**JAWABAN UAS PRAKTIKUM ALGORITMA PEMROGRAMAN**

**KELAS E**

Nama: MUHAMMAD FATIH MAULANA

NIM: 1237050140

Kelas: IF2-E Praktikum Algoritma Pemrograman

26 Juni 2024

1. Membuat List Berdasarkan Fungsi Tambah\_Produk (Yang Bisa Diinput Oleh User)

using System;

using System.Collections.Generic;

namespace TokoElektronikABC

{

class Program

{

// List produk elektronik

static List<Dictionary<string, object>> listProduk = new List<Dictionary<string, object>>();

static void Main(string[] args)

{

bool running = true;

while (running)

{

Console.WriteLine("\nMenu:");

Console.WriteLine("1. Tambah Produk");

Console.WriteLine("2. Tampilkan Daftar Produk");

Console.WriteLine("3. Cari Produk Berdasarkan Kode");

Console.WriteLine("4. Hitung Produk Berdasarkan Kategori");

Console.WriteLine("5. Perbarui Produk");

Console.WriteLine("6. Keluar");

Console.Write("Pilih opsi (1-6): ");

string pilihan = Console.ReadLine();

switch (pilihan)

{

case "1":

TambahProdukDariUser();

break;

case "2":

DaftarProduk();

break;

case "3":

CariProdukDariUser();

break;

case "4":

HitungProdukKategoriDariUser();

break;

case "5":

PerbaruiProdukDariUser();

break;

case "6":

running = false;

break;

default:

Console.WriteLine("Pilihan tidak valid. Silakan pilih opsi yang benar.");

break;

}

}

}

// Fungsi untuk menambahkan produk baru dari input user

static void TambahProdukDariUser()

{

Console.Write("Masukkan kode produk: ");

string kode = Console.ReadLine();

Console.Write("Masukkan nama produk: ");

string nama = Console.ReadLine();

Console.Write("Masukkan kategori produk: ");

string kategori = Console.ReadLine();

Console.Write("Masukkan stok produk: ");

int stok = int.Parse(Console.ReadLine());

TambahProduk(kode, nama, kategori, stok);

}

// Fungsi untuk menambahkan produk baru

static void TambahProduk(string kode, string nama, string kategori, int stok)

{

// Membuat dictionary baru untuk produk

Dictionary<string, object> produkBaru = new Dictionary<string, object>

{

{ "kode", kode },

{ "nama", nama },

{ "kategori", kategori },

{ "stok", stok }

};

// Menambahkan produk baru ke dalam list

listProduk.Add(produkBaru);

Console.WriteLine($"Produk dengan kode {kode} berhasil ditambahkan.");

}

Penjelasan Main:

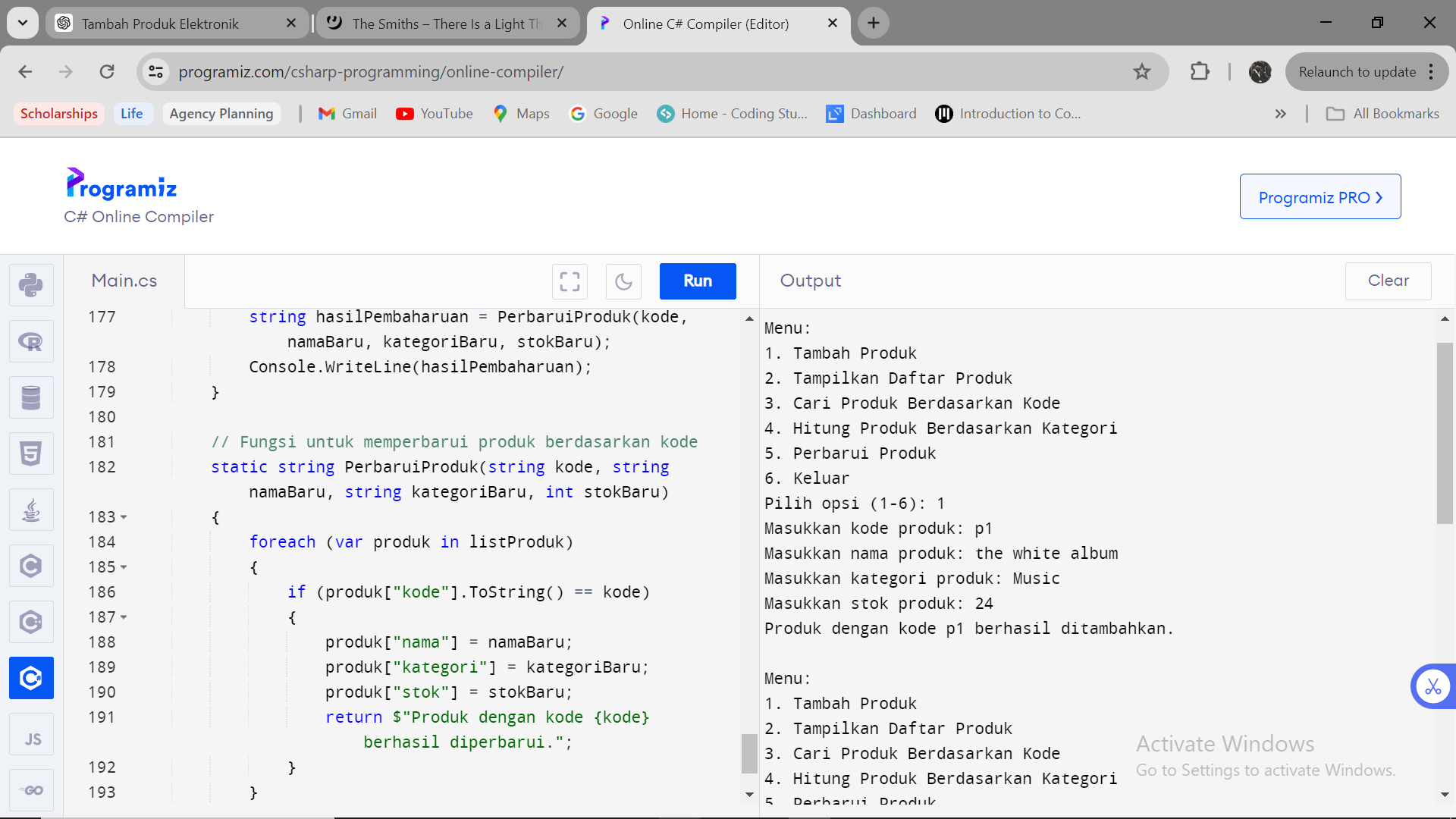
Fungsi ini adalah titik masuk utama dari program. Ini menjalankan loop utama yang menampilkan menu dan memungkinkan pengguna untuk memilih berbagai opsi seperti menambah produk, menampilkan daftar produk, mencari produk berdasarkan kode, menghitung produk berdasarkan kategori, memperbarui produk, atau keluar dari program.

Penjelasan TambahProdukDariUser:

Fungsi ini meminta input dari pengguna untuk atribut produk seperti kode, nama, kategori, dan stok, kemudian memanggil fungsi TambahProduk untuk menambahkan produk baru ke dalam list listProduk.

Penjelasn TambahProduk:

Fungsi ini membuat dictionary baru untuk produk berdasarkan parameter yang diberikan (kode, nama, kategori, stok) dan menambahkannya ke list listProduk. Juga menampilkan pesan konfirmasi bahwa produk berhasil ditambahkan.

Output:

1. Menampilkan Semua Produk Dengan Prosedur Daftar\_Produk

// Prosedur untuk menampilkan daftar produk

static void DaftarProduk()

{

if (listProduk.Count == 0)

{

Console.WriteLine("Tidak ada produk yang tersedia.");

return;

}

Console.WriteLine("Daftar Produk:");

foreach (var produk in listProduk)

{

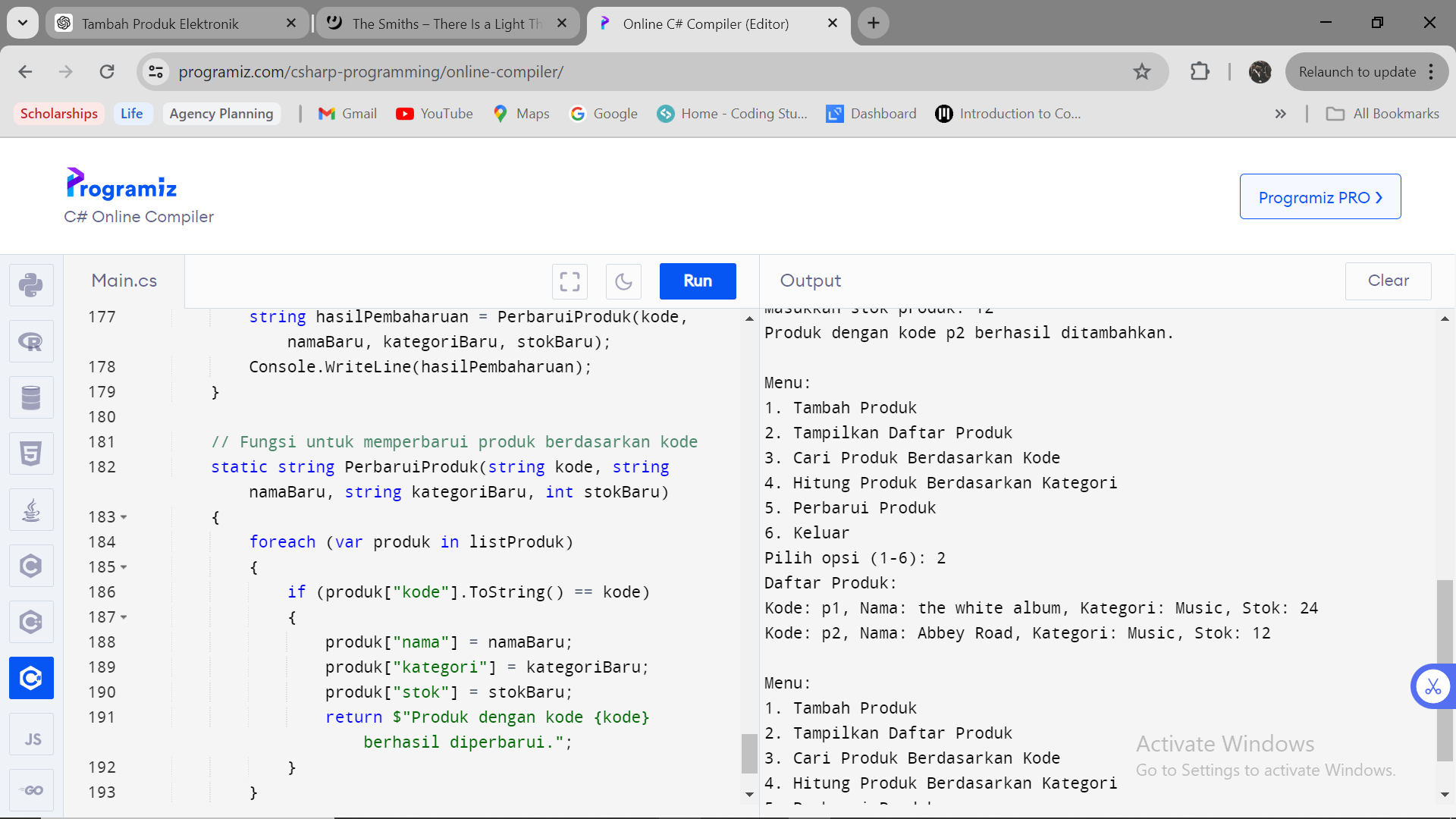
Console.WriteLine($"Kode: {produk["kode"]}, Nama: {produk["nama"]}, Kategori: {produk["kategori"]}, Stok: {produk["stok"]}");

}

}

Penjelasan DaftarProduk:

Fungsi ini menampilkan semua produk yang ada dalam list listProduk. Jika tidak ada produk dalam list, fungsi ini menampilkan pesan bahwa tidak ada produk yang tersedia.

Output:

1. Function Cari\_Produk Untuk Mencari Dan Menampilkan Produk Sesuai Code

// Fungsi untuk mencari produk berdasarkan kode dari input user

static void CariProdukDariUser()

{

Console.Write("Masukkan kode produk yang ingin dicari: ");

string kode = Console.ReadLine();

var hasilPencarian = CariProduk(kode);

if (hasilPencarian != null)

{

Console.WriteLine("Produk ditemukan:");

Console.WriteLine($"Kode: {hasilPencarian["kode"]}, Nama: {hasilPencarian["nama"]}, Kategori: {hasilPencarian["kategori"]}, Stok: {hasilPencarian["stok"]}");

}

else

{

Console.WriteLine("Produk tidak ditemukan.");

}

}

// Fungsi untuk mencari produk berdasarkan kode

static Dictionary<string, object> CariProduk(string kode)

{

foreach (var produk in listProduk)

{

if (produk["kode"].ToString() == kode)

{

return produk;

}

}

return null;

}

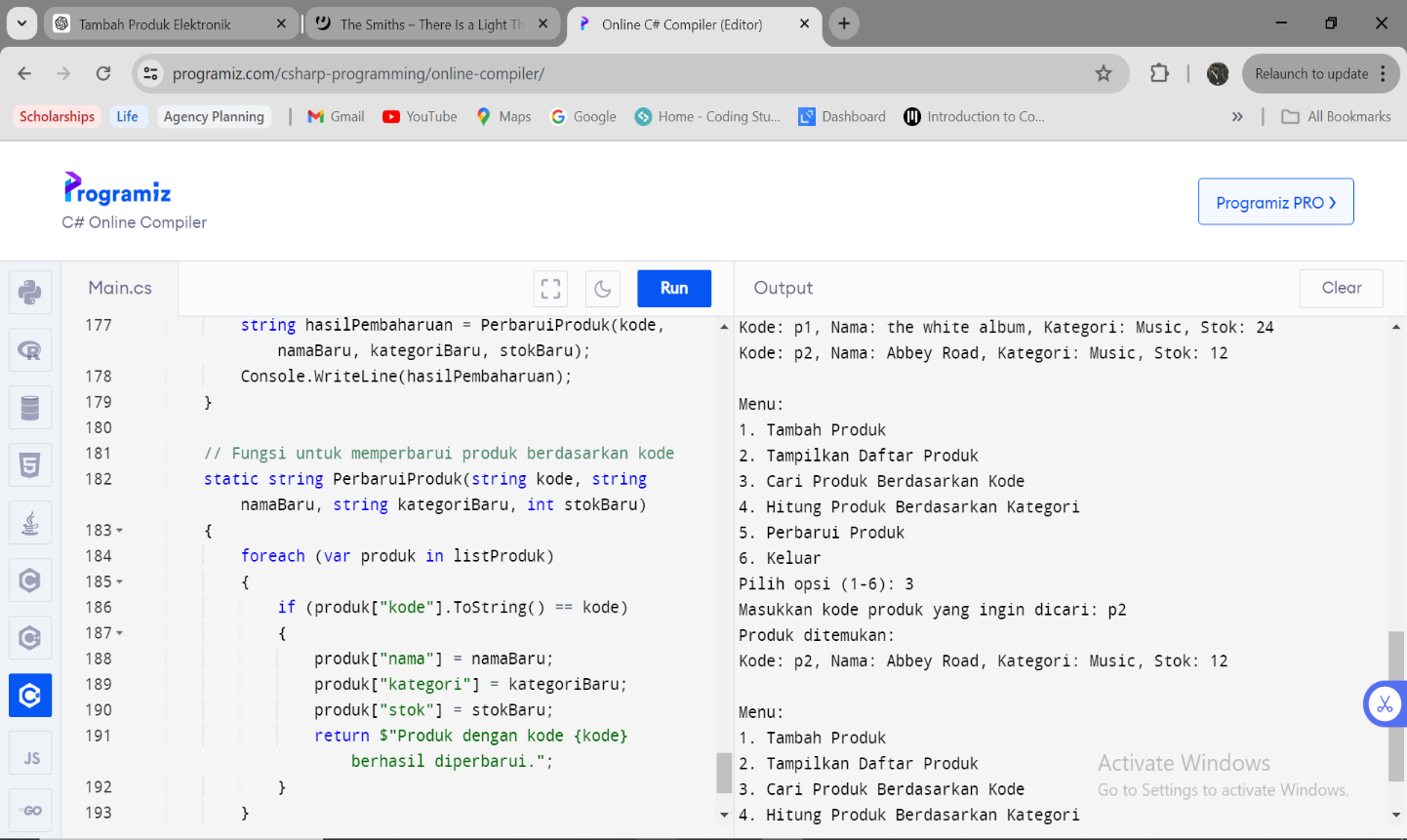
Penjelasan CariProdukDariUser:

Fungsi ini meminta kode produk dari pengguna, memanggil fungsi CariProduk untuk mencari produk berdasarkan kode, dan menampilkan informasi produk jika ditemukan. Jika produk tidak ditemukan, fungsi ini menampilkan pesan yang sesuai.

Penjelasan CariProduk:

Fungsi ini mencari produk dalam list listProduk berdasarkan kode yang diberikan. Jika ditemukan, fungsi ini mengembalikan dictionary produk tersebut; jika tidak, fungsi ini mengembalikan null.

Output:



1. Menghitung Ada Berapa Jumlah Produk Didalam Satu Kategori

// Fungsi untuk menghitung jumlah produk dalam kategori tertentu dari input user

static void HitungProdukKategoriDariUser()

{

Console.Write("Masukkan kategori produk yang ingin dihitung: ");

string kategori = Console.ReadLine();

int jumlah = HitungProdukKategori(kategori);

Console.WriteLine($"Jumlah produk dalam kategori '{kategori}': {jumlah}");

}

// Fungsi untuk menghitung jumlah produk dalam kategori tertentu

static int HitungProdukKategori(string kategori)

{

int jumlah = 0;

foreach (var produk in listProduk)

{

if (produk["kategori"].ToString() == kategori)

{

jumlah++;

}

}

return jumlah;

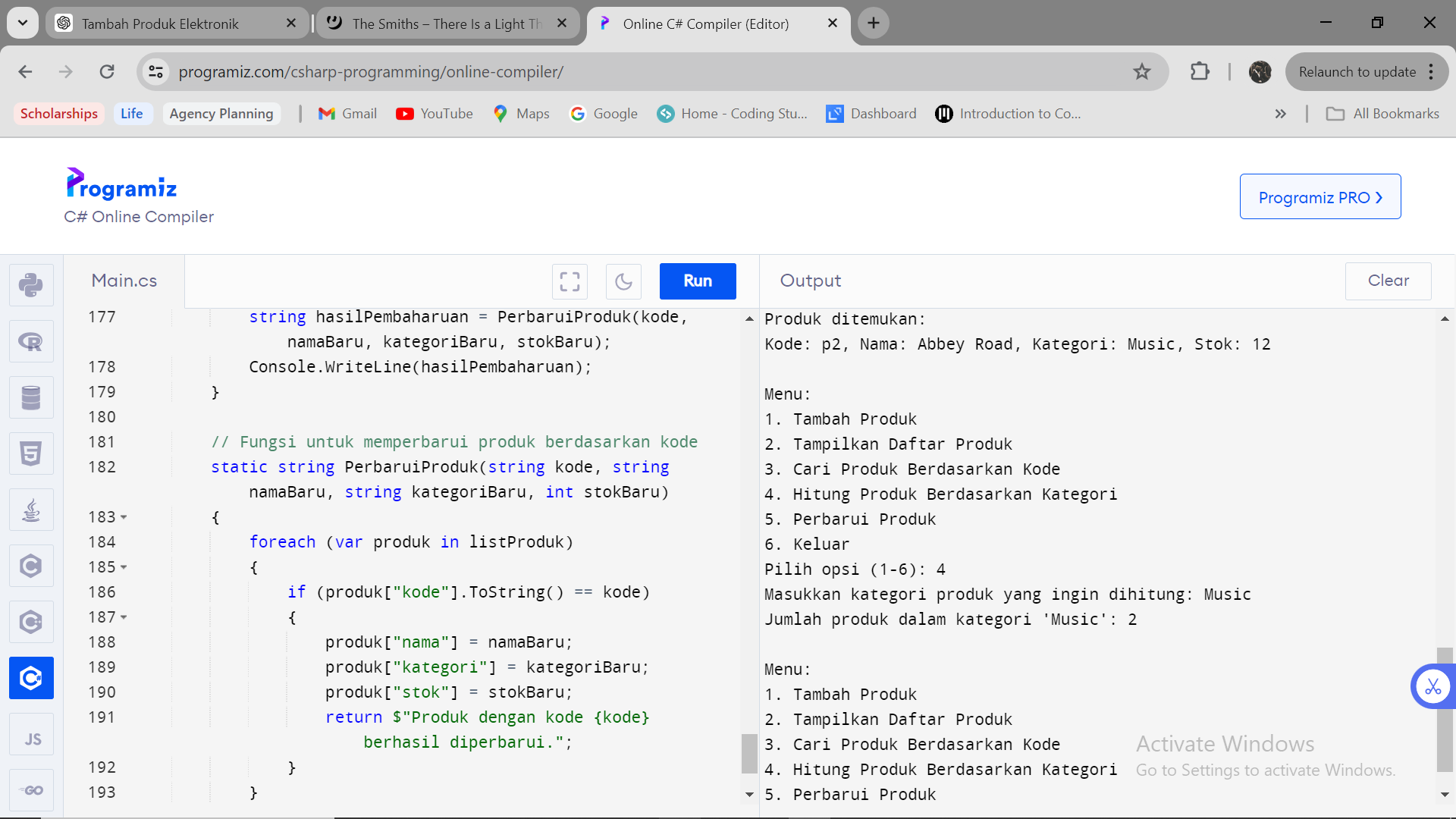
}

Penjelasan HitungProdukKategoriDariUser:

Fungsi ini meminta kategori produk dari pengguna, memanggil fungsi HitungProdukKategori untuk menghitung jumlah produk dalam kategori tersebut, dan menampilkan jumlahnya.

Penjelasan HitungProdukKategori:

Fungsi ini menghitung jumlah produk dalam list listProduk yang sesuai dengan kategori yang diberikan. Fungsi ini menggunakan loop untuk menghitung dan mengembalikan jumlah produk yang cocok.

Output:

1. Fungsi Perbarui\_Produk Untuk Update Isi List Produk Berdasarkan Kode Produk

// Fungsi untuk memperbarui produk berdasarkan kode dari input user

static void PerbaruiProdukDariUser()

{

Console.Write("Masukkan kode produk yang ingin diperbarui: ");

string kode = Console.ReadLine();

Console.Write("Masukkan nama baru produk: ");

string namaBaru = Console.ReadLine();

Console.Write("Masukkan kategori baru produk: ");

string kategoriBaru = Console.ReadLine();

Console.Write("Masukkan stok baru produk: ");

int stokBaru = int.Parse(Console.ReadLine());

string hasilPembaharuan = PerbaruiProduk(kode, namaBaru, kategoriBaru, stokBaru);

Console.WriteLine(hasilPembaharuan);

}

// Fungsi untuk memperbarui produk berdasarkan kode

static string PerbaruiProduk(string kode, string namaBaru, string kategoriBaru, int stokBaru)

{

foreach (var produk in listProduk)

{

if (produk["kode"].ToString() == kode)

{

produk["nama"] = namaBaru;

produk["kategori"] = kategoriBaru;

produk["stok"] = stokBaru;

return $"Produk dengan kode {kode} berhasil diperbarui.";

}

}

return $"Produk dengan kode {kode} tidak ditemukan.";

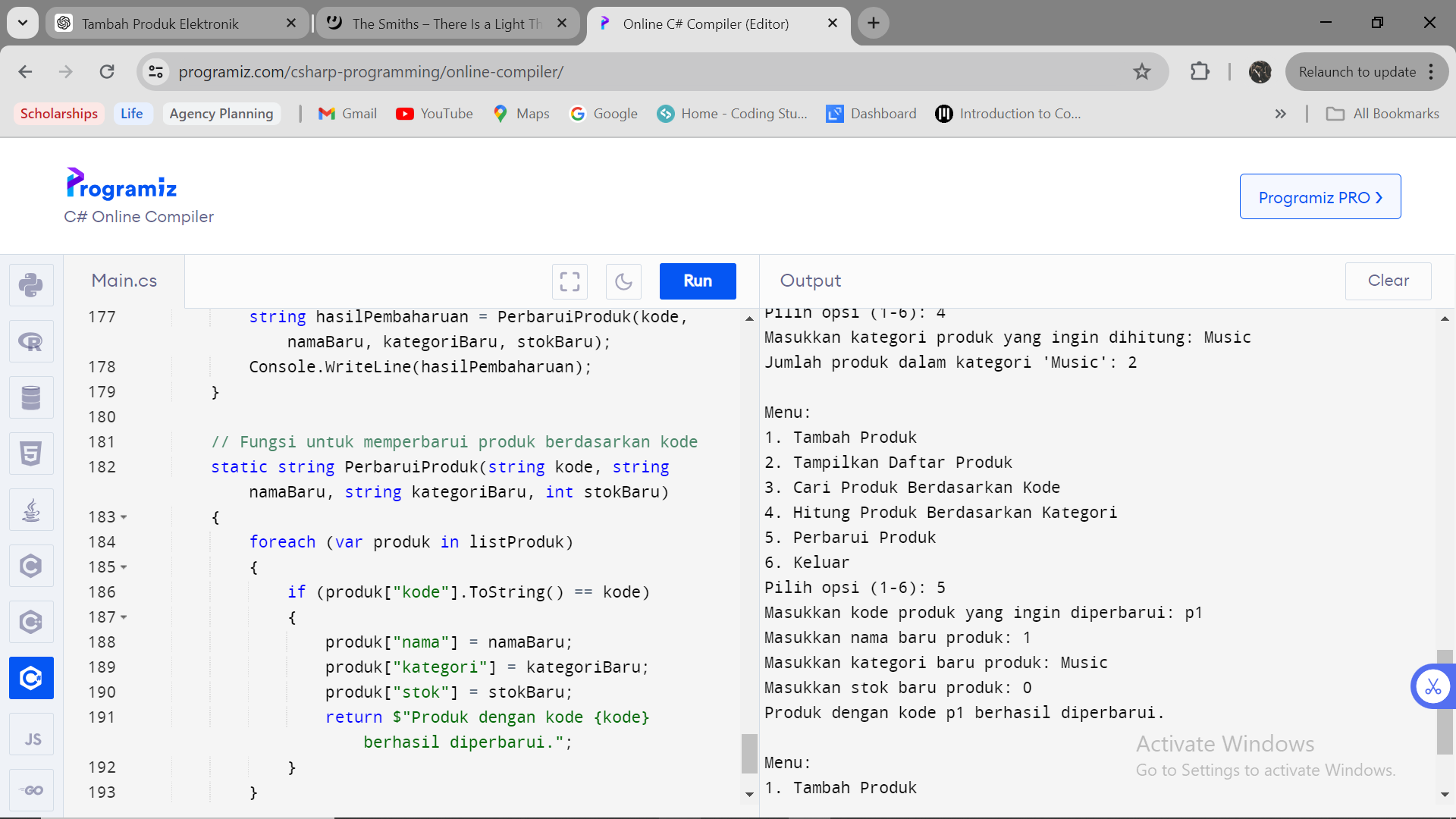
}

}

}

Penjelasan function PerbaruiProduk:

Fungsi ini mencari produk dalam list listProduk berdasarkan kode yang diberikan. Jika ditemukan, fungsi ini memperbarui atribut nama, kategori, dan stok produk dengan nilai baru yang diberikan dan menampilkan pesan konfirmasi. Jika produk tidak ditemukan, fungsi ini menampilkan pesan bahwa produk tidak ditemukan.

Output: