**LAPORAN PRAKTIKUM**

**ALGORITMA PEMROGRAMAN**

**“POST TEST 9: ARRAY 1-2 DIMENSI”**

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 mengurutkan data array 2 dimensi berukuran 3x3 dengan menggunakan algoritma bubble sort! 3](#_Toc138321932)

[2. Seperti nomor 1, gunakan subprogam dalam flowchart untuk mengurutkan data array 2 dimensi berukuran 3x3 dengan menggunakan algoritma bubble sort! 10](#_Toc138321933)

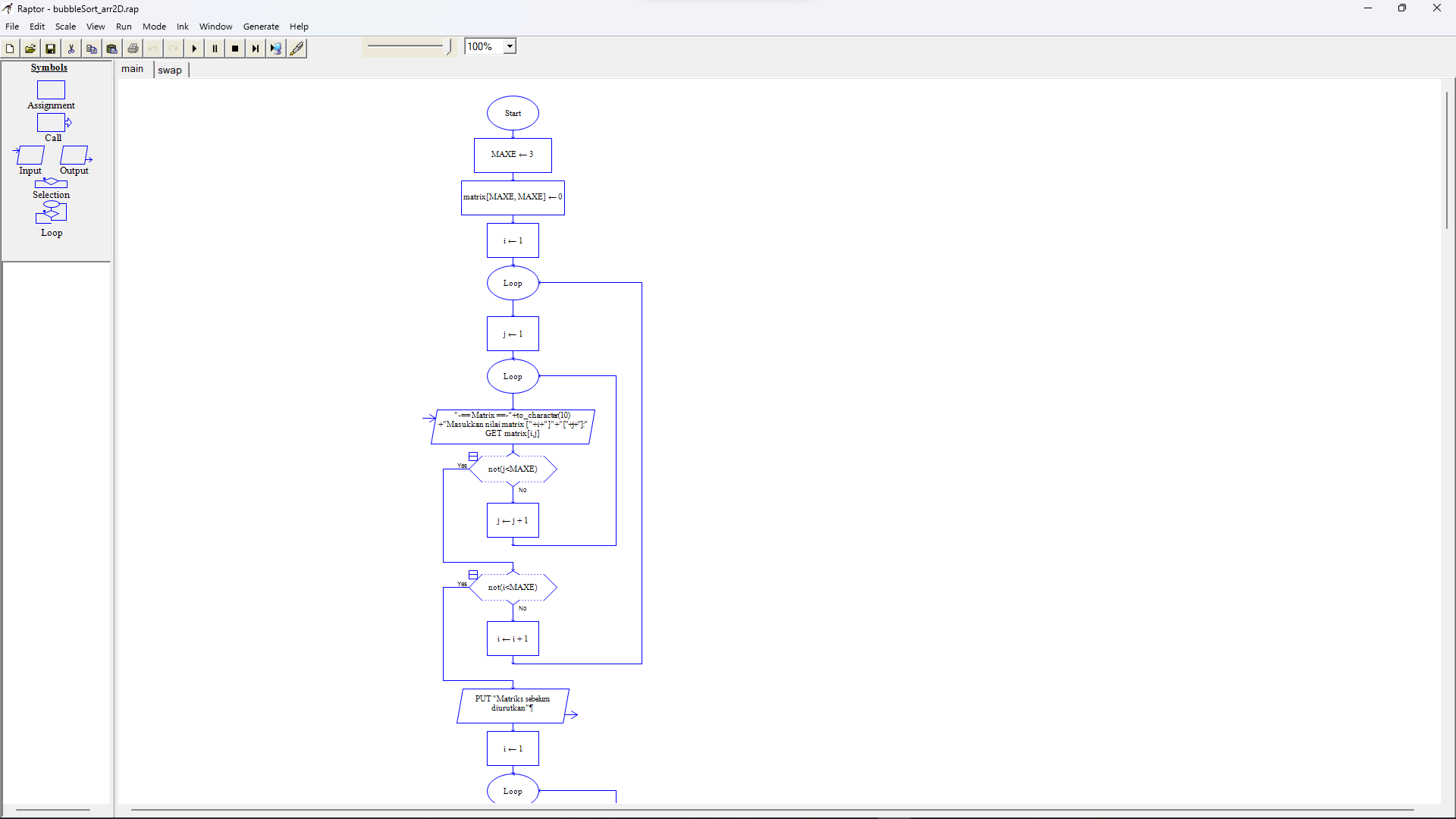
[3. Konversikan hasil dari flowchart nomor 1 dan 2 menjadi progam C++ Hasil konversi! 14](#_Toc138321934)

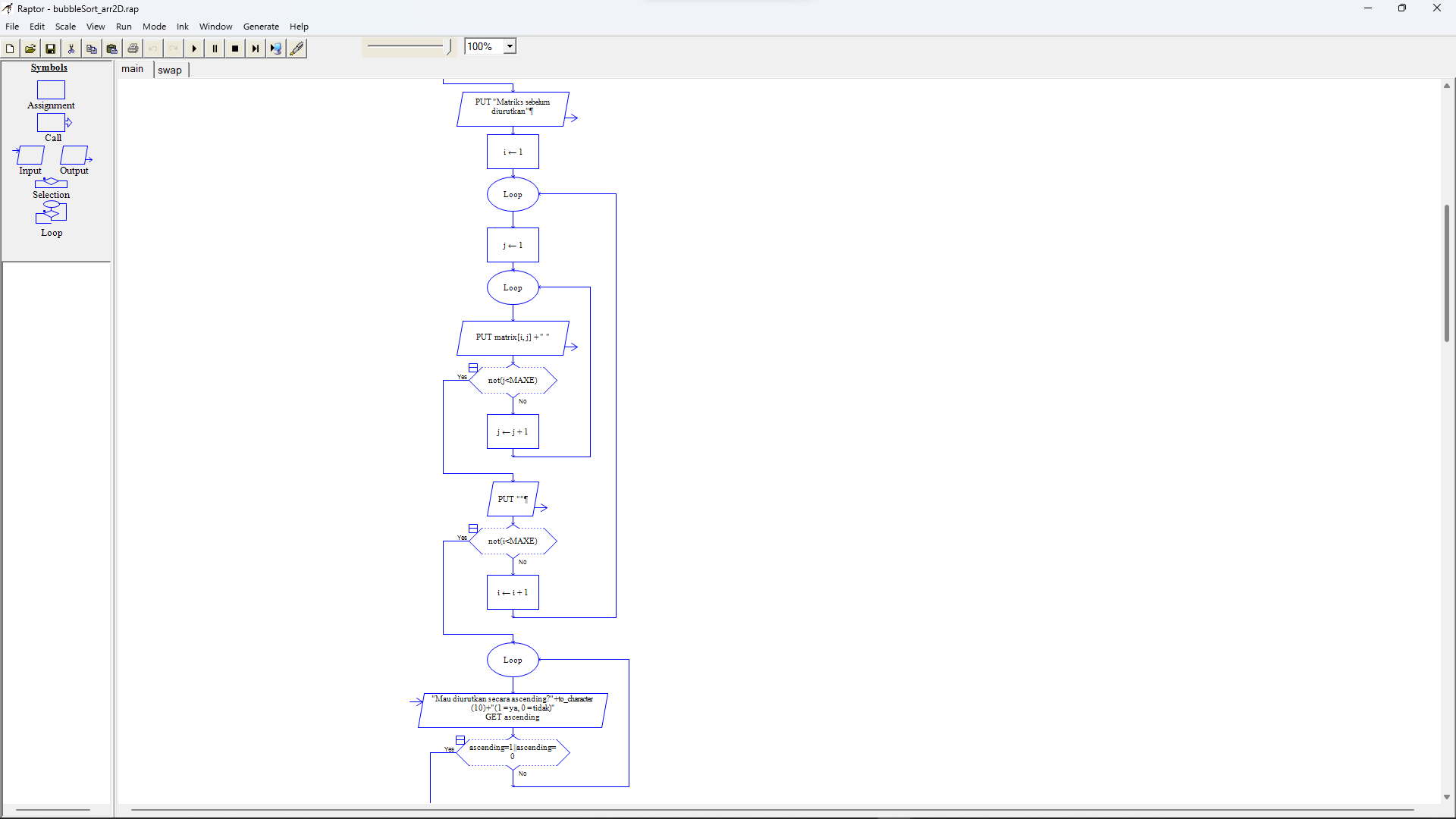
[Konversi hasil flowchart nomor 1 15](#_Toc138321935)

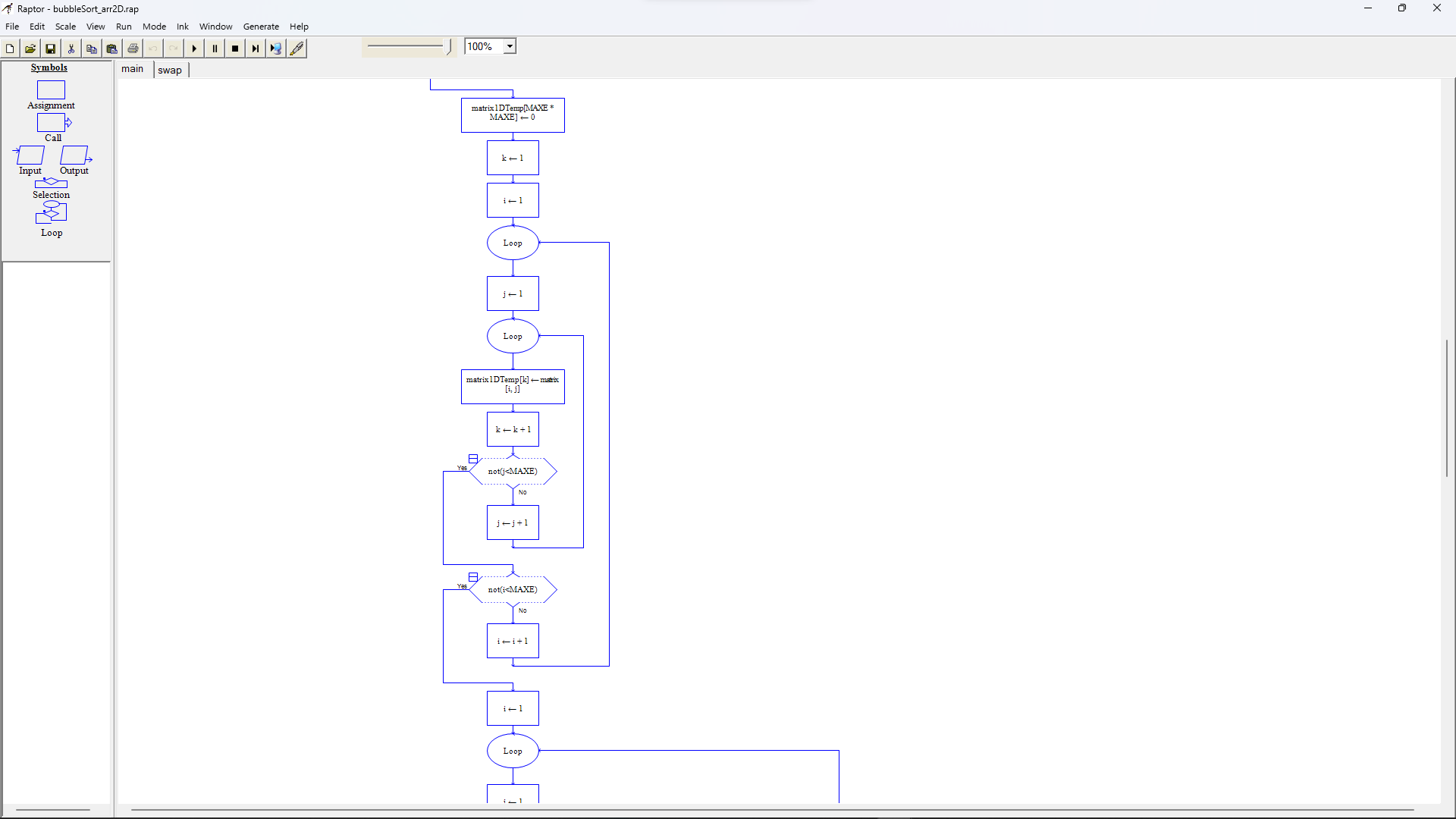
[Konversi hasil flowchart nomor 2 (dengan subprogram) 18](#_Toc138321936)

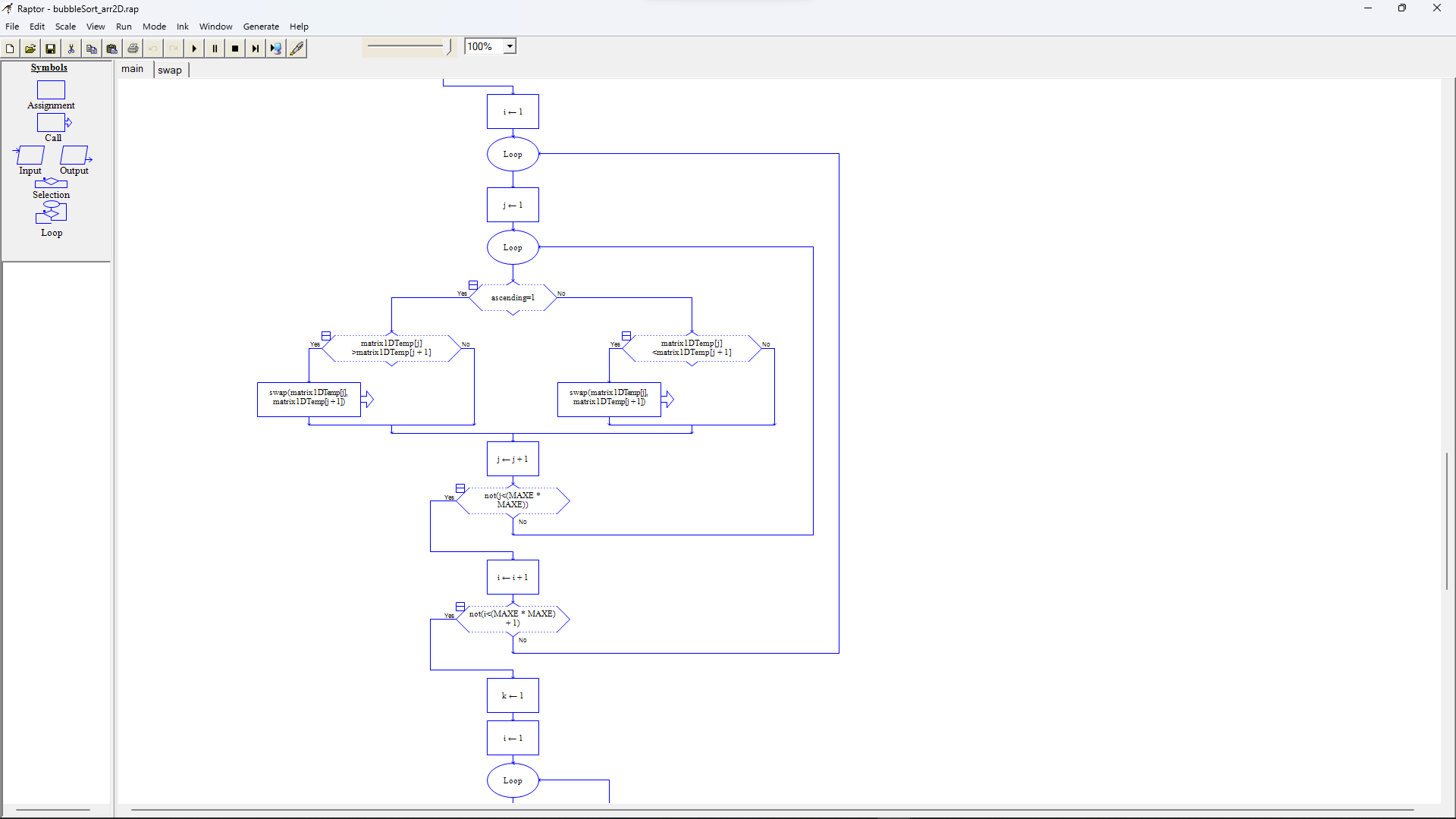
1. Buat lah flowchart untuk mengurutkan data array 2 dimensi berukuran 3x3 dengan menggunakan algoritma bubble sort!

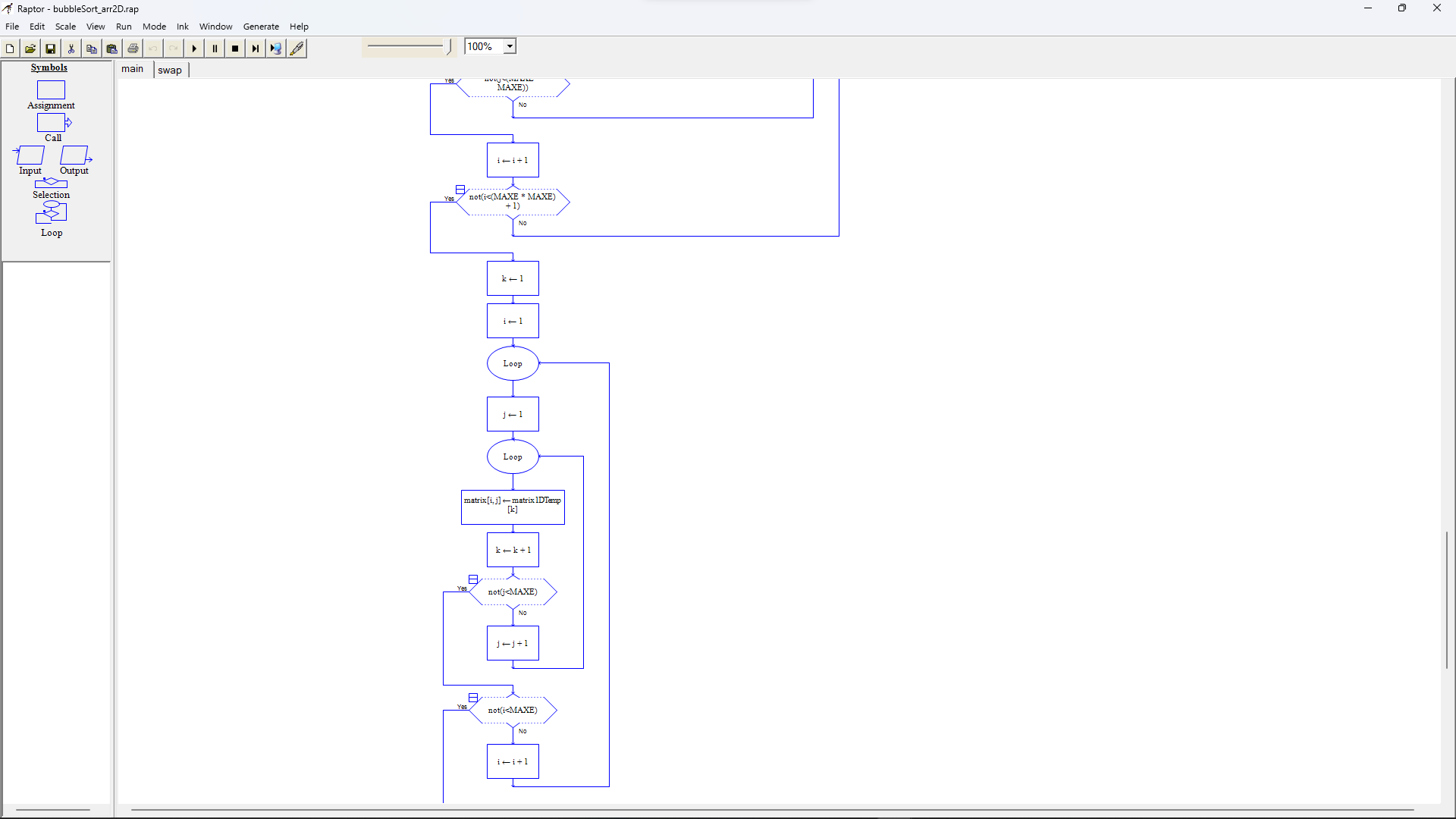
Berikut adalah flowchart untuk mengurutkan data array 2 dimensi berukuran 3x3 dengan menggunakan algoritma bubble sort!

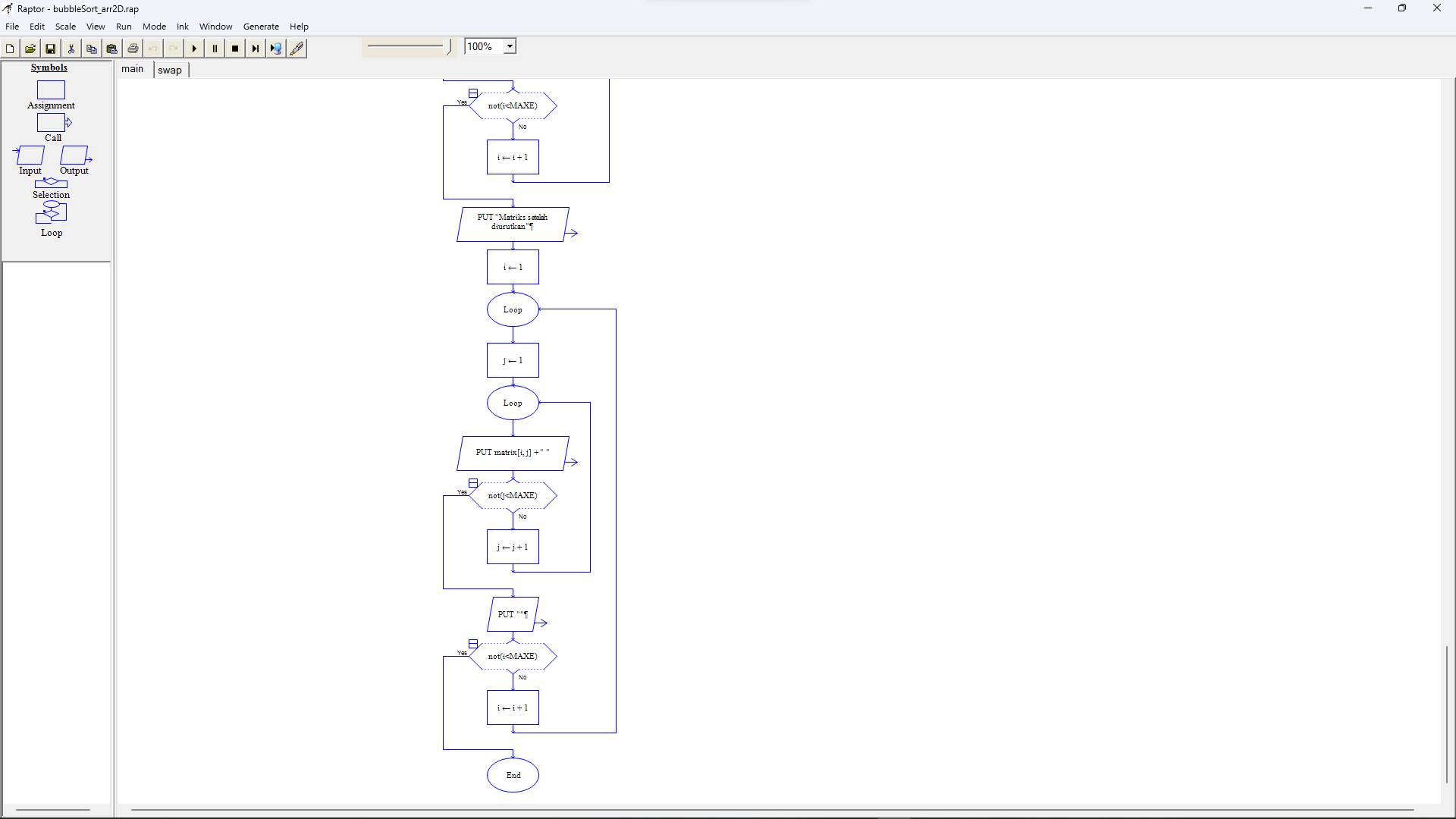












Gambar 1 Flowchart utama. (Sumber: Penulis)

Berikut adalah flowchart penuhnya





Gambar 2 fungsi swap() sebagai subprogram bantuan. (Sumber: Penulis)

Flowchart tersebut merupakan implementasi dari algoritma Bubble Sort untuk mengurutkan matriks dua dimensi. Berikut penjelasan alur flowchart tersebut dalam bentuk langkah-langkah yang dilakukan oleh program, tanpa menyertakan tipe data variabelnya.

Langkah pertama dalam flowchart adalah inisialisasi beberapa variabel yang akan digunakan. Variabel MAXE diatur dengan nilai 3, yang mengindikasikan ukuran maksimal matriks. Selanjutnya, elemen matriks diatur semua nilainya menjadi 0.

Setelah inisialisasi, program akan meminta pengguna untuk memasukkan nilai-nilai matriks. Hal ini dilakukan dengan menggunakan dua perulangan bersarang. Perulangan pertama digunakan untuk mengiterasi baris matriks (variabel i), sementara perulangan kedua digunakan untuk mengiterasi kolom matriks (variabel j). Pada setiap iterasi, program akan meminta pengguna untuk memasukkan nilai matriks[i][j]. Nilai-nilai ini akan disimpan dalam matriks.

Setelah matriks dimasukkan, program akan menampilkan matriks sebelum diurutkan. Ini juga dilakukan dengan menggunakan dua perulangan bersarang untuk mengiterasi setiap elemen matriks. Program akan mencetak setiap elemen dan memberikan baris baru setelah mencetak semua elemen dalam satu baris.

Selanjutnya, program akan meminta pengguna untuk memilih apakah ingin mengurutkan matriks secara menaik (ascending) atau tidak. Pengguna diminta untuk memasukkan nilai 1 jika ingin mengurutkan secara menaik, atau memasukkan nilai 0 jika tidak ingin mengurutkan.

Setelah pengguna memilih, program akan membuat salinan satu dimensi dari matriks (matrix1DTemp) dengan menggabungkan semua elemen matriks ke dalam satu dimensi. Ini dilakukan dengan menggunakan dua perulangan bersarang lagi untuk mengiterasi setiap elemen matriks dan menyalinnya ke matriks satu dimensi.

Setelah salinan matriks dibuat, program akan menjalankan algoritma Bubble Sort untuk mengurutkan matriks. Algoritma ini akan membandingkan setiap elemen dengan elemen berikutnya dalam matriks dan menukar posisi jika diperlukan. Jika pengguna memilih pengurutan menaik (ascending), maka elemen akan ditukar jika elemen saat ini lebih besar dari elemen berikutnya. Jika pengguna memilih untuk tidak mengurutkan secara menaik, maka elemen akan ditukar jika elemen saat ini lebih kecil dari elemen berikutnya.

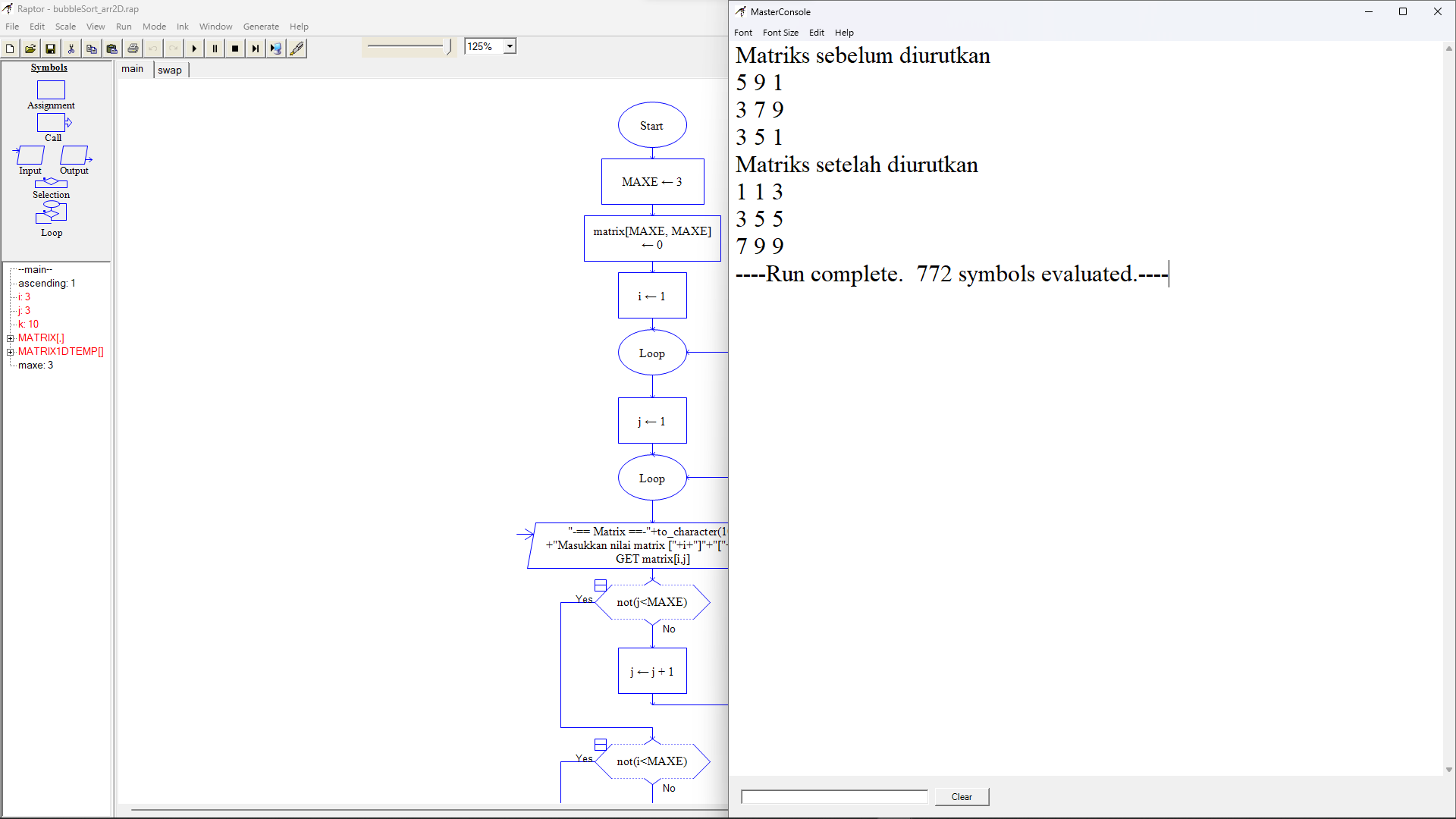
Perulangan digunakan untuk melintasi seluruh matriks sebanyak (MAXE\*MAXE)+1 kali, karena di setiap iterasi, elemen terbesar atau terkecil akan "naik" ke posisi yang benar. Variabel i dan j digunakan untuk melacak posisi saat ini dalam matriks.

Setelah pengurutan selesai, program akan mengembalikan elemen-elemen matriks yang sudah diurutkan dari matriks satu dimensi ke matriks dua dimensi semula. Hal ini dilakukan dengan menggunakan dua perulangan bersarang lagi untuk melintasi matriks dua dimensi dan mengisinya dengan elemen-elemen dari matriks satu dimensi.

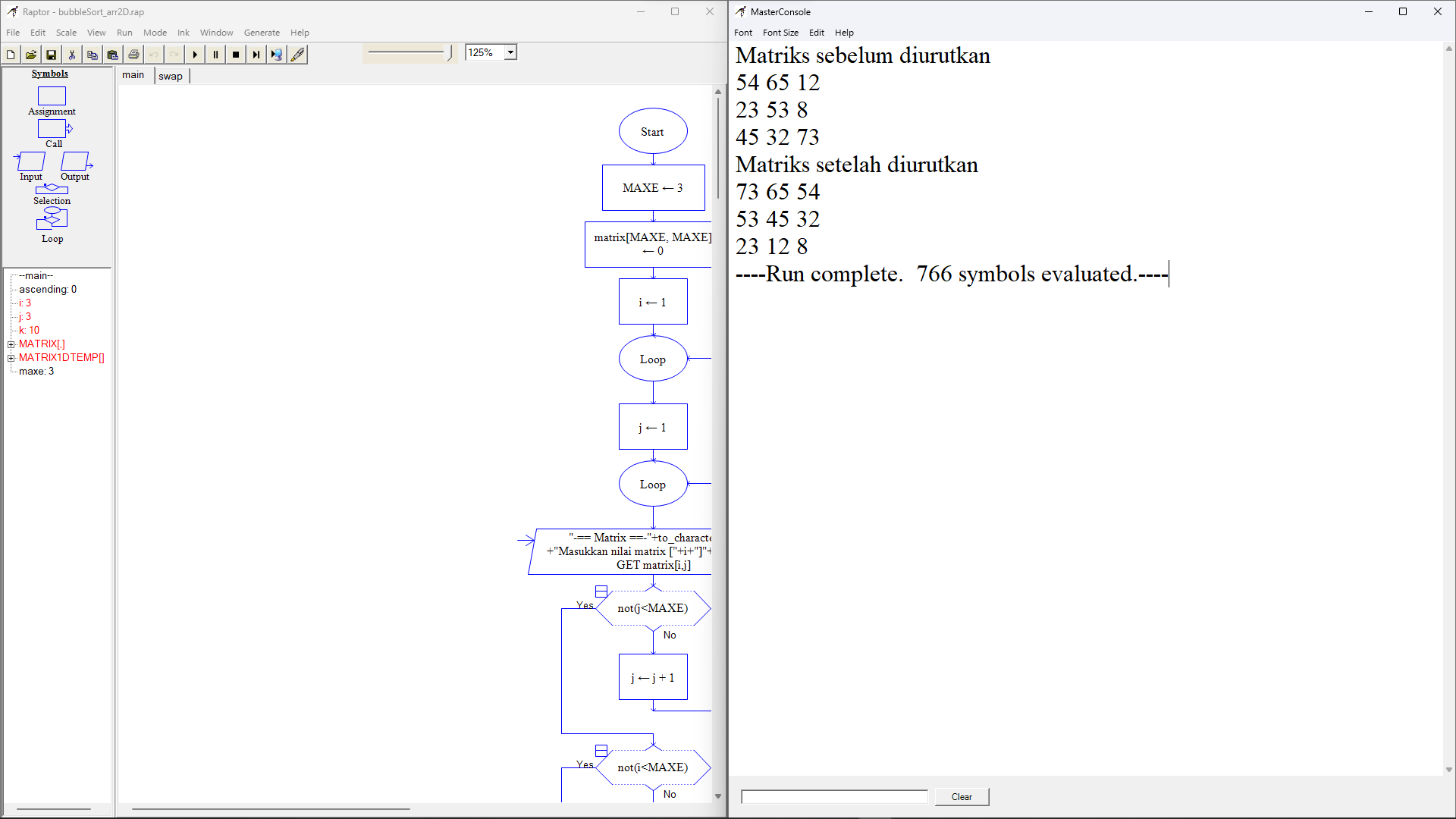
Terakhir, program akan menampilkan matriks setelah diurutkan. Ini juga menggunakan dua perulangan bersarang untuk mencetak setiap elemen dalam matriks dengan baris baru setelah mencetak semua elemen dalam satu baris.

Dengan demikian, itulah alur flowchart yang dijelaskan dalam bentuk langkah-langkah yang dilakukan oleh program untuk mengurutkan matriks dua dimensi menggunakan algoritma Bubble Sort.

Berikut adalah outputnya:



Gambar 3 Output mengurutkan secara Ascending. (Sumber: Penulis)



Gambar 4 Output mengurutkan secara descending. (Sumber: Penulis)

1. Seperti nomor 1, gunakan subprogam dalam flowchart untuk mengurutkan data array 2 dimensi berukuran 3x3 dengan menggunakan algoritma bubble sort!

Berikut adalah flowchart untuk mengurutkan data array 2 dimensi berukuran 3x3 dengan menggunakan algoritma bubble sort dan menggunakan subprogram.



Gambar 5 fungsi utama flowchart. (Sumber: Penulis)



Gambar 6 fungsi swap() sebagai subprogram bantuan. (Sumber: Penulis)

Berikut adalah penjelasan mengenai flowchart RAPTOR yang menggunakan subprogram:

* Subprogram "swap":

Subprogram ini digunakan untuk menukar nilai antara dua variabel. Variabel "a" dan "b" diterima sebagai parameter oleh subprogram ini. Nilai dari "a" disimpan dalam variabel sementara "temp", kemudian nilai "a" diubah menjadi nilai "b" dan nilai "b" diubah menjadi nilai "temp". Ini merupakan langkah yang umum dalam algoritma pengurutan seperti Bubble Sort.

* Subprogram "inputMatrix":

Subprogram ini bertanggung jawab untuk meminta pengguna memasukkan nilai-nilai matriks. Variabel yang digunakan dalam subprogram ini adalah "matrix" yang merupakan matriks dua dimensi, dan "MAXE" yang menyimpan ukuran maksimal matriks. Dalam perulangan bersarang, subprogram ini meminta pengguna memasukkan nilai untuk setiap elemen matriks[i][j]. Nilai-nilai ini kemudian disimpan dalam matriks.

* Subprogram "printArr2D":

Subprogram ini digunakan untuk mencetak matriks dalam format yang terstruktur. Matriks dan "MAXE" (ukuran maksimal matriks) diterima sebagai parameter. Dalam perulangan bersarang, subprogram ini mencetak setiap elemen matriks dengan menggunakan pernyataan Put. Setelah mencetak semua elemen dalam satu baris, subprogram ini memberikan baris baru dan melanjutkan dengan baris berikutnya.

* Subprogram "bubbleSort":

Subprogram ini adalah inti dari algoritma Bubble Sort. Ini melakukan pengurutan matriks menggunakan algoritma Bubble Sort yang dibahas sebelumnya. Subprogram ini menerima matriks, variabel "ascending" yang menentukan apakah pengurutan dilakukan secara menaik atau menurun, dan "MAXE" (ukuran maksimal matriks) sebagai parameter. Pertama, subprogram ini membuat salinan satu dimensi dari matriks menggunakan subprogram "convertMatrixTo1D". Kemudian, menggunakan perulangan bersarang, subprogram ini membandingkan dan menukar elemen-elemen matriks satu dimensi sesuai dengan aturan pengurutan yang ditentukan oleh variabel "ascending". Setelah pengurutan selesai, subprogram ini mengubah kembali matriks satu dimensi ke matriks dua dimensi menggunakan subprogram "convert1DToMatrix".

* Subprogram "convertMatrixTo1D":

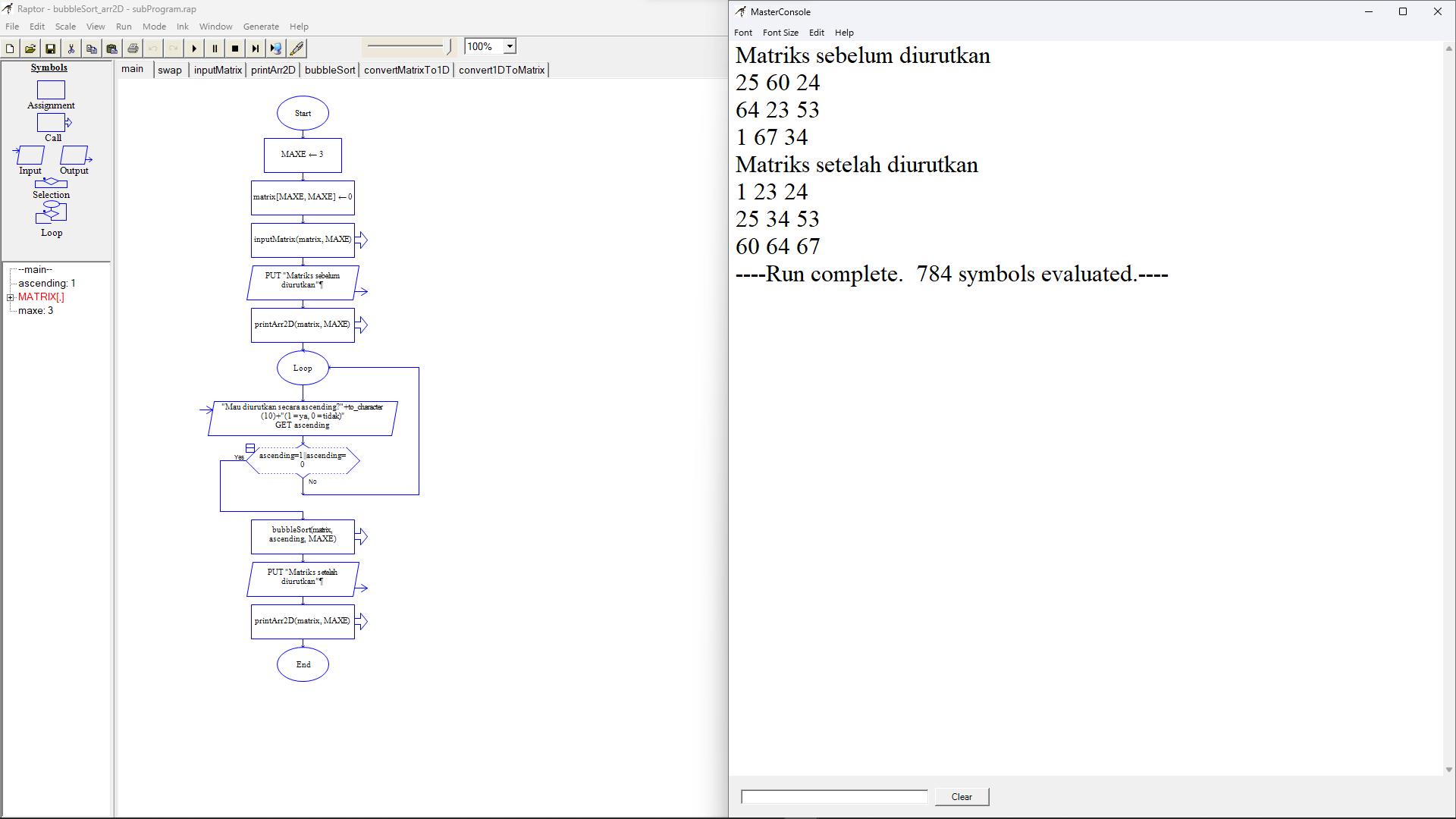
Subprogram ini digunakan untuk mengubah matriks dua dimensi menjadi matriks satu dimensi. Matriks dua dimensi dan "MAXE" diterima sebagai parameter. Dalam perulangan bersarang, subprogram ini menyalin setiap elemen matriks dua dimensi ke matriks satu dimensi. Setiap elemen matriks dua dimensi ditentukan oleh variabel "i" dan "j", sedangkan indeks pada matriks satu dimensi ditentukan oleh variabel "k".

* Subprogram "convert1DToMatrix":

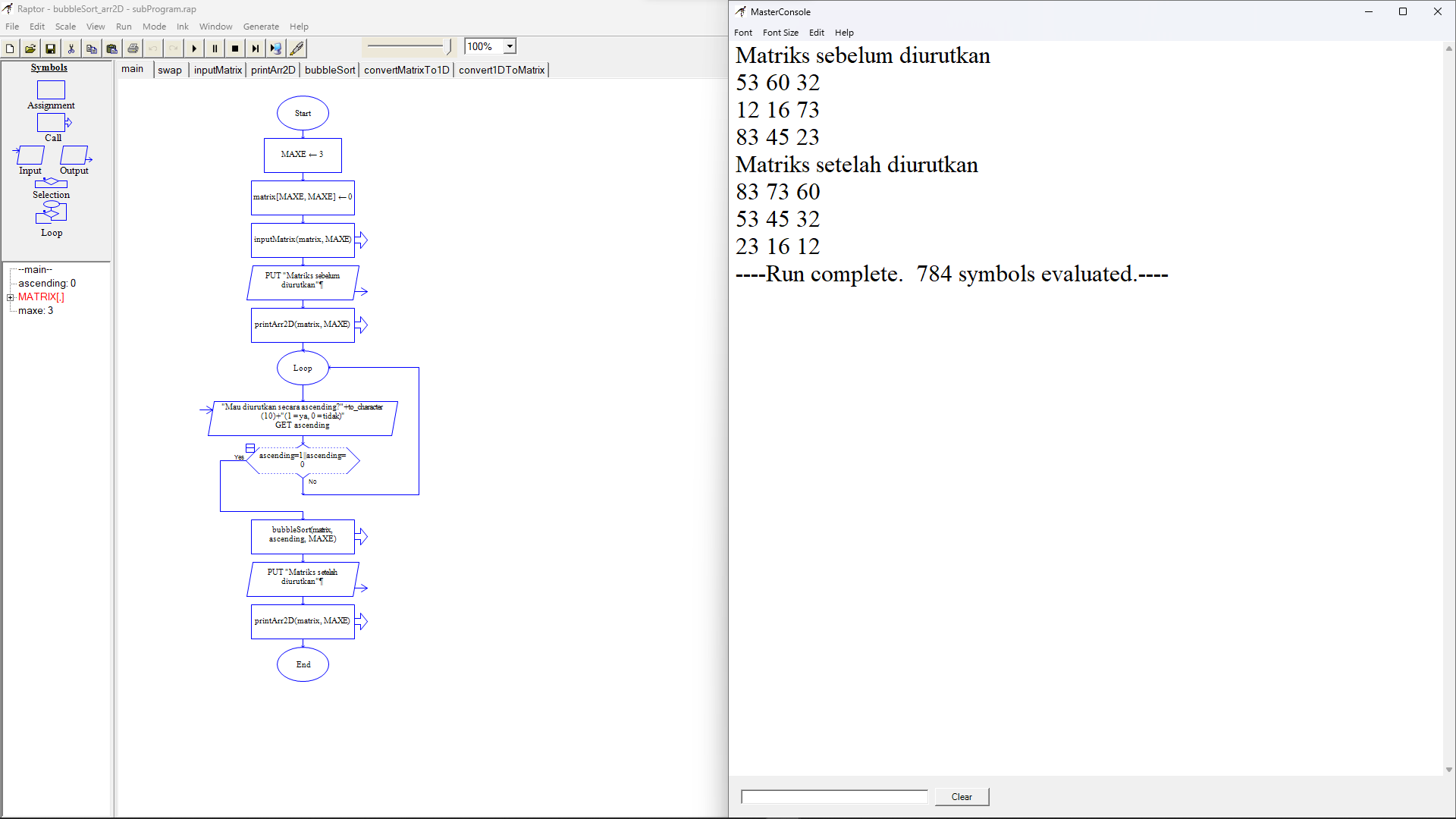
Subprogram ini digunakan untuk mengubah matriks satu dimensi kembali menjadi matriks dua dimensi setelah pengurutan selesai. Matriks dua dimensi dan matriks satu dimensi diterima sebagai parameter. Seperti subprogram sebelumnya, perulangan bersarang digunakan untuk menyalin setiap elemen dari matriks satu dimensi ke matriks dua dimensi.

Setelah definisi subprogram, program utama dimulai. Pertama, ukuran maksimal matriks ditentukan dan matriks dua dimensi yang sesuai dideklarasikan. Kemudian, subprogram "inputMatrix" digunakan untuk meminta pengguna memasukkan nilai-nilai matriks. Setelah itu, matriks sebelum diurutkan dicetak menggunakan subprogram "printArr2D". Pengguna juga diminta untuk memilih apakah pengurutan dilakukan secara menaik atau menurun melalui input variabel "ascending". Kemudian, subprogram "bubbleSort" dipanggil untuk mengurutkan matriks sesuai dengan pilihan pengguna. Setelah pengurutan selesai, matriks setelah diurutkan dicetak menggunakan subprogram "printArr2D".

Berikut adalah output flowchart Raptornya.



Gambar 7 Output secara ascending dengan flowchart subProgram. (Sumber: Penulis)



Gambar 8 Output secara ascending dengan flowchart subProgram. (Sumber: Penulis)

1. Konversikan hasil dari flowchart nomor 1 dan 2 menjadi progam C++ Hasil konversi!

## Konversi hasil flowchart nomor 1



Gambar 9 Flowvhart yang sudah dikonversikan ke dalam bahasa C++. (Sumber: Penulis)

Kode di atas adalah implementasi dalam bahasa C++ untuk mengurutkan matriks 2 dimensi berukuran 3x3 menggunakan algoritma bubble sort.

Pertama, kita mengimpor pustaka <bits/stdc++.h> yang menyediakan berbagai pustaka standar dalam C++. Kemudian, kita mendefinisikan konstanta MAX dengan nilai 3 menggunakan preprocessor directive #define.

Fungsi main merupakan titik masuk eksekusi program. Pada awalnya, kita mendeklarasikan matriks matrix berukuran 3x3 dengan tipe data int. Kemudian, kita menggunakan dua perulangan for bersarang untuk mengisi nilai matriks sesuai input pengguna. Setiap elemen matriks diminta dari pengguna dengan pesan yang sesuai.

Selanjutnya, kita menampilkan matriks sebelum diurutkan dengan menggunakan perulangan for bersarang. Nilai setiap elemen matriks ditampilkan menggunakan cout. Setelah menampilkan matriks awal, kita meminta pengguna untuk memilih apakah ingin mengurutkan secara ascending atau tidak. Hal ini dilakukan dengan perulangan while (true) yang berjalan terus menerus hingga pengguna memasukkan input yang valid (1 atau 0). Jika input tidak valid, pesan kesalahan ditampilkan dan input diabaikan.

Setelah memperoleh pilihan pengguna, kita mendeklarasikan array matrix1DTemp dengan ukuran MAX \* MAX untuk menyimpan elemen-elemen matriks dalam satu dimensi. Kemudian, kita menggunakan perulangan for bersarang untuk mengisinya. Elemen-elemen matriks asli disalin ke dalam array satu dimensi.

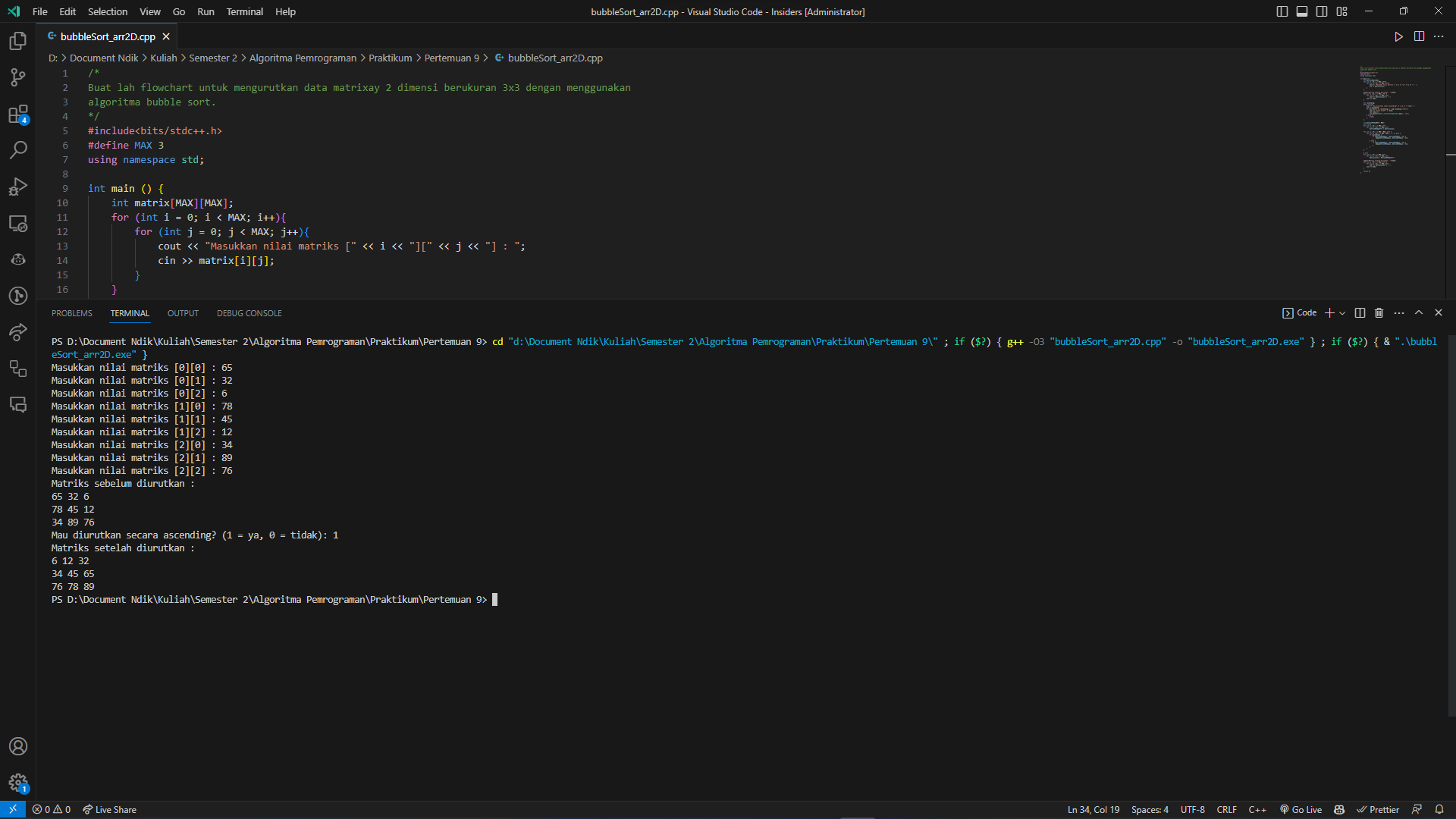
Selanjutnya, kita menerapkan algoritma bubble sort untuk mengurutkan elemen-elemen dalam matrix1DTemp. Terdapat dua perulangan for bersarang. Perulangan luar for pertama berjalan sebanyak MAX \* MAX kali, sedangkan perulangan dalam for kedua berjalan dengan batasan yang semakin berkurang pada setiap iterasi. Di dalam perulangan, kita membandingkan pasangan elemen berturut-turut dalam array. Jika elemen-elemen tersebut tidak terurut sesuai dengan kriteria ascending atau descending yang ditentukan oleh pengguna, maka elemen-elemen tersebut ditukar menggunakan fungsi swap.

Setelah selesai mengurutkan elemen-elemen, kita memindahkan kembali elemen-elemen yang telah diurutkan dari array satu dimensi matrix1DTemp ke matriks matrix dengan menggunakan perulangan for bersarang. Kita menggunakan variabel k untuk mengindeks array satu dimensi, dan setiap elemen matriks diisi dengan nilai dari matrix1DTemp.

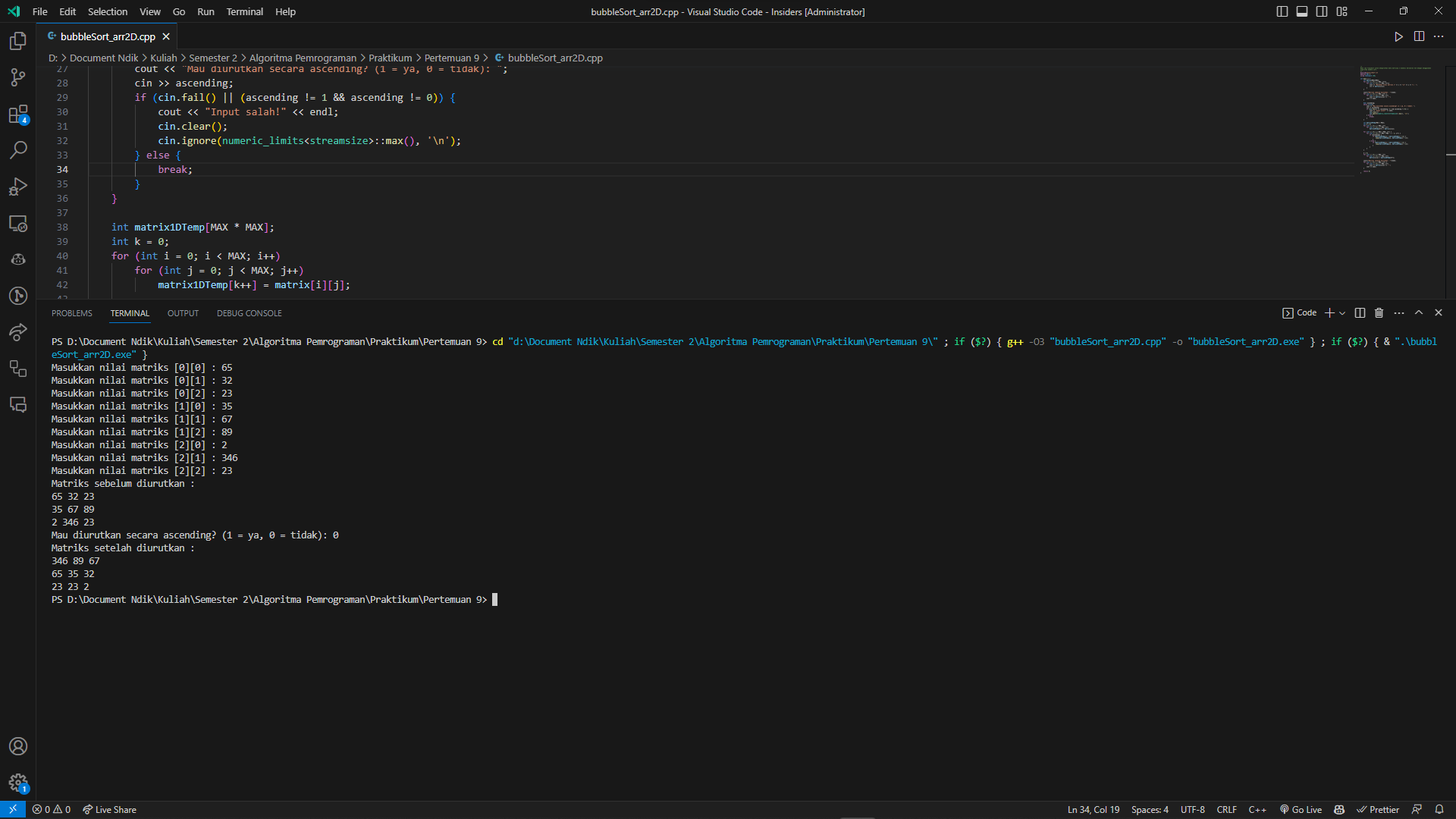
Akhirnya, kita menampilkan matriks setelah diurutkan dengan menggunakan perulangan for bersarang dan cout. Setiap elemen matriks ditampilkan, diikuti oleh karakter spasi, dan diakhiri dengan karakter baris baru. Setelah menampilkan matriks hasil, program selesai dengan mengembalikan nilai 0.

Kode di atas menggabungkan konsep dasar pemrograman C++ seperti variabel, tipe data, perulangan, penggunaan array, fungsi, dan operasi input/output untuk mengimplementasikan algoritma bubble sort pada matriks 2 dimensi.

Berikut adalah output flowchart RAPTOR yang sudah dikonversikan ke dalam bahasa C++



Gambar 10 Output secara ascending di C++. (Sumber: Penulis)



Gambar 11 Output secara descending di C++. (Sumber: Penulis)

## Konversi hasil flowchart nomor 2 (dengan subprogram)



Gambar 12 Flowchart yang sudah dikonversikan ke dalam bahasa C++ dan menggunakan subProgram. (Sumber: Penulis)

Kode di atas implementasi dalam bahasa C++ untuk mengurutkan matriks 2 dimensi berukuran 3x3 menggunakan algoritma bubble sort menggunakan subprogram atau fungsi untuk memisahkan logika program ke dalam bagian-bagian yang lebih terorganisir.

Dalam program tersebut, terdapat beberapa fungsi yang memiliki peran masing-masing. Berikut adalah daftar fungsi-fungsi yang ada dalam program beserta penjelasan rinci tentang fungsinya:

* inputMatrix(int matrix[MAX][MAX]): Fungsi ini digunakan untuk mengisi nilai matriks dengan input dari pengguna. Fungsi ini menggunakan dua perulangan for bersarang untuk mengiterasi baris dan kolom matriks. Setiap elemen matriks diminta dari pengguna melalui cout dan diambil nilai inputnya menggunakan cin. Fungsi ini tidak mengembalikan nilai.
* printArr2D(int matrix[MAX][MAX]): Fungsi ini digunakan untuk menampilkan matriks ke layar. Fungsi ini juga menggunakan dua perulangan for bersarang untuk mengiterasi semua elemen matriks. Setiap elemen matriks ditampilkan menggunakan cout, diikuti oleh karakter spasi, dan diakhiri dengan karakter baris baru. Fungsi ini tidak mengembalikan nilai.
* convertMatrixTo1D(int matrix[MAX][MAX], int matrix1DTemp[MAX \* MAX]): Fungsi ini digunakan untuk mengonversi matriks 2 dimensi menjadi array satu dimensi. Fungsi ini mengambil matriks 2 dimensi dan array satu dimensi sebagai argumen. Selanjutnya, elemen-elemen matriks disalin ke dalam array satu dimensi secara berurutan menggunakan dua perulangan for bersarang. Fungsi ini tidak mengembalikan nilai.
* convert1DToMatrix(int matrix[MAX][MAX], int matrix1DTemp[MAX \* MAX]): Fungsi ini digunakan untuk mengonversi array satu dimensi menjadi matriks 2 dimensi. Fungsi ini mengambil array satu dimensi dan matriks 2 dimensi sebagai argumen. Selanjutnya, elemen-elemen dari array satu dimensi disalin ke dalam matriks sesuai dengan urutan yang benar menggunakan dua perulangan for bersarang. Fungsi ini tidak mengembalikan nilai.
* bubbleSort(int matrix[MAX][MAX], bool ascending): Fungsi ini merupakan implementasi algoritma bubble sort untuk mengurutkan matriks. Fungsi ini menerima argumen matriks 2 dimensi dan sebuah boolean ascending yang menentukan apakah pengurutan dilakukan secara ascending atau descending. Pertama, fungsi ini menggunakan fungsi convertMatrixTo1D untuk mengonversi matriks menjadi array satu dimensi. Selanjutnya, terdapat dua perulangan for bersarang. Perulangan luar berjalan sebanyak MAX \* MAX kali, sedangkan perulangan dalam berjalan dengan batasan yang semakin berkurang pada setiap iterasi. Di dalam perulangan, fungsi ini membandingkan pasangan elemen berturut-turut dalam array. Jika elemen-elemen tersebut tidak terurut sesuai dengan kriteria ascending atau descending yang ditentukan oleh ascending, maka elemen-elemen tersebut ditukar menggunakan fungsi swap. Setelah selesai mengurutkan elemen-elemen, fungsi ini menggunakan fungsi convert1DToMatrix untuk mengonversi kembali array satu dimensi ke matriks.
* main(): Fungsi main merupakan fungsi utama yang menjalankan alur program. Pada fungsi ini, pertama-tama matriks diisi menggunakan fungsi inputMatrix. Kemudian, matriks sebelum diurutkan ditampilkan ke layar menggunakan fungsi printArr2D. Selanjutnya, pengguna diminta untuk memilih apakah pengurutan akan dilakukan secara ascending atau descending dengan menggunakan perulangan while dan validasi input menggunakan cin.fail(). Setelah memperoleh input yang valid, fungsi bubbleSort dipanggil untuk mengurutkan matriks. Terakhir, matriks yang sudah diurutkan ditampilkan ke layar menggunakan fungsi printArr2D.

Fungsi inputMatrix digunakan untuk mengisi nilai matriks dengan input dari pengguna. Pada fungsi ini, terdapat dua perulangan for bersarang yang berjalan sebanyak MAX kali untuk mengiterasi baris dan kolom matriks. Setiap elemen matriks diminta dari pengguna melalui cout dan diambil nilai inputnya menggunakan cin.

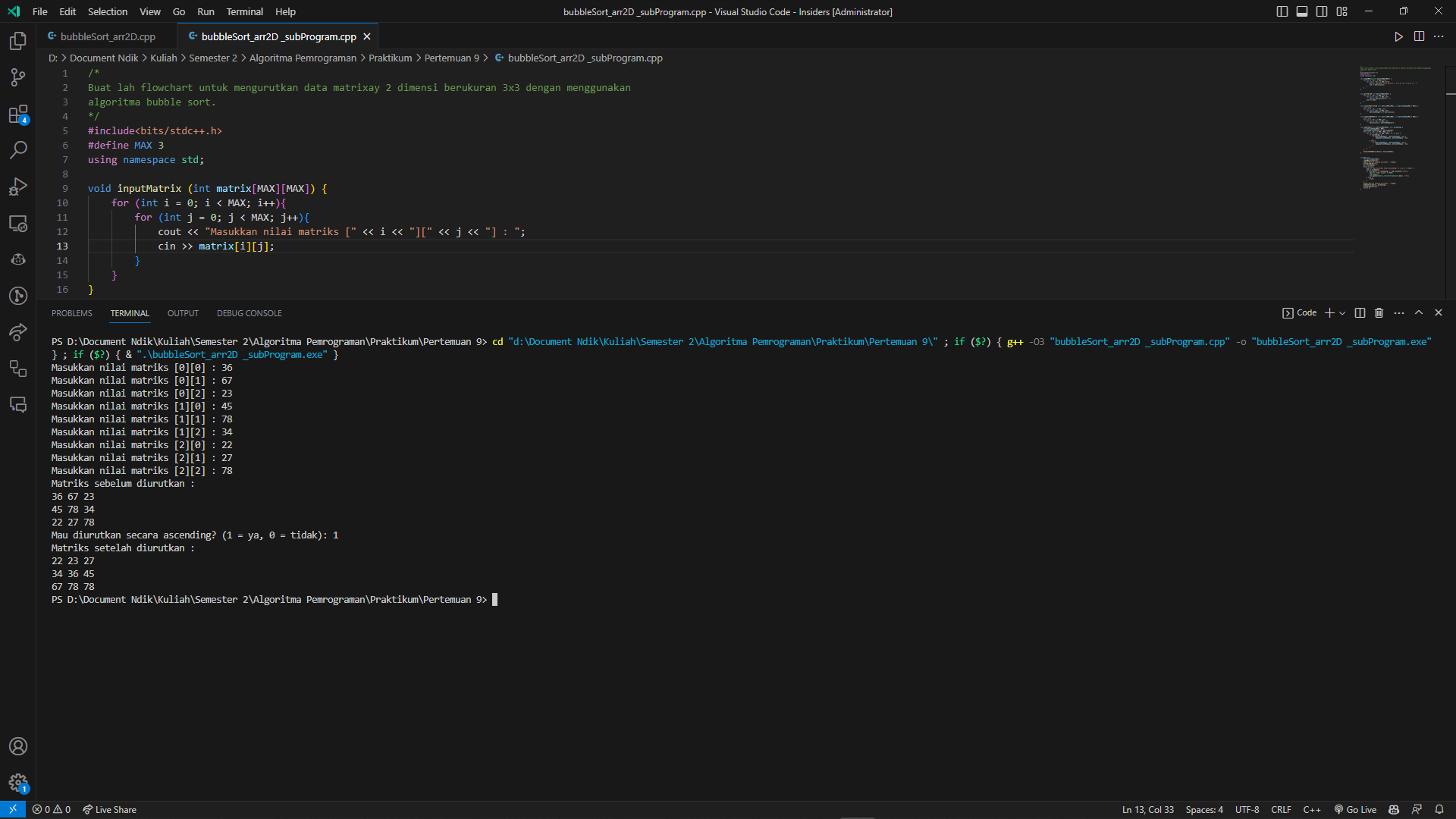
Fungsi printArr2D digunakan untuk menampilkan matriks ke layar. Sama seperti fungsi inputMatrix, fungsi ini juga menggunakan dua perulangan for bersarang untuk mengiterasi semua elemen matriks. Setiap elemen matriks ditampilkan menggunakan cout, diikuti oleh karakter spasi, dan diakhiri dengan karakter baris baru.

Fungsi convertMatrixTo1D dan convert1DToMatrix digunakan untuk mengonversi matriks 2 dimensi menjadi array satu dimensi, dan sebaliknya. Fungsi convertMatrixTo1D mengambil matriks 2 dimensi dan array satu dimensi sebagai argumen, dan mengisi array satu dimensi dengan elemen-elemen matriks secara berurutan. Sedangkan fungsi convert1DToMatrix melakukan sebaliknya, yaitu mengambil array satu dimensi dan matriks 2 dimensi sebagai argumen, dan mengisi matriks dengan elemen-elemen dari array satu dimensi sesuai urutan.

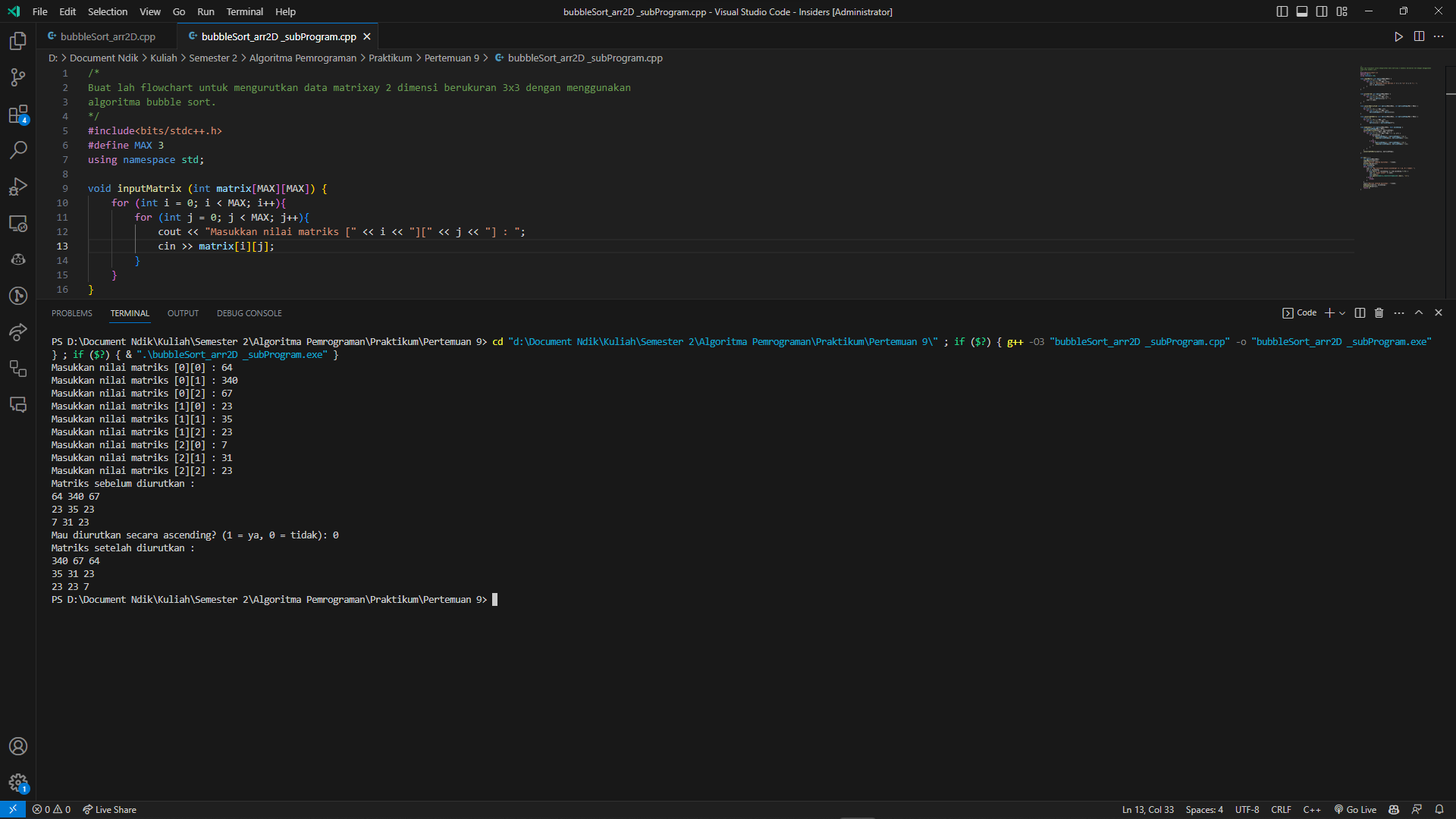
Fungsi bubbleSort merupakan implementasi algoritma bubble sort untuk mengurutkan matriks. Fungsi ini menerima argumen matriks 2 dimensi dan sebuah boolean ascending yang menentukan apakah pengurutan dilakukan secara ascending atau descending. Di dalam fungsi ini, kita mengonversi matriks ke array satu dimensi menggunakan fungsi convertMatrixTo1D. Kemudian, terdapat dua perulangan for bersarang. Perulangan luar berjalan sebanyak MAX \* MAX kali, sedangkan perulangan dalam berjalan dengan batasan yang semakin berkurang pada setiap iterasi. Di dalam perulangan, kita membandingkan pasangan elemen berturut-turut dalam array. Jika elemen-elemen tersebut tidak terurut sesuai dengan kriteria ascending atau descending yang ditentukan oleh ascending, maka elemen-elemen tersebut ditukar menggunakan fungsi swap. Setelah selesai mengurutkan elemen-elemen, kita mengonversi kembali array satu dimensi ke matriks menggunakan fungsi convert1DToMatrix.

Selanjutnya, pada fungsi main, kita mendeklarasikan matriks matrix berukuran 3x3. Kemudian, kita memanggil fungsi inputMatrix untuk mengisi nilai matriks dari pengguna. Setelah itu, kita menampilkan matriks sebelum diurutkan menggunakan fungsi printArr2D. Selanjutnya, kita mengambil input dari pengguna untuk menentukan apakah pengurutan dilakukan secara ascending atau descending. Input tersebut dicek menggunakan perulangan while dan fungsi cin.fail() untuk memastikan bahwa input yang dimasukkan adalah angka 0 atau 1. Jika input tidak valid, pesan kesalahan ditampilkan dan input diabaikan menggunakan cin.clear() dan cin.ignore(). Setelah mendapatkan input yang valid, kita memanggil fungsi bubbleSort untuk mengurutkan matriks, dan kemudian menampilkan matriks yang sudah diurutkan menggunakan fungsi printArr2D.

Berikut adalah contoh output programnya yang menggunakan subprogram.



Gambar 13 Diurutkan secara ascending. (Sumber: Penulis)



Gambar 14 Diurutkan secara descending. (Sumber: Penulis)

Untuk mengakses flowchart RAPTOR maupun kodingan, dapat mengakses sourcecode di link gihthub berikut.

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