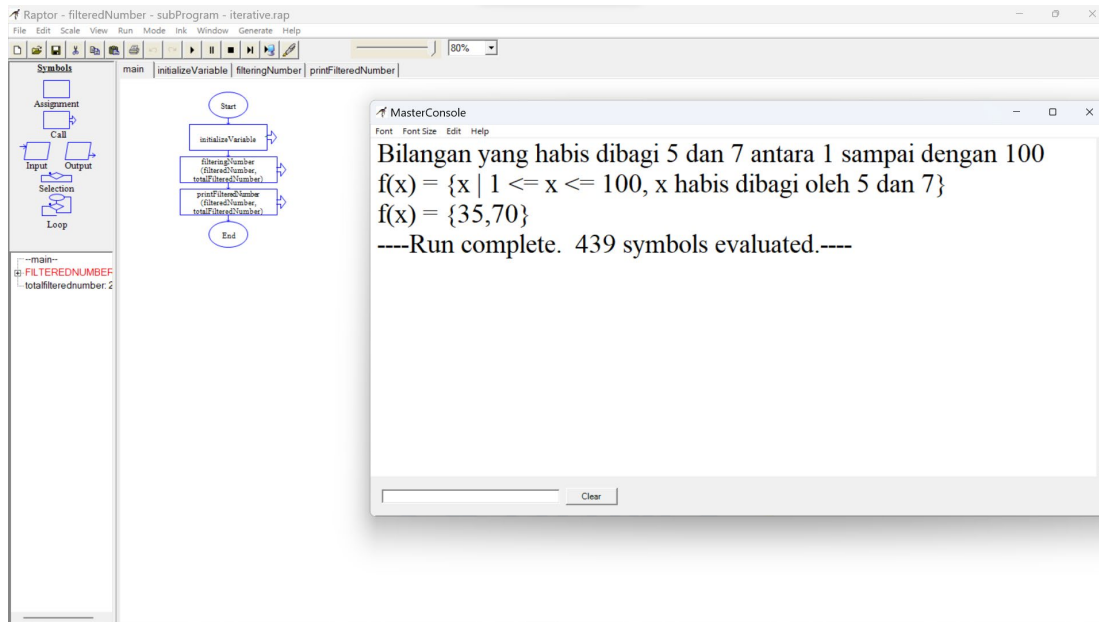
Gambar 10 SubProgramprintFilteredNumber(filteredNumber,totalFilteredNumber). (Sumber: Penulis)

Berikut adalah penjelasan alur program dalam bahasa flowchart:

- 1) Mulai
- 2) Deklarasi variabel dan array filteredNumber dan totalFilteredNumber.
- 3) Panggil prosedur filteringNumber dengan parameter filteredNumber dan totalFilteredNumber.

- i. Inisialisasi variabel: `indexFilteredNumber`, `minIteration`, `maxIteration`, dan `indexIteration`.
- ii. Lakukan loop:
 - a. Jika `indexIteration` habis dibagi 5 dan 7, lakukan:
 - a. Simpan `indexIteration` ke dalam `filteredNumber` pada posisi `indexFilteredNumber`.
 - b. Tambahkan 1 ke `indexFilteredNumber`.
 - c. Tambahkan 1 ke `totalFilteredNumber`.
 - b. Jika tidak, lanjutkan.
 - c. Keluar dari loop jika `indexIteration` sama dengan `maxIteration`.
 - d. Tambahkan 1 ke `indexIteration`.
- 4) Panggil prosedur `printFilteredNumber` dengan parameter `filteredNumber` dan `totalFilteredNumber`.
 - i. Cetak informasi tentang bilangan yang habis dibagi 5 dan 7 antara 1 dan 100.
 - ii. Cetak `"f(x) = {"`
 - iii. Inisialisasi variabel: `indexIteration`, `maxIteration`, dan `indexFilteredNumber`.
 - iv. Lakukan loop:
 - a. Cetak elemen `filteredNumber` pada posisi `indexFilteredNumber`.
 - b. Tambahkan 1 ke `indexIteration`.
 - c. Keluar dari loop jika `indexIteration` lebih besar dari `maxIteration`.
 - d. Cetak `","` dan tambahkan 1 ke `indexFilteredNumber`.
 - v. Cetak `"}"`
- 5) Selesai

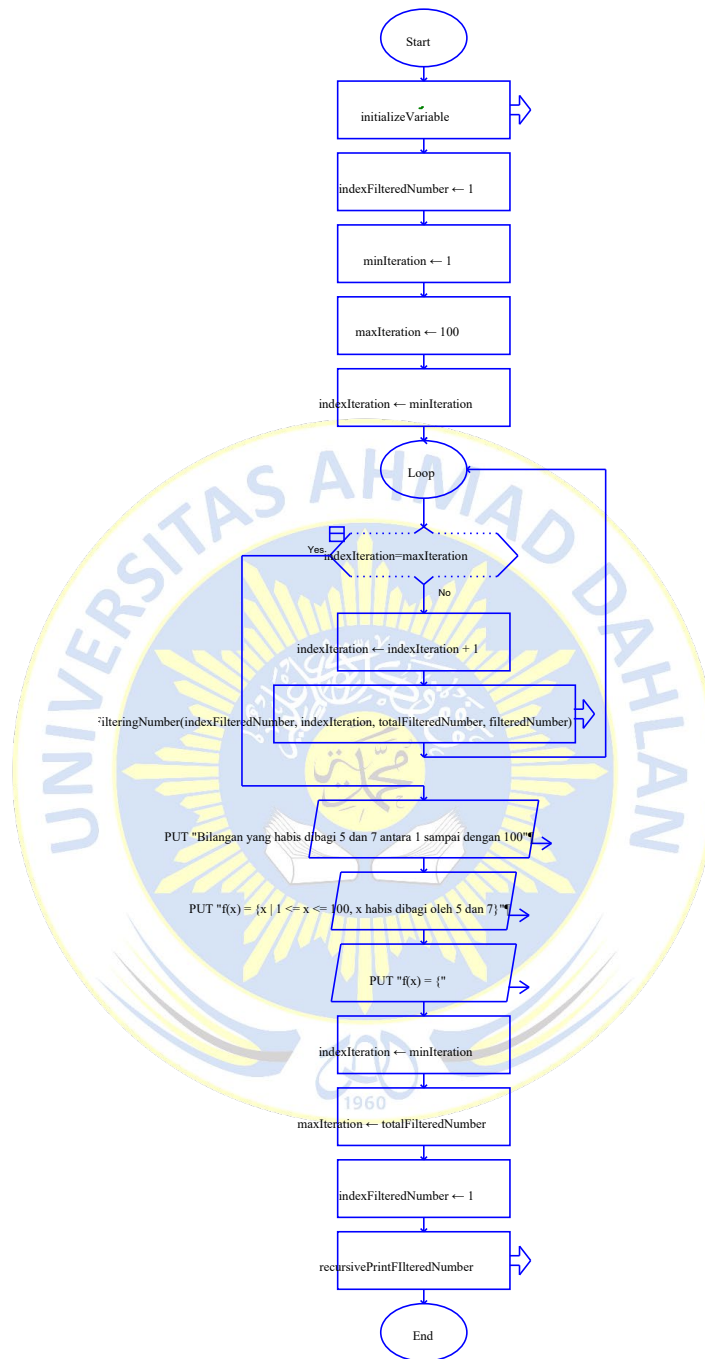
Berikut adalah contoh outputnya:

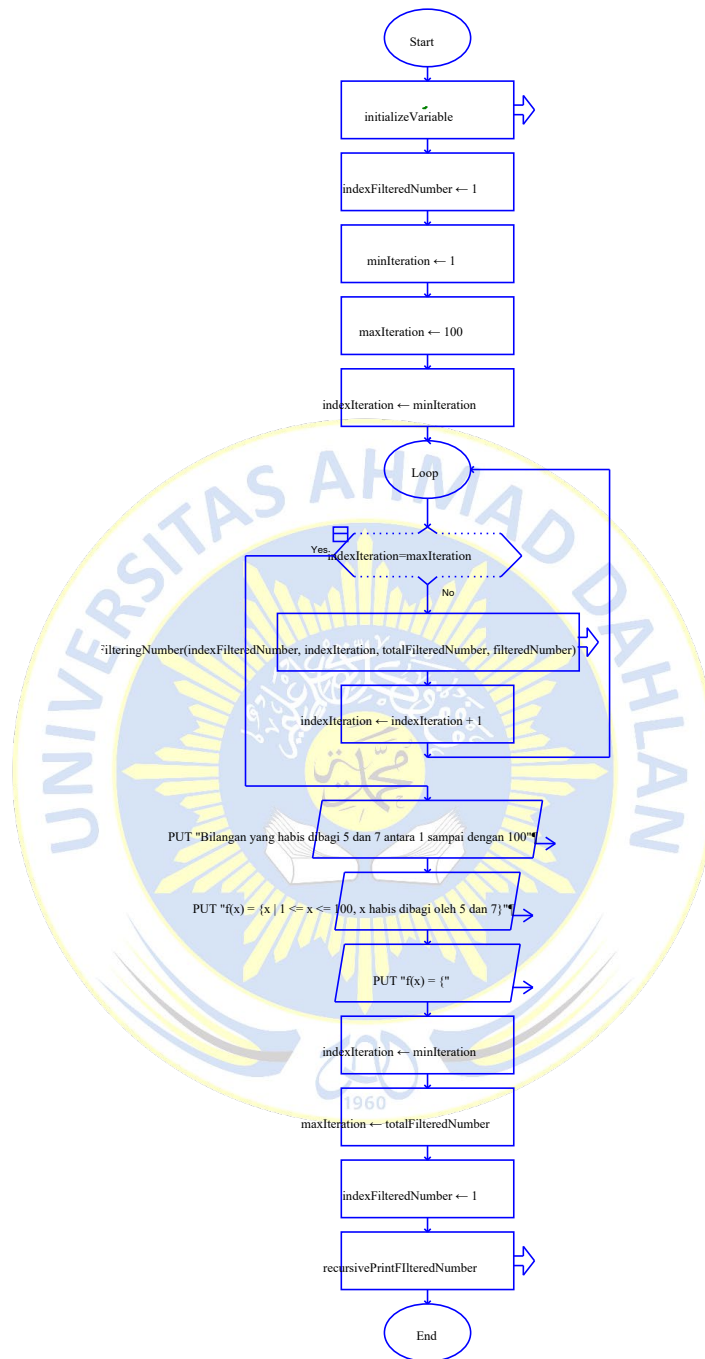


Gambar 11 Output dengan subProgram iteratif. (Sumber: Penulis)

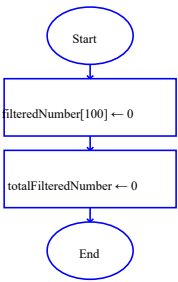
Berikut adalah flowchart subprogram rekursif



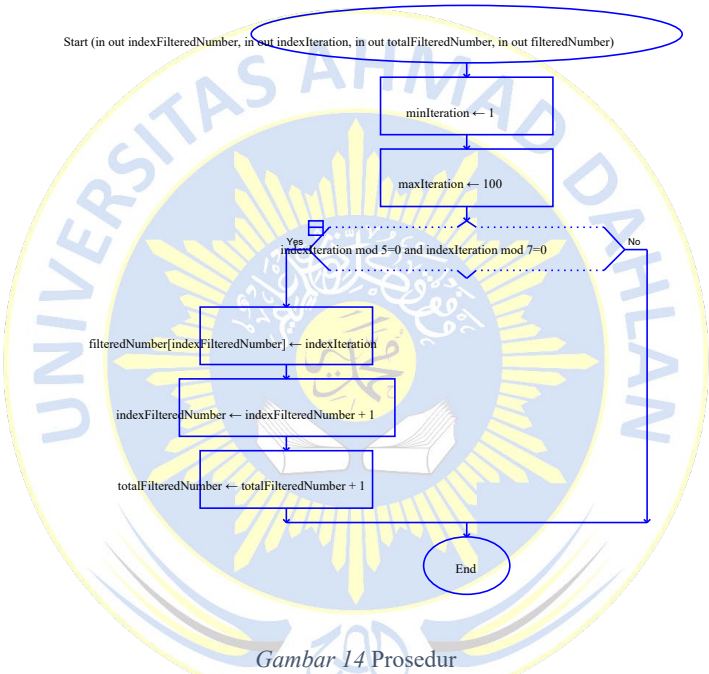




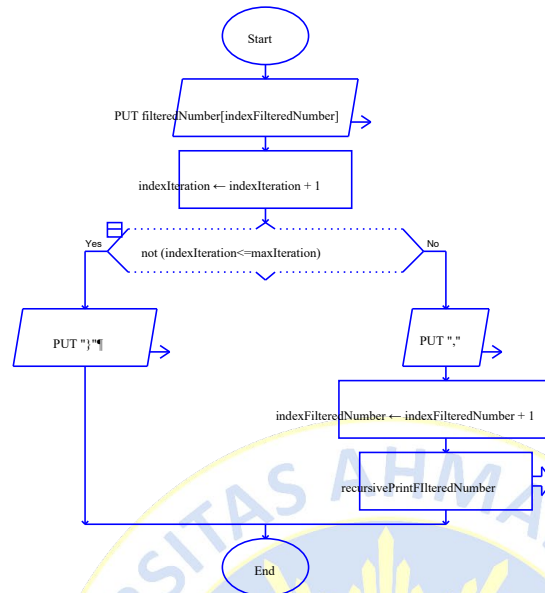
Gambar 12 Fungsi Utama Flowchart. (Sumber: Penulis)



Gambar 13 SubChart initializeVariable (Sumber: Penulis)



Gambar 14 Prosedur recursiveFilteringNumber(indexFilteredNumber,indexIteration,totalFilteredNumber,filteredNumber). (Sumber: Penulis)

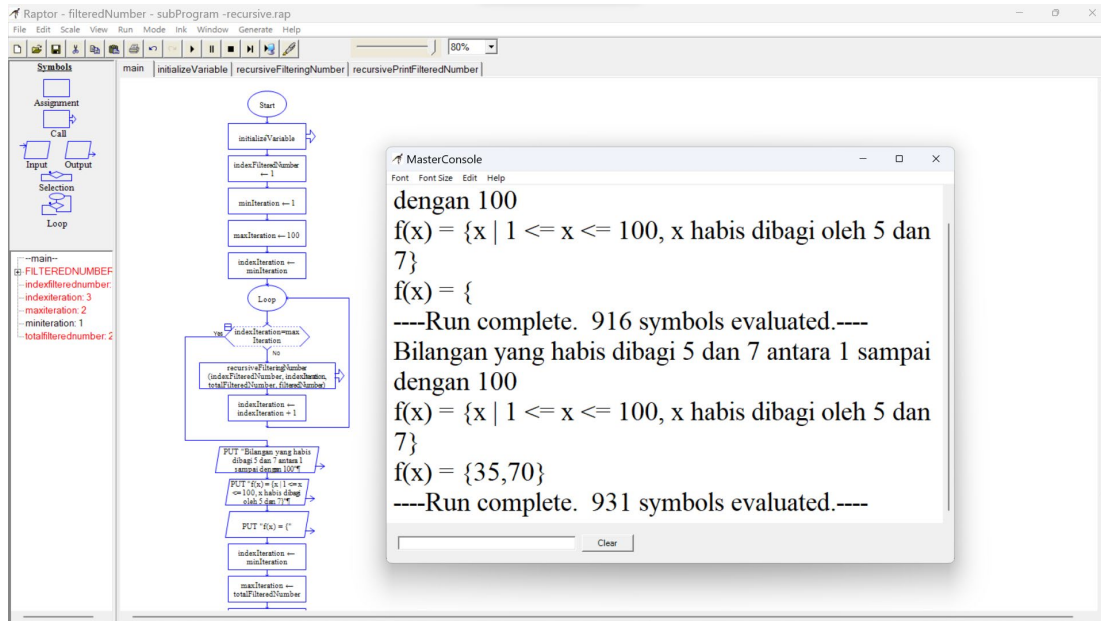


Gambar 15 subchart recursivePrintFilteredNumber. (Sumber: Penulis)

Flowchart ini menggunakan rekursi untuk mencari angka-angka yang habis dibagi 5 dan 7 antara 1 hingga 100. Flowchart ini menggunakan Raptor Berikut penjabaran flowchart:

- 1) Inisialisasi variabel seperti minIteration, maxIteration, indexFilteredNumber, totalFilteredNumber, dan filteredNumber.
- 2) Mulai iterasi dari minIteration hingga maxIteration menggunakan loop while dalam main.
- 3) Panggil fungsi rekursif recursiveFilteringNumber dengan parameter indexIteration, totalFilteredNumber, dan filteredNumber.
- 4) Di dalam fungsi recursiveFilteringNumber, jika indexIteration habis dibagi 5 dan 7, tambahkan ke array filteredNumber dan increment totalFilteredNumber serta indexFilteredNumber.
- 5) Kembalikan nilai array filteredNumber dari fungsi rekursif.
- 6) Cetak bilangan yang habis dibagi oleh 5 dan 7 dalam urutan yang ditemukan.

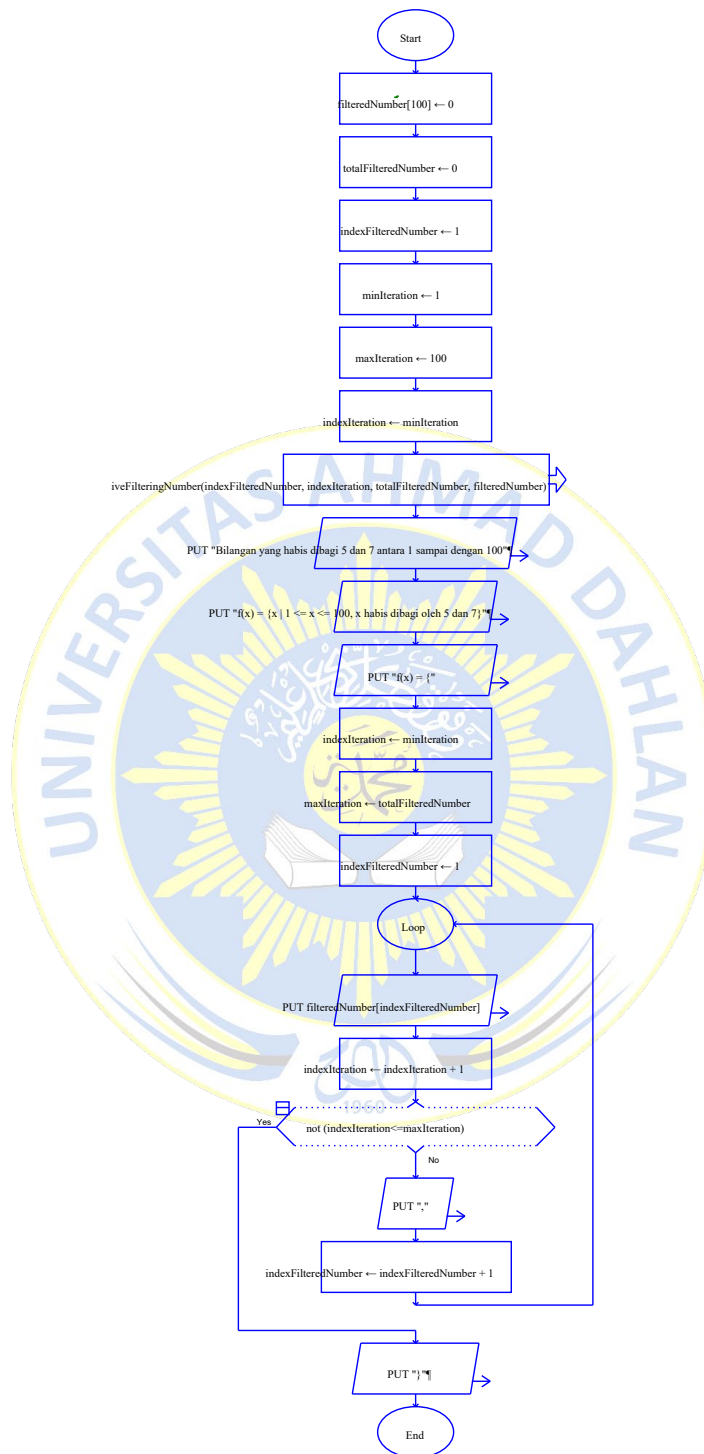
Berikut contoh outputnya



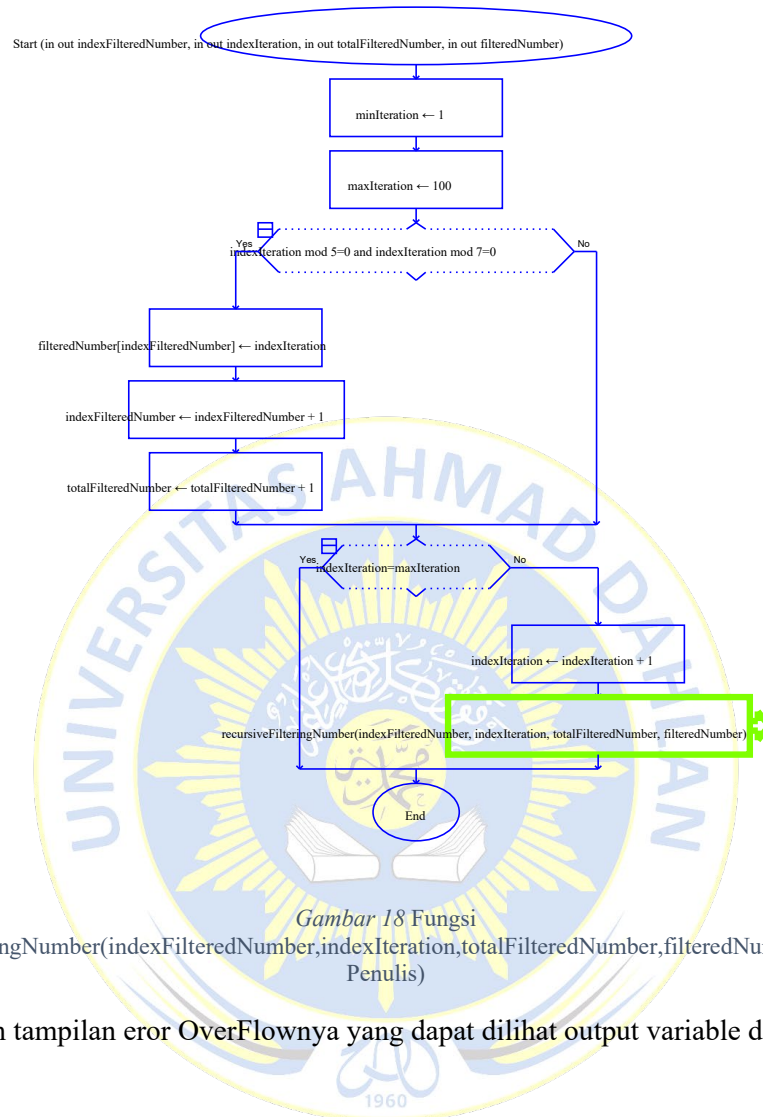
Gambar 16 Output dengan subProgram rekursif. (Sumber: Penulis)

Ada beberapa perbedaan dalam sintaks dan fitur yang didukung. Karena Raptor tidak mendukung pemanggilan fungsi rekursif di dalam fungsi itu sendiri, kode ini menempatkan pemanggilan fungsi rekursif di luar fungsi (dalam loop while pada main). Untuk menyelesaikan masalah dengan cara alternatif, Anda telah mengerjakan nomor sebelumnya (nomor 1) tanpa subprogram dengan subchart.

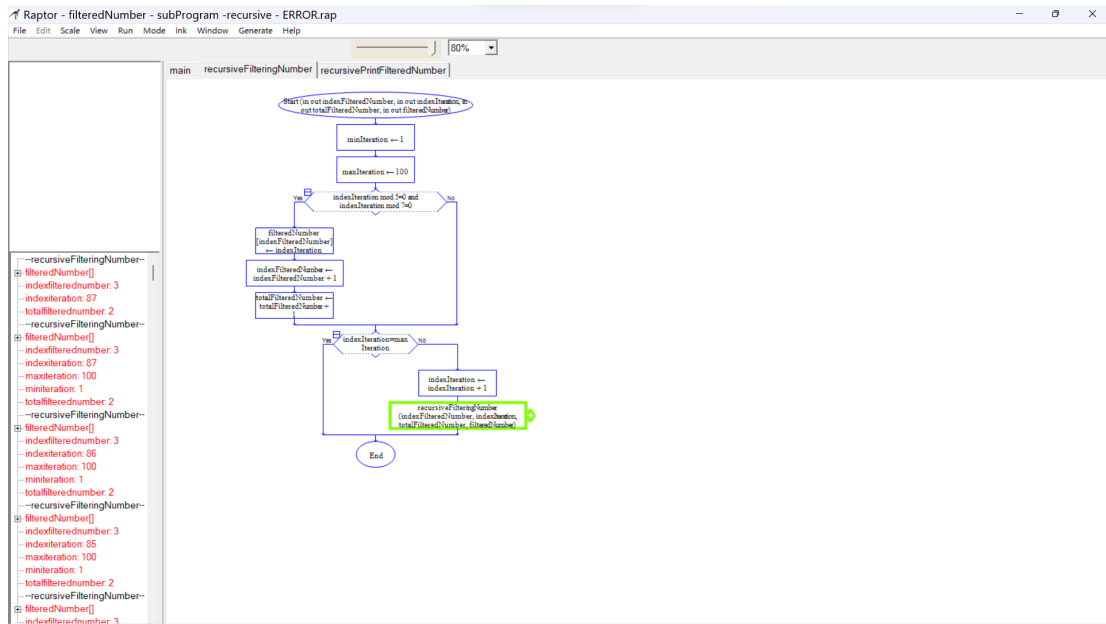
Berikut gambaran flowchart yang benar tetapi ada error OverFlow.



Gambar 17 Flowchart benar tetapi Eror. (Sumber: Penulis)



Berikut adalah tampilan eror OverFlownya yang dapat dilihat output variable di menu kiri.



Gambar 19 Terus mengulang padahal sudah dikasih batas 100. Setelah sampai 100 malah turun lagi 99 98 97 dst. (Sumber: Penulis)

3. Konversikan hasil dari flowchart nomor 1 dan 2 menjadi program C++.

Jawab:

Iteratif tanpa subProgram

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int min_iteration, index_filtered_number, max_iteration, index_iteration, total_filtered_number;
7      const int array_size = 100;
8      int filtered_number[array_size];
9
10     filtered_number[0] = 0;
11     total_filtered_number = 0;
12     index_filtered_number = 1;
13     min_iteration = 1;
14     max_iteration = 100;
15     index_iteration = min_iteration;
16
17     while (true) {
18         if (index_iteration % 5 == 0 && index_iteration % 7 == 0) {
19             filtered_number[index_filtered_number - 1] = index_iteration;
20             index_filtered_number++;
21             total_filtered_number++;
22         }
23
24         if (index_iteration == max_iteration) {
25             break;
26         }
27
28         index_iteration++;
29     }
30
31     cout << "Bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100" << endl;
32     cout << "f(x) = {x | 1 <= x <= 100, x habis dibagi oleh 5 dan 7}" << endl;
33     cout << "f(x) = {";
34
35     index_iteration = min_iteration;
36     max_iteration = total_filtered_number;
37     index_filtered_number = 1;
38
39     while (true) {
40         cout << filtered_number[index_filtered_number - 1];
41         index_iteration++;
42
43         if (!(index_iteration <= max_iteration)) {
44             break;
45         }
46
47         cout << ", ";
48         index_filtered_number++;
49     }
50
51     cout << "}" << endl;
52     return 0;
53 }

```

Gambar 20 Kodongan iteratif tanpa subProgram. (Sumber: Penulis)

Kodingan ini merupakan program C++ yang mencari bilangan yang habis dibagi 5 dan 7 dalam rentang angka dari 1 hingga 100. Pertama, beberapa variabel dideklarasikan seperti `min_iteration`, `index_filtered_number`, `max_iteration`, `index_iteration`, dan

total_filtered_number. Selanjutnya, konstanta array_size diinisialisasi dengan nilai 100 dan sebuah array bernama filtered_number dibuat dengan ukuran array_size.

Pada awal program, beberapa variabel diinisialisasi untuk mengatur proses iterasi dan penyimpanan hasil. filtered_number[0] diset ke 0, total_filtered_number diset ke 0, index_filtered_number diset ke 1, min_iteration diset ke 1, dan max_iteration diset ke 100. Kemudian, index_iteration diinisialisasi dengan nilai min_iteration.

Program kemudian memasuki loop while pertama yang akan berjalan selamanya sampai kondisi break terpenuhi. Di dalam loop ini, program mengecek apakah index_iteration habis dibagi oleh 5 dan 7. Jika habis, nilai index_iteration ditambahkan ke array filtered_number pada posisi index_filtered_number - 1, lalu index_filtered_number dan total_filtered_number bertambah 1.

Setelah itu, program mengecek apakah index_iteration sama dengan max_iteration. Jika kondisi ini terpenuhi, loop akan dihentikan. Jika tidak, index_iteration akan bertambah 1 dan loop akan berlanjut.

Setelah loop pertama selesai, program akan mencetak informasi tentang bilangan yang ditemukan. Pertama, program mencetak bahwa bilangan tersebut habis dibagi 5 dan 7 antara 1 hingga 100. Kemudian, program mencetak definisi matematika dari himpunan yang dihasilkan. Setelah itu, program mencetak anggota himpunan dalam format kurung kurawal.

Untuk mencetak anggota himpunan, program memasuki loop while kedua yang juga akan berjalan selamanya sampai kondisi break terpenuhi. Di dalam loop ini, program mencetak nilai filtered_number pada posisi index_filtered_number - 1. Kemudian, index_iteration bertambah 1. Program mengecek apakah index_iteration kurang dari atau sama dengan max_iteration

(total_filtered_number). Jika tidak, loop akan dihentikan. Jika ya, program mencetak tanda koma dan spasi, lalu index_filtered_number bertambah 1, dan loop berlanjut.

Setelah loop kedua selesai, program mencetak kurung kurawal penutup dan mengakhiri baris. Terakhir, program mengembalikan 0 sebagai status keluar yang menandakan eksekusi berhasil.

Output:

```

34
35 index_iteration = min_iteration;
36 max_iteration = total_filtered_number;
37 index_filtered_number = 1;
38
39 while (true) {
40     cout << filtered_number[index_filtered_number - 1];
41     index_iteration++;
42
43     if ((index_iteration <= max_iteration)) {
44         break;
45     }
46
47     cout << ", ";
48     index_filtered_number++;
49 }
50
51 cout << "}" << endl;
52 return 0;
53 }

```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

PS D:\Document Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4" ; if (\$?) { g++ filteredNumber_iterative.cpp -o filteredNumber_iterative ; if (\$?) { .\filteredNumber_iterative }
bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100
f(x) = {x | 1 <= x <= 100, x habis dibagi oleh 5 dan 7}
f(x) = {35,70}
PS D:\Document Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4>

Gambar 21 output Kodingan iteratif tanpa subProgram. (Sumber: Penulis)

Iteratif dengan subProgram



```

1  #include <iostream>
2  using namespace std;
3
4  // Fungsi rekursif untuk mencetak bilangan yang habis dibagi 5 dan 7 antara minIteration sampai maxIteration
5  void recursivePrintFilteredNumber(int currentNumber, int maxIteration) {
6      if (currentNumber <= maxIteration) {
7          if (currentNumber % 5 == 0 && currentNumber % 7 == 0) {
8              cout << currentNumber;
9
10             // Mengecek apakah ada bilangan habis dibagi 5 dan 7 setelah currentNumber
11             bool isLast = true;
12             for (int i = currentNumber + 1; i <= maxIteration; i++) {
13                 if (i % 5 == 0 && i % 7 == 0) {
14                     isLast = false;
15                     break;
16                 }
17             }
18
19             if (!isLast) {
20                 cout << ", ";
21             }
22             recursivePrintFilteredNumber(currentNumber + 1, maxIteration);
23         }
24     }
25 }
26
27 int main() {
28     int minIteration = 1;
29     int maxIteration = 100;
30
31     cout << "Bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 adalah " << endl;
32     cout << "f(x) = {" << endl;
33     recursivePrintFilteredNumber(minIteration, maxIteration);
34     cout << "}" << endl;
35 }

```

Gambar 22 Kodingan iteratif dengan subProgram. (Sumber: Penulis)

Dalam kode ini, tujuan utama adalah untuk mencetak angka-angka yang habis dibagi oleh 5 dan 7 di antara angka 1 hingga 100. Kode ini ditulis dalam bahasa pemrograman C++ dan menggunakan beberapa fungsi untuk mencapai tujuannya.

Pertama-tama, kode program ini menyertakan pustaka "iostream" yang memungkinkan penggunaan operasi input dan output dasar, seperti 'cout'. Selanjutnya, 'using namespace std' memungkinkan kita menggunakan elemen standar dari namespace 'std' tanpa harus menambahkan awalan 'std::'.

Konstanta 'array_size' didefinisikan dengan nilai 100 untuk mengatur ukuran array yang akan digunakan nanti. Kemudian, ada tiga fungsi yang didefinisikan dalam kode ini: 'initializeVariable', 'filteringNumber', dan 'printFilteredNumber'.

Fungsi 'initializeVariable' menginisialisasi tiga variabel integer dengan referensi (&) yang akan digunakan dalam proses iterasi. Variabel 'min_iteration' diinisialisasi menjadi 1, 'max_iteration' menjadi 100, dan 'index_iteration' sama dengan 'min_iteration'.

Fungsi 'filteringNumber' memiliki dua parameter integer dengan referensi ('&index_filtered_number') dan satu parameter integer array ('filtered_number[]'). Fungsi ini melakukan iterasi dari 'min_iteration' hingga 'max_iteration' (dalam hal ini, 1 hingga 100) dan memeriksa apakah angka tersebut habis dibagi oleh 5 dan 7. Jika kondisi ini terpenuhi, angka tersebut ditambahkan ke array 'filtered_number' dan indeks 'index_filtered_number' ditingkatkan dengan satu.

Fungsi 'printFilteredNumber' mencetak angka-angka yang telah difilter dalam format yang dapat dibaca. Fungsi ini menggunakan parameter array integer ('filtered_number[]') dan satu parameter integer ('total_filtered_number'). Output yang dihasilkan menampilkan angka yang habis dibagi oleh 5 dan 7 antara 1 hingga 100.

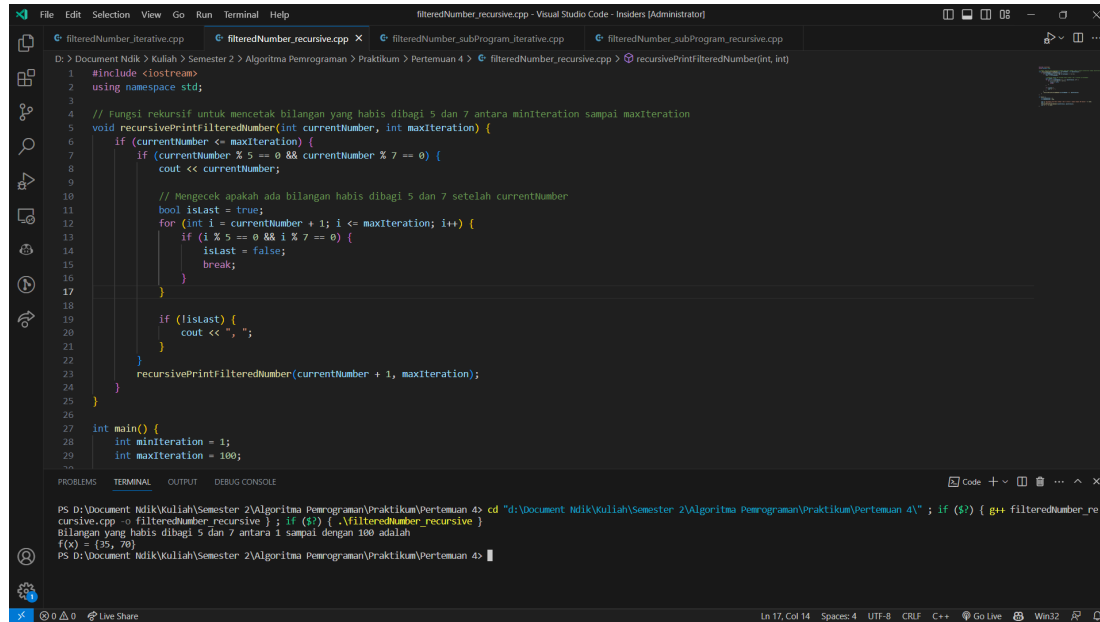
Dalam fungsi 'main', beberapa variabel dideklarasikan, termasuk 'min_iteration', 'max_iteration', 'index_iteration', 'index_filtered_number', dan array 'filtered_number'. Kemudian, fungsi 'initializeVariable' dipanggil untuk mengatur nilai-nilai awal variabel tersebut.

Selanjutnya, 'index_filtered_number' diinisialisasi menjadi 1 dan fungsi 'filteringNumber' dipanggil untuk mengisi array 'filtered_number' dengan angka-angka yang memenuhi kondisi (habis dibagi oleh 5 dan 7). Setelah itu, variabel 'total_filtered_number' dihitung sebagai 'index_filtered_number - 1'.

Akhirnya, fungsi 'printFilteredNumber' dipanggil untuk mencetak angka-angka yang telah difilter. Fungsi 'main' mengembalikan 0 untuk menandakan bahwa program telah berakhir dengan sukses.

Secara keseluruhan, kode ini merupakan contoh sederhana dari penggunaan fungsi dan struktur kontrol dalam bahasa pemrograman C++ untuk menyelesaikan masalah khusus, yaitu mencetak angka-angka yang habis dibagi oleh 5 dan 7 di antara angka 1 hingga 100.

Output:



```

1 #include <iostream>
2 using namespace std;
3
4 // Fungsi rekursif untuk mencetak bilangan yang habis dibagi 5 dan 7 antara miniteration sampai maxiteration
5 void recursivePrintFilteredNumber(int currentNumber, int maxiteration) {
6     if (currentNumber <= maxiteration) {
7         if (currentNumber % 5 == 0 && currentNumber % 7 == 0) {
8             cout << currentNumber;
9
10            // Mengecek apakah ada bilangan habis dibagi 5 dan 7 setelah currentNumber
11            bool istailast = true;
12            for (int i = currentNumber + 1; i <= maxiteration; i++) {
13                if (i % 5 == 0 && i % 7 == 0) {
14                    istailast = false;
15                    break;
16                }
17            }
18
19            if (!istailast) {
20                cout << ", ";
21            }
22            recursivePrintFilteredNumber(currentNumber + 1, maxiteration);
23        }
24    }
25 }
26
27 int main() {
28     int miniteration = 1;
29     int maxiteration = 100;
30
31     recursivePrintFilteredNumber(miniteration, maxiteration);
32 }

```

Terminal Output:

```

PS D:\Document\Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document\Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4\" ; if ($?) { g++ filteredNumber_re
cursive.cpp -o filteredNumber_recursive } ; if ($?) { .\filteredNumber_recursive }
Bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100 adalah
f(x) = {35, 70}
PS D:\Document\Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4>

```

Gambar 23 Kodingan iteratif dengan subProgram. (Sumber: Penulis)

Rekursif dengan subProgram

Untuk rekursif wajib menggunakan subProgram, karena agar bisa memanggil fungsi sendirinya. Jadi, rekursif untuk mencetak bilangan yang habis dibagi 5 dan 7 antara 1 harus menggunakan subProgram.


```

1  #include <iostream>
2  using namespace std;
3
4  // Fungsi rekursif untuk mencetak bilangan yang habis dibagi 5 dan 7 antara minIteration sampai maxIteration
5  void recursivePrintFilteredNumber(int currentNumber, int maxIteration) {
6      if (currentNumber <= maxIteration) {
7          if (currentNumber % 5 == 0 && currentNumber % 7 == 0) {
8              cout << currentNumber;
9
10             // Mengecek apakah ada bilangan habis dibagi 5 dan 7 setelah currentNumber
11             bool isLast = true;
12             for (int i = currentNumber + 1; i <= maxIteration; i++) {
13                 if (i % 5 == 0 && i % 7 == 0) {
14                     isLast = false;
15                     break;
16                 }
17             }
18
19             if (!isLast) {
20                 cout << ", ";
21             }
22         }
23         recursivePrintFilteredNumber(currentNumber + 1, maxIteration);
24     }
25 }
26
27 int main() {
28     int minIteration = 1;
29     int maxIteration = 100;
30
31     cout << "Bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100" << endl;
32     cout << "f(x) = {x | 1 <= x <= 100, x habis dibagi oleh 5 dan 7}" << endl;
33     cout << "f(x) = {" << endl;
34     recursivePrintFilteredNumber(minIteration, maxIteration);
35     cout << "}" << endl;
36 }

```

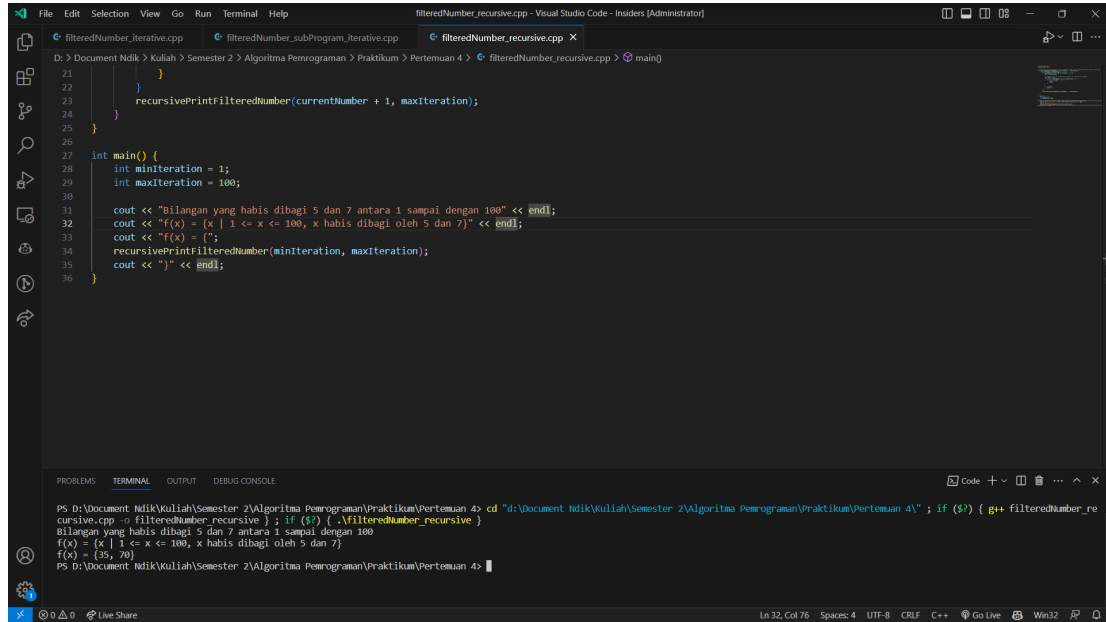
Gambar 24 Kodingan rekursif dengan subProgram. (Sumber: Penulis)

Output:

Kode C++ ini mencetak bilangan yang habis dibagi 5 dan 7 dalam rentang 1 hingga 100 menggunakan konsep rekursi. Fungsi `recursivePrintFilteredNumber` mengambil dua parameter, yaitu `currentNumber` dan `maxIteration`. Fungsi ini memeriksa apakah `currentNumber` habis dibagi 5 dan 7; jika ya, angka tersebut dicetak.

Selanjutnya, kode mengecek apakah ada bilangan berikutnya yang juga habis dibagi 5 dan 7 dengan menggunakan loop `for`. Jika ditemukan, variabel `isLast` diatur menjadi `false` dan loop dihentikan. Jika `isLast` bernilai `false`, tanda koma dan spasi akan dicetak untuk memisahkan angka-angka. Fungsi kemudian dipanggil secara rekursif dengan parameter `currentNumber + 1` dan `maxIteration`.

Dalam fungsi main, nilai minIteration diatur ke 1 dan maxIteration ke 100. Kode mencetak informasi tentang apa yang akan dilakukan program sebelum memanggil fungsi recursivePrintFilteredNumber dengan parameter minIteration dan maxIteration. Output diakhiri dengan tanda kurung kurawal penutup "}".



```

21
22
23     recursivePrintFilteredNumber(currentNumber + 1, maxIteration);
24 }
25
26
27 int main() {
28     int minIteration = 1;
29     int maxIteration = 100;
30
31     cout << "Bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100" << endl;
32     cout << "f(x) = {x | 1 <= x <= 100, x habis dibagi oleh 5 dan 7}" << endl;
33     cout << "f(x) = {" << endl;
34     recursivePrintFilteredNumber(minIteration, maxIteration);
35     cout << "}" << endl;
36 }
    
```

```

PS D:\Document Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4" & if ($?) { g++ filteredNumber_re
cursive.cpp -o filteredNumber_recursive } ; if ($?) { .\filteredNumber_recursive }
Bilangan yang habis dibagi 5 dan 7 antara 1 sampai dengan 100
f(x) = {x | 1 <= x <= 100, x habis dibagi oleh 5 dan 7}
f(x) = {35, 70}
PS D:\Document Ndik\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4>
    
```

Gambar 25 output rekursif dengan subProgram. (Sumber: Penulis)

Untuk mengakses kodingan maupun Flowchart, bisa mengakses link berikut:

<https://github.com/IRedDragonICY/Programming-Algorithms>