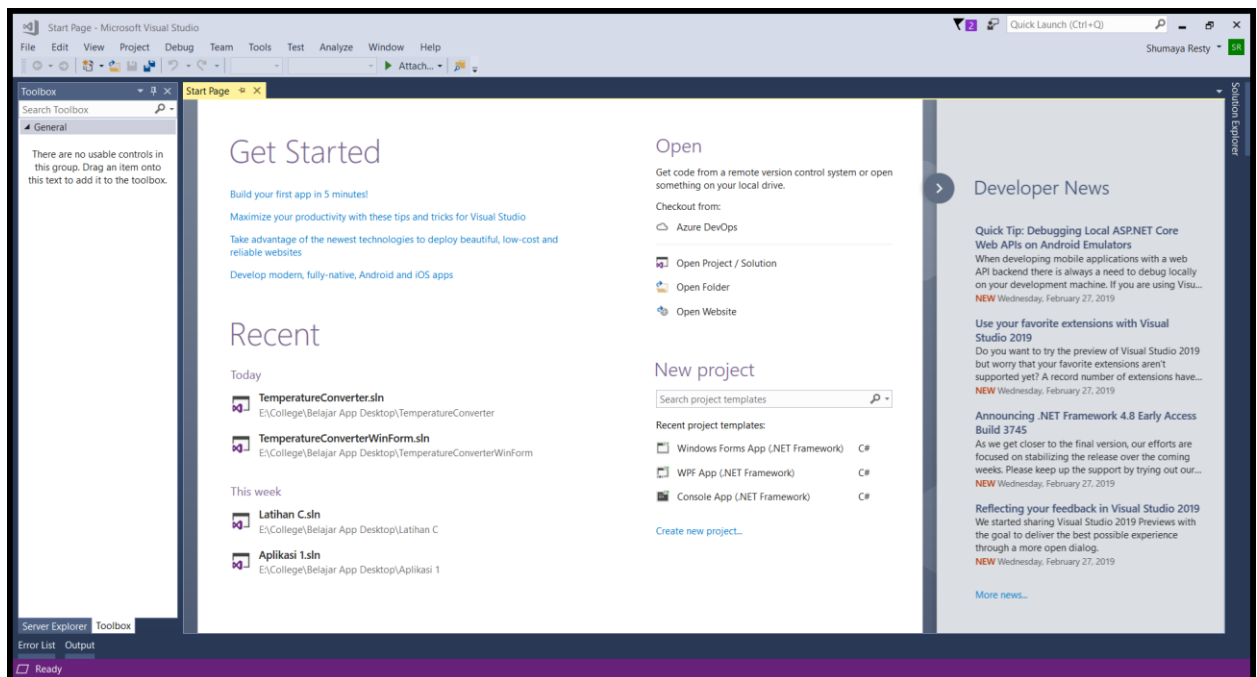


Pertemuan 2

Visual Studio

Microsoft Visual Studio adalah sebuah IDE yang dapat digunakan untuk mengembangkan aplikasi berbasis desktop. Akan tetapi, kegunaan Visual Studio tidak hanya sebatas itu. Aplikasi Web, aplikasi mobile, aplikasi web, dan juga web service juga dapat dibentuk dengan menggunakan VS sebagai media pengembangan. Visual Studio memiliki kode editor yang kita gunakan untuk menulis kode aplikasi. Hebatnya, VS mendukung sekitar 36 bahasa pemrograman berbeda seperti C, C#, C++, VB.Net, JS, HTML, XML, Python, Ruby, dll. Saat ini, Visual Studio 2017 merupakan versi terbaru yang dirilis.

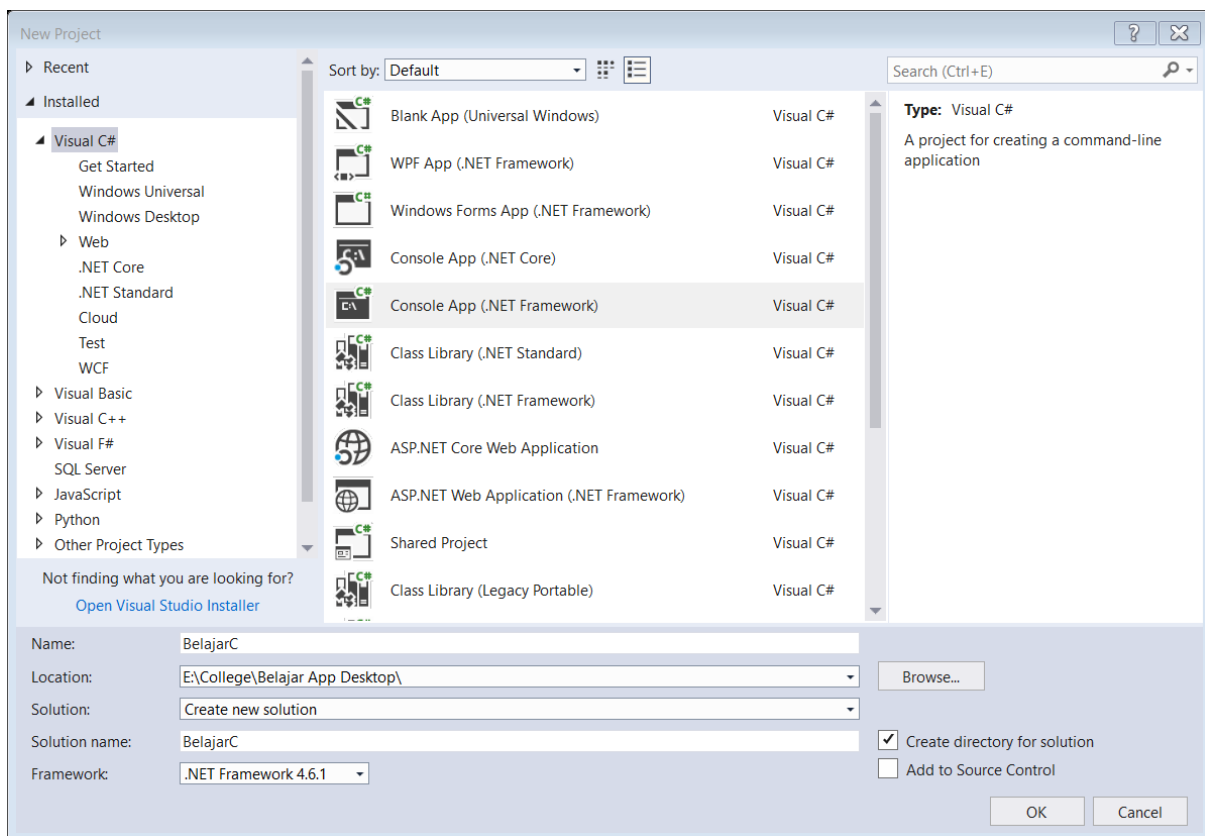


Visual Studio menggunakan platform pengembangan milik Microsoft seperti Windows API, Windows Forms, Windows Presentation Foundation (WPF), Windows Store dan Microsoft Silverlight. Adapun WinForm dan WPF akan kita pelajari pada modul ini.

Bahasa Pemrograman C#

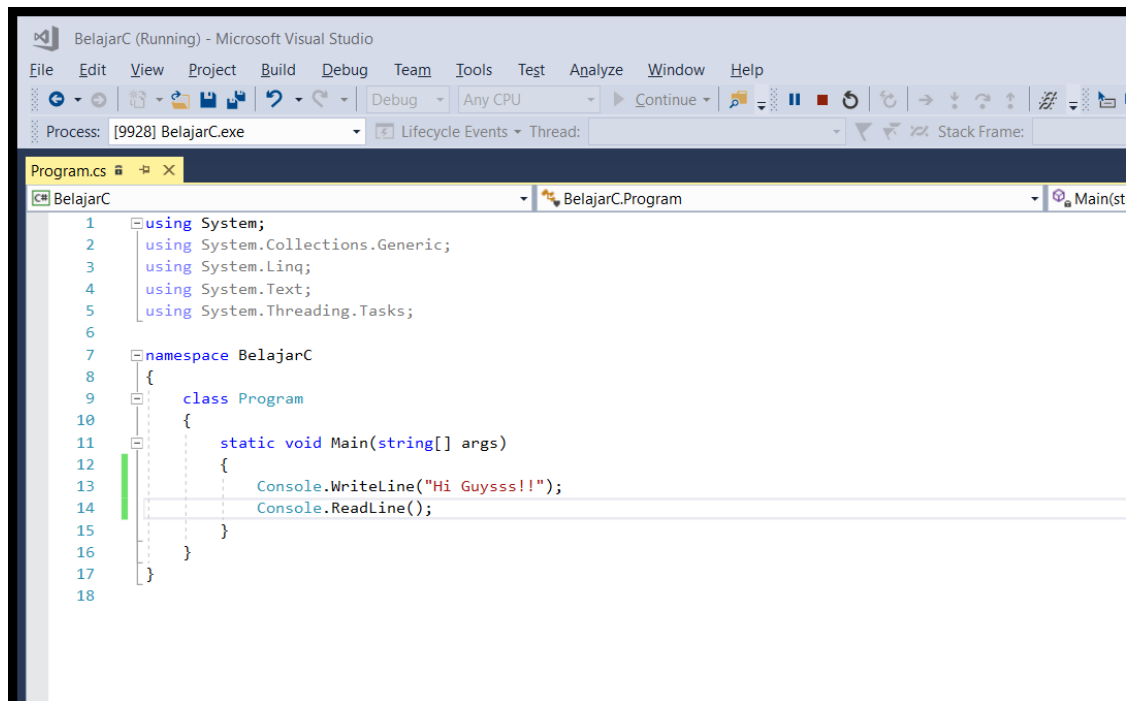
Jika C bukanlah bahasa pemrograman pertama yang dipelajari dan kita telah familiar dengan bahasa pemrograman Java, maka C# akan terlihat serupa dan tidak begitu sulit untuk dipahami. Untuk mempelajari Bahasa C#, kita akan menggunakan Console App (.Net Framework) yang telah disediakan oleh Visual Studio 2017.

Klik New → Project → Console App (.Net Framework). Lalu ganti Name menjadi „BelajarC“. Silahkan pilih atau buat folder penyimpanan masing-masing. Lalu arahkan location ke folder tersebut dan klik OK.

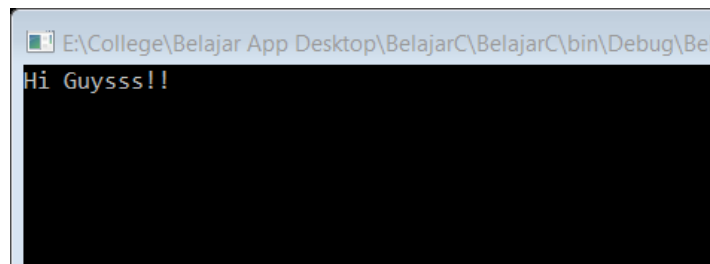


Maka halaman Program.cs akan terbuka. Secara default, akan tersedia beberapa baris kode dasar yang ditampilkan pada halaman. Akan tetapi, tidak terjadi hal yang signifikan. Silahkan tekan F5 untuk menjalankan aplikasi, maka layar hitam akan muncul selama beberapa saat dan kemudian kembali tertutup.

Untuk menampilkan teks pada layar hitam tersebut, kita dapat menambahkan dua baris kode seperti pada gambar di bawah ini.



Kemudian kembali jalankan aplikasi. Jika berhasil, maka akan muncul kotak hitam bertulisan seperti pada gambar di bawah ini.



Variables

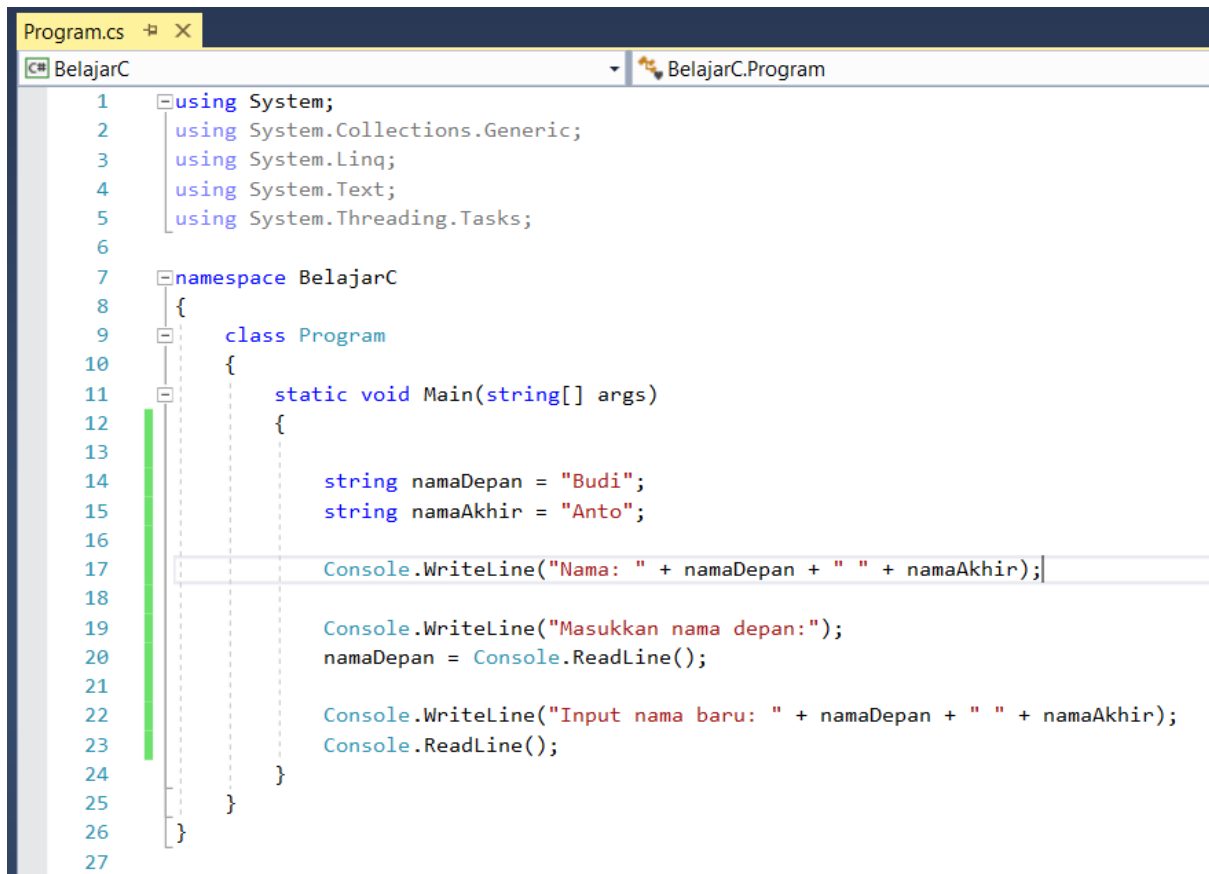
Penulisan variabel serupa dengan penulisan pada Java, yaitu < tipe data > < namaVar >;

Jika ingin langsung memberikan nilai pada variabel, maka penulisan menjadi :

< tipe data > < namaVar > = < value >; **atau** private string nama = "Indah";

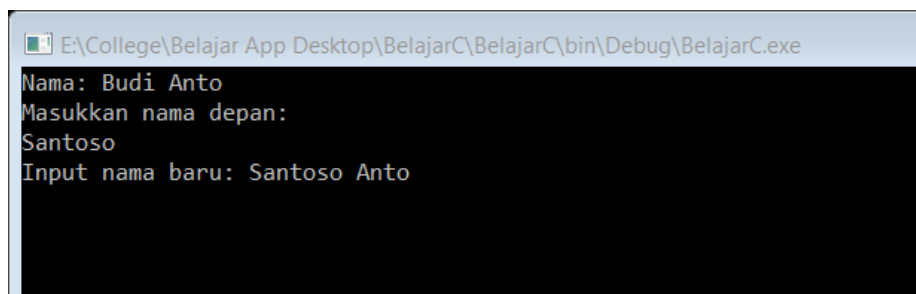
Silahkan buat proyek baru dengan menggunakan Console seperti cara di atas atau gunakan file yang sama seperti sebelumnya.

Tambahkan kode program seperti gambar di bawah ini. Amati apa yang terjadi.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace BelajarC
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13
14            string namaDepan = "Budi";
15            string namaAkhir = "Anto";
16
17            Console.WriteLine("Nama: " + namaDepan + " " + namaAkhir);
18
19            Console.WriteLine("Masukkan nama depan:");
20            namaDepan = Console.ReadLine();
21
22            Console.WriteLine("Input nama baru: " + namaDepan + " " + namaAkhir);
23            Console.ReadLine();
24        }
25    }
26 }
27
```

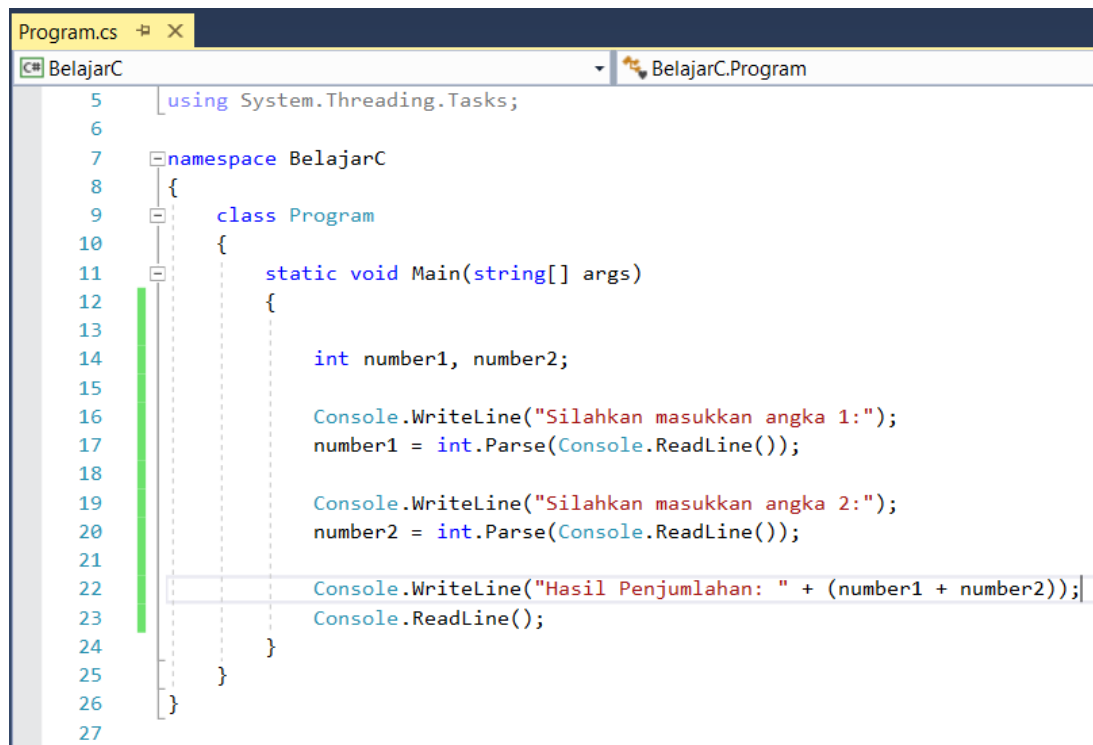
Jika kode program ditulis dengan benar, maka ketika dijalankan akan menghasilkan output seperti pada gambar di bawah ini.



```
E:\College\Belajar App Desktop\BelajarC\BelajarC\bin\Debug\BelajarC.exe
Nama: Budi Anto
Masukkan nama depan:
Santoso
Input nama baru: Santoso Anto
```

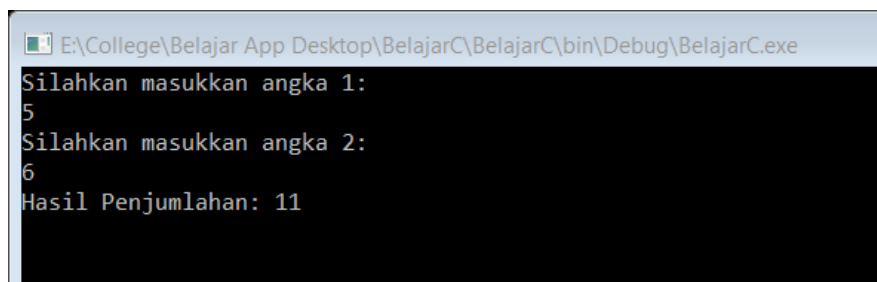
Silahkan buat projek baru dengan menggunakan Console seperti cara di atas atau gunakan file yang sama seperti sebelumnya. Untuk membuat komen pada baris kode, cukup gunakan **//baris_kode** atau **/*kode*/**.

Pada percobaan kali ini kita akan melakukan penjumlahan atas inputan (berupa angka) dari pengguna aplikasi. Silahkan ketikkan kode program seperti pada gambar berikut.



```
5  using System.Threading.Tasks;
6
7  namespace BelajarC
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13
14             int number1, number2;
15
16             Console.WriteLine("Silahkan masukkan angka 1:");
17             number1 = int.Parse(Console.ReadLine());
18
19             Console.WriteLine("Silahkan masukkan angka 2:");
20             number2 = int.Parse(Console.ReadLine());
21
22             Console.WriteLine("Hasil Penjumlahan: " + (number1 + number2));
23             Console.ReadLine();
24         }
25     }
26 }
27
```

Adapun cara pendeklarasian variabel dengan tipe data yang sama dapat dilakukan dalam sebaris kode, hal ini cukup serupa dengan bahasa pemrograman Java yang telah dipelajari. Jika kode program telah ditulis dengan benar, maka jalankan aplikasi dan akan muncul hasil seperti di bawah ini.

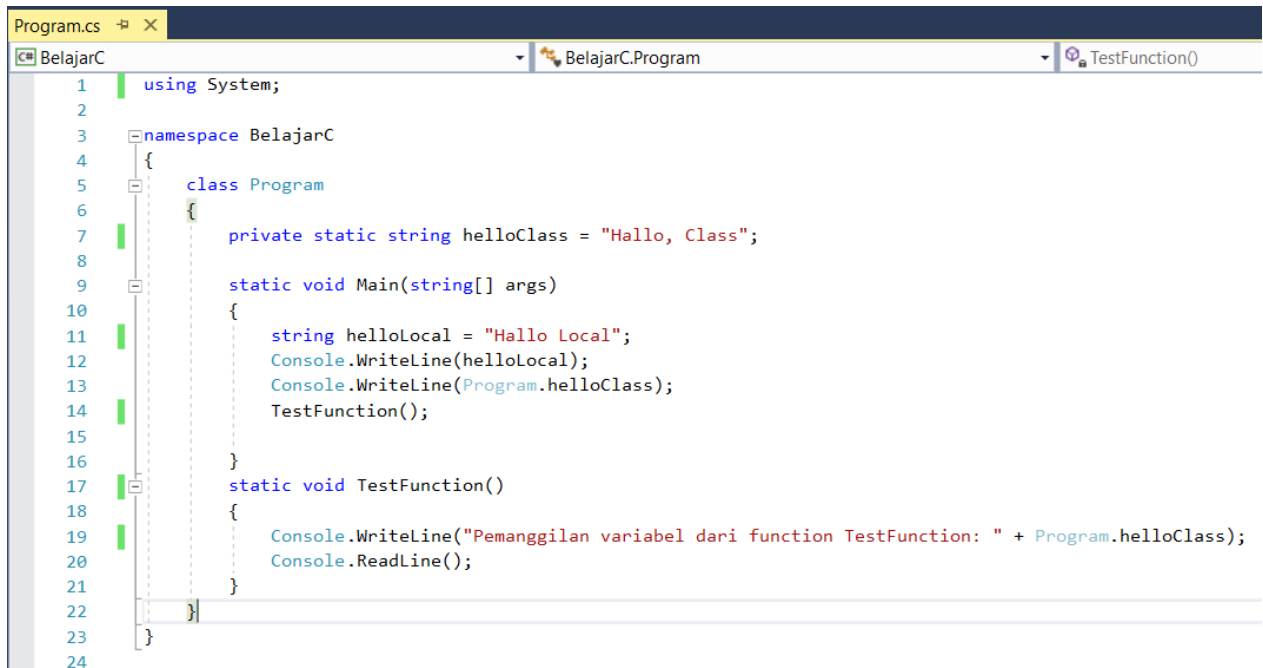


```
E:\College\Belajar App Desktop\BelajarC\BelajarC\bin\Debug\BelajarC.exe
Silahkan masukkan angka 1:
5
Silahkan masukkan angka 2:
6
Hasil Penjumlahan: 11
```

Function

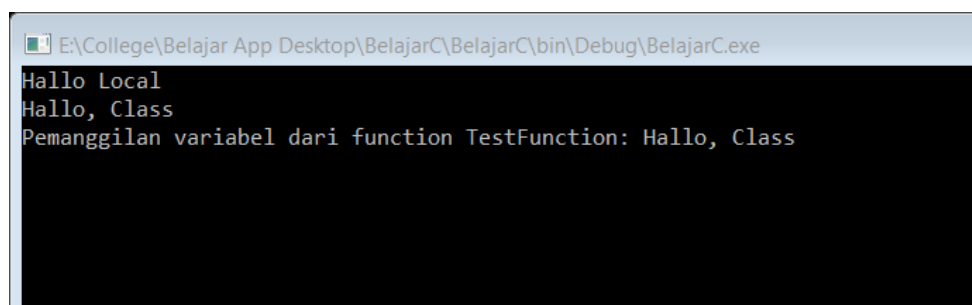
Function memungkinkan kita untuk melakukan enkapsulasi kode dan menggunakannya kembali saat dibutuhkan. Selain membuat penulisan kode yang memiliki fungsi dan guna yang sama sekali saja, function juga memudahkan programmer untuk melakukan modifikasi/perubahan terhadap kode tersebut tanpa harus bersusah payah.

Kali ini kita akan mencoba mempelajari sebuah function void yang berarti bahwa fungsi tersebut tidak mengembalikan nilai apapun. Pada contoh kali ini, function yang dibentuk juga tidak menerima parameter apapun. Beri nama function "TestFunction".



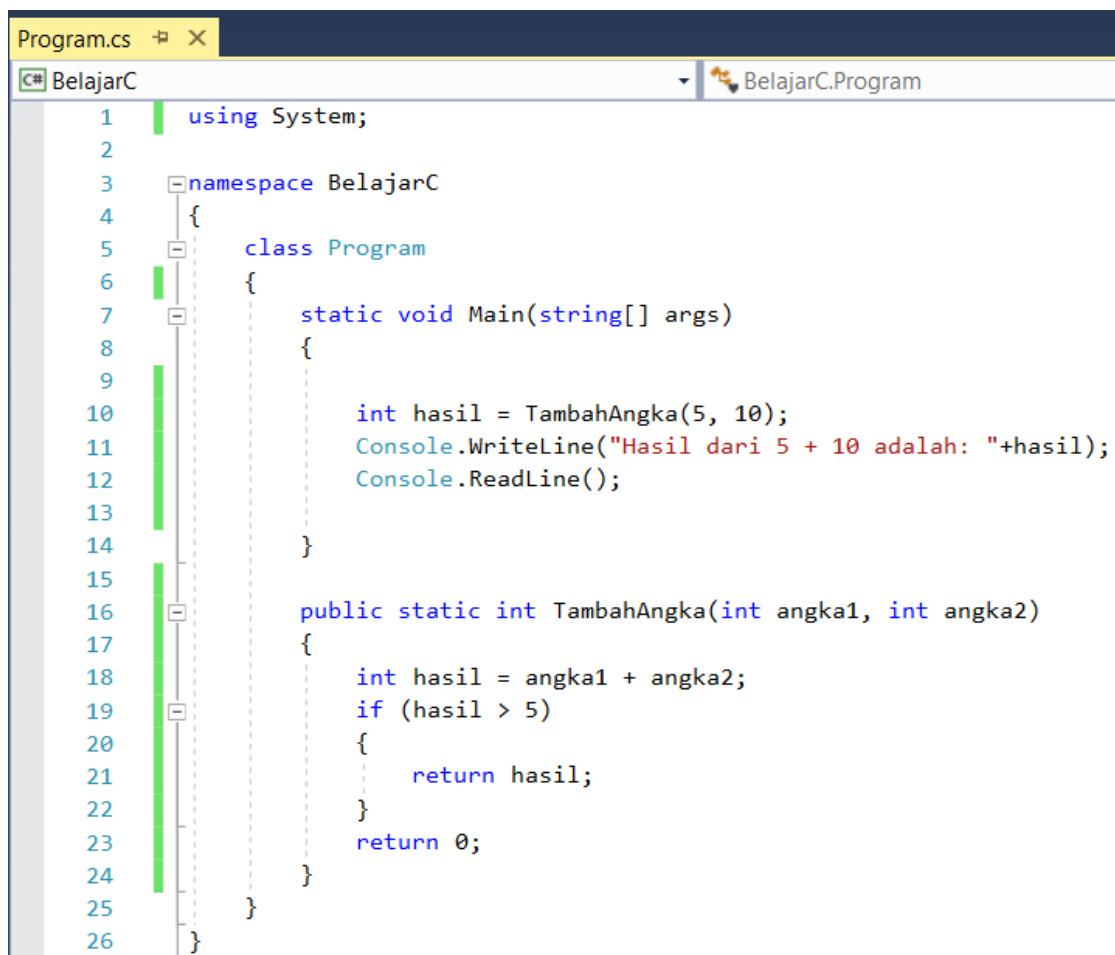
```
1  using System;
2
3  namespace BelajarC
4  {
5      class Program
6      {
7          private static string helloClass = "Hallo, Class";
8
9          static void Main(string[] args)
10         {
11             string helloLocal = "Hallo Local";
12             Console.WriteLine(helloLocal);
13             Console.WriteLine(Program.helloClass);
14             TestFunction();
15         }
16
17         static void TestFunction()
18         {
19             Console.WriteLine("Pemanggilan variabel dari function TestFunction: " + Program.helloClass);
20             Console.ReadLine();
21         }
22     }
23 }
24
```

Dapat diperhatikan bahwa untuk pemanggilan function cukup sederhana yaitu dengan memanggil nama fungsi di dalam Main() method. Jika sesuai, maka aplikasi akan berjalan seperti ini.



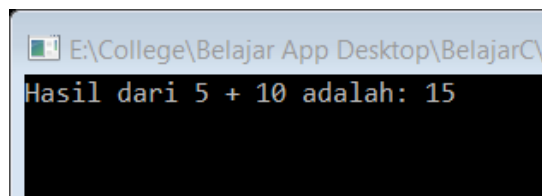
```
E:\College\Belajar App Desktop\BelajarC\BelajarC\bin\Debug\BelajarC.exe
Hallo Local
Hallo, Class
Pemanggilan variabel dari function TestFunction: Hallo, Class
```

Pada contoh function kedua ini, kita akan mencoba membuat sebuah function yang bernama "TambahAngka" yang dapat menerima parameter serta mengembalikan nilai yang telah diolah dalam fungsi tersebut.



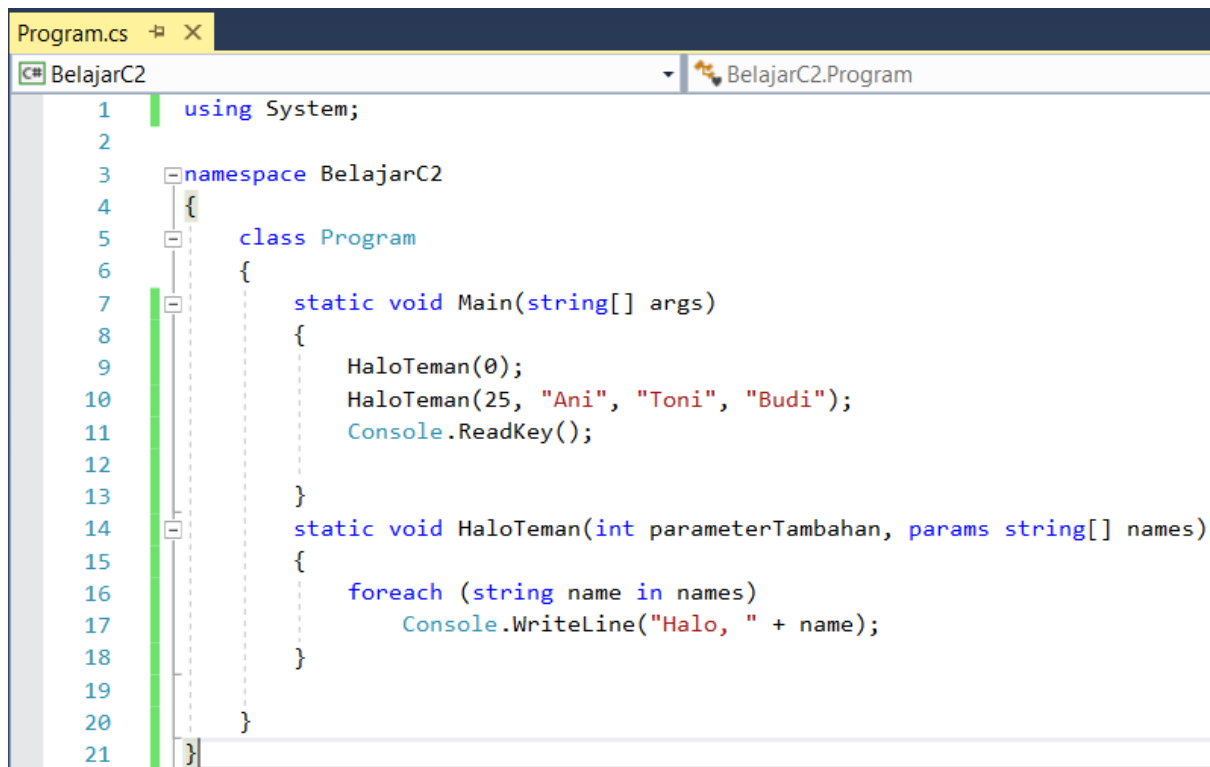
```
1  using System;
2
3  namespace BelajarC
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9
10             int hasil = TambahAngka(5, 10);
11             Console.WriteLine("Hasil dari 5 + 10 adalah: "+hasil);
12             Console.ReadLine();
13
14         }
15
16         public static int TambahAngka(int angka1, int angka2)
17         {
18             int hasil = angka1 + angka2;
19             if (hasil > 5)
20             {
21                 return hasil;
22             }
23             return 0;
24         }
25     }
26 }
```

Bentuk pemanggilan function pada contoh kali ini berbeda dari sebelumnya. Kita harus memberikan inputan berupa dua buah angka sesuai dengan parameter yang telah di set pada function tersebut.



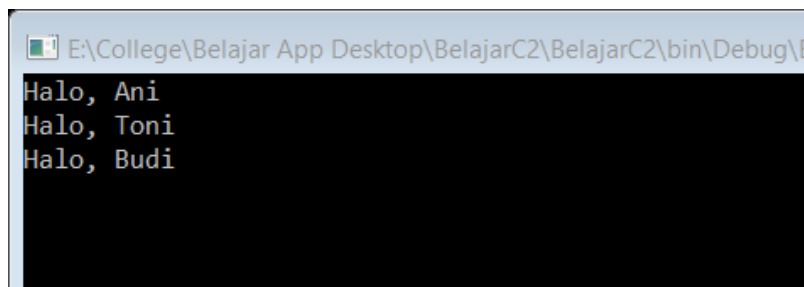
```
E:\College\Belajar App Desktop\BelajarC\
Hasil dari 5 + 10 adalah: 15
```

Dalam kasus tertentu, tak sedikit kode program yang mengharuskan sebuah fungsi untuk menerima parameter dalam jumlah yang banyak. Dalam hal itu, kita bisa menambahkan parameter yang mampu menerima inputan yang cukup banyak dengan menggunakan array atau list parameter. Silahkan buat sebuah Projek baru, beri nama "BelajarC2". Ketikkan baris kode dibawah ini.



```
1  using System;
2
3  namespace BelajarC2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              HaloTeman(0);
10             HaloTeman(25, "Ani", "Toni", "Budi");
11             Console.ReadKey();
12         }
13
14         static void HaloTeman(int parameterTambahan, params string[] names)
15         {
16             foreach (string name in names)
17                 Console.WriteLine("Halo, " + name);
18         }
19     }
20 }
21
```

Dapat diperhatikan dari baris kode diatas bahwa penggunaan keyword „params” memungkinkan program untuk menerima sejumlah inputan parameter dalam sekali tulis. Selain itu, function yang dibentuk dengan menggunakan params juga dapat menerima parameter dengan tipe data yang berbeda.



```
E:\College\Belajar App Desktop\BelajarC2\BelajarC2\bin\Debug\
Halo, Ani
Halo, Toni
Halo, Budi
```

Kemudian silahkan run kode program tersebut, dan akan menampilkan hasil seperti gambar di atas.

Latihan dan Tugas Analisa (Buat di laporan) :

1. Apa fungsi .WriteLine?
2. Apa fungsi .ReadLine()?
3. Apa yang terjadi ketika .ReadLine() dihapus dari program di atas? Jelaskan alasannya.

4. Pada kode program ke 4, mengapa pendeklarasian variabel helloClass dilakukan diluar method Main() ?
5. Pada kode program ke 4, mengapa kita tidak bisa memanggil .helloLocal ?
6. Pada kode program ke 5, apa yang terjadi ketika "return 0" dihapus dari baris kode? Pesan error apa yang muncul dan mengapa?
7. Apa baris kode yang harus ditambahkan pada kode program ke 6 jika ingin menampilkan nilai 0 dan 25 yang telah di set sebagai parameter?
8. Mengapa ketika params dibuat di awal seperti berikut, program menjadi error?

```
static void HaloTeman(params string[] names, int parameterTambahan)
```

9. Apakah mungkin untuk membuat 2 params dalam sebuah function?