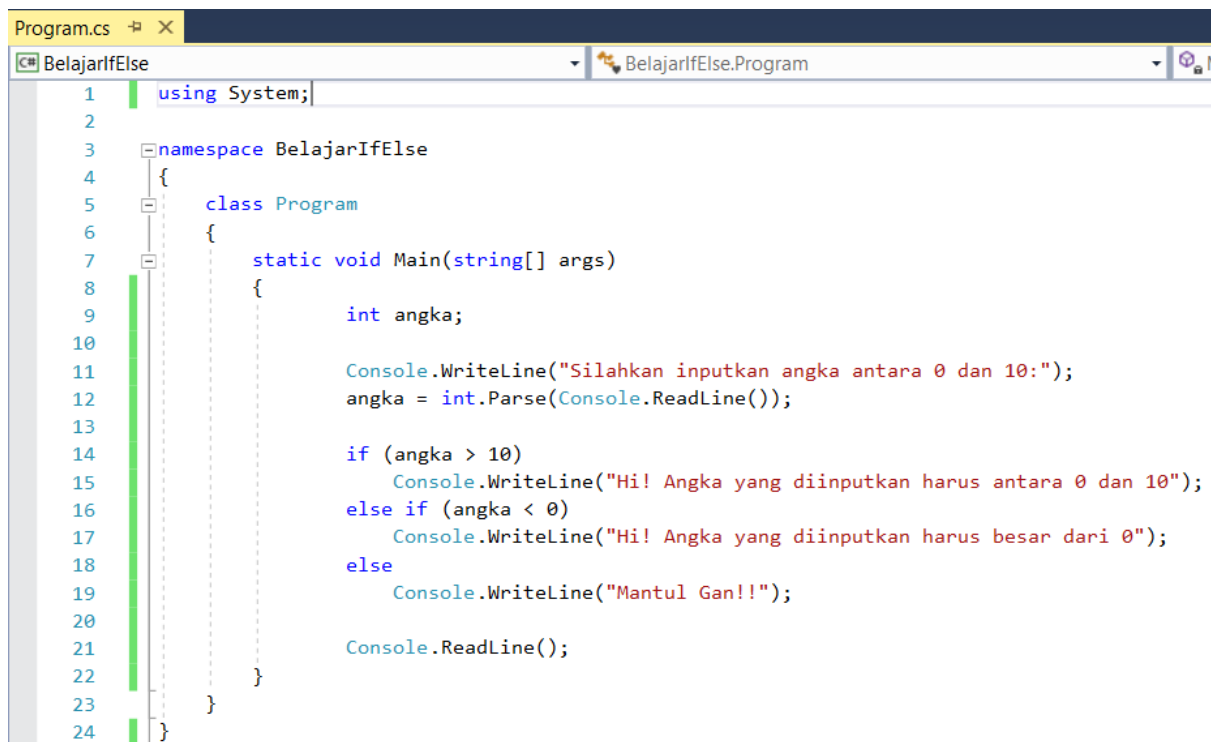


Pertemuan 3

If else statement

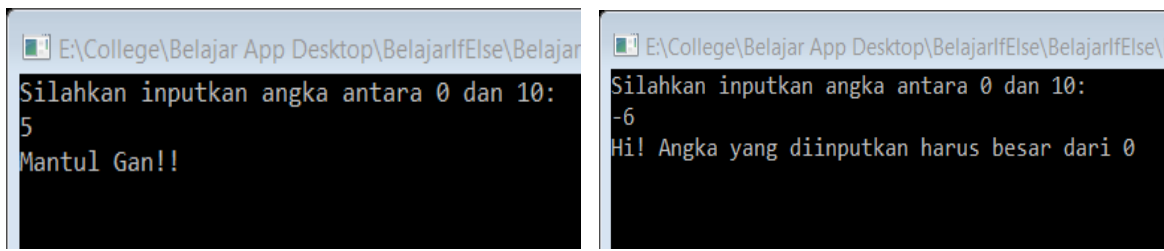
Salah satu bentuk pernyataan yang paling umum digunakan pada pemrograman adalah if-else statement. Memiliki pemahaman untuk menuliskannya ke dalam Bahasa pemrograman C# sangat penting untuk mendukung pengembangan aplikasi. Secara umum, cara penulisan pernyataan if-else pada C# serupa dengan Bahasa pemrograman Java.

Silahkan buat proyek baru, kemudian beri nama „BelajarIfElse“. Silahkan tambahkan kode program seperti pada contoh di bawah ini.



```
1  using System;
2
3  namespace BelajarIfElse
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int angka;
10
11              Console.WriteLine("Silahkan inputkan angka antara 0 dan 10:");
12              angka = int.Parse(Console.ReadLine());
13
14              if (angka > 10)
15                  Console.WriteLine("Hi! Angka yang diinputkan harus antara 0 dan 10");
16              else if (angka < 0)
17                  Console.WriteLine("Hi! Angka yang diinputkan harus besar dari 0");
18              else
19                  Console.WriteLine("Mantul Gan!!");
20
21              Console.ReadLine();
22          }
23      }
24  }
```

Pada contoh ini, kita akan membuat sebuah program yang menerima inputan angka antara 0 hingga 10. Dimana ketika pengguna memberikan inputan selain angka tersebut, maka akan muncul output yang telah didefinisikan pada kode program.



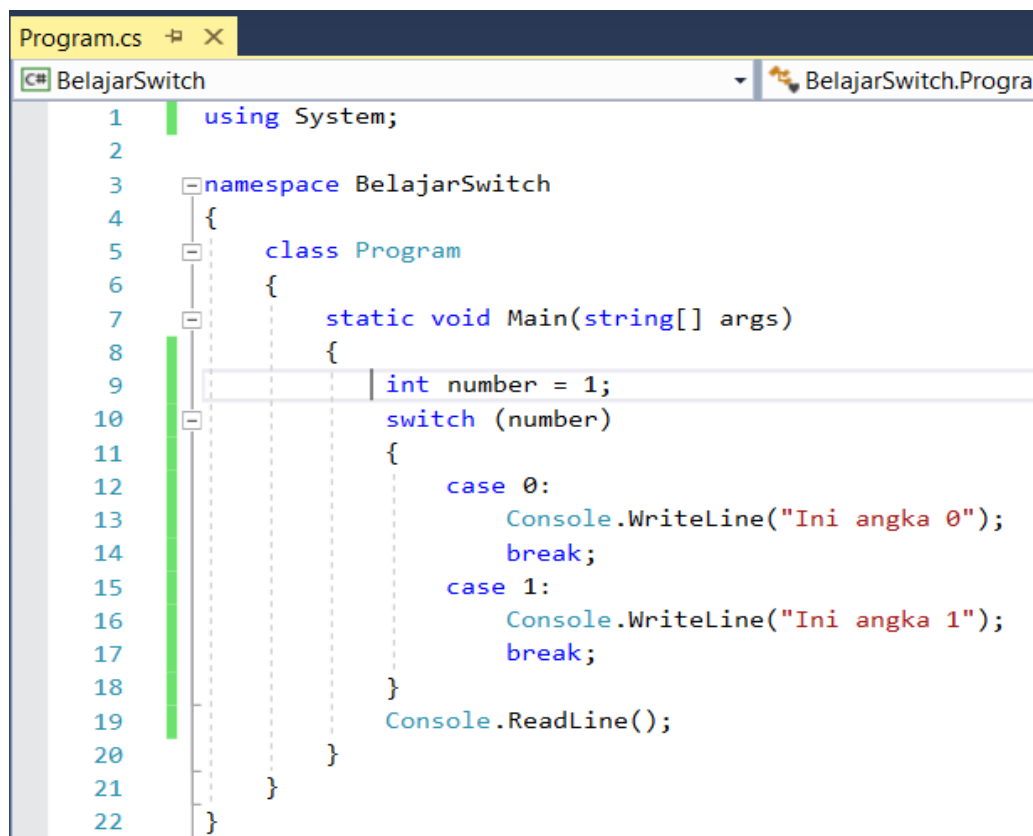
```
Silahkan inputkan angka antara 0 dan 10:
5
Mantul Gan!!

Silahkan inputkan angka antara 0 dan 10:
-6
Hi! Angka yang diinputkan harus besar dari 0
```

Sesuai dengan hasil pengambilan keputusan yang telah ditetapkan pada kode program, maka ketika aplikasi dijalankan akan muncul output seperti pada gambar di atas.

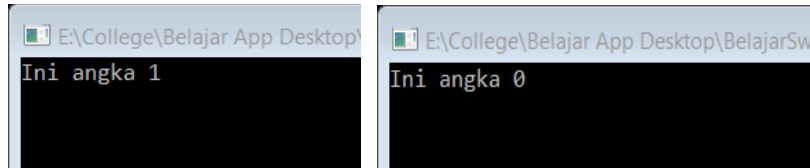
Switch statement

Seperti halnya dalam Bahasa pemrograman lainnya, C# juga memungkinkan kita untuk menggunakan model pernyataan Switch untuk menyederhanakan penulisan kode dengan kondisi yang cukup banyak. Untuk mencobanya, silahkan bentuk proyek baru dan beri nama "BelajarSwitch".



```
Program.cs
C# BelajarSwitch
1 using System;
2
3 namespace BelajarSwitch
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int number = 1;
10            switch (number)
11            {
12                case 0:
13                    Console.WriteLine("Ini angka 0");
14                    break;
15                case 1:
16                    Console.WriteLine("Ini angka 1");
17                    break;
18            }
19            Console.ReadLine();
20        }
21    }
22 }
```

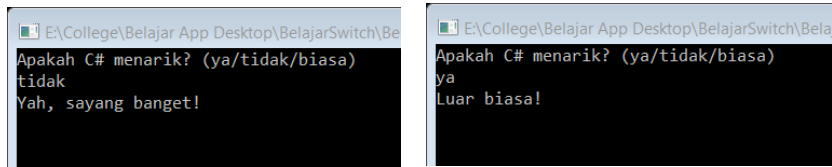
Pada program di atas, kita akan melakukan pengecekan apakah inputan sesuai dengan salah satu kondisi yang ditetapkan pada pernyataan switch tersebut. Silahkan tekan F5 untuk melihat hasil dari aplikasi.



Adapun contoh program switch yang kedua adalah dengan menggunakan tipe data string sebagai inputan. Silahkan bentuk proyek baru atau tetap menggunakan proyek yang sama dengan di atas.

```
Program.cs* [icon] X
C# BelajarSwitch BelajarSwitch.Program
1 using System;
2
3 namespace BelajarSwitch
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("Apakah C# menarik? (ya/tidak/biasa)");
10            string input = Console.ReadLine();
11            switch (input.ToLower())
12            {
13                case "ya":
14                case "biasa":
15                    Console.WriteLine("Luar biasa!");
16                    break;
17                case "tidak":
18                    Console.WriteLine("Yah, sayang banget!");
19                    break;
20            }
21            Console.ReadLine();
22        }
23    }
24 }
```

Pada kode program di atas, penggunaan huruf besar dan kecil tidak akan menjadi persoalan. Sebab kita telah memastikan bahwa inputan string akan diubah menjadi huruf kecil dengan menggunakan fungsi **.ToLower()**. Kemudian, silahkan run kode program di atas untuk mendapatkan hasil seperti pada contoh di bawah ini.

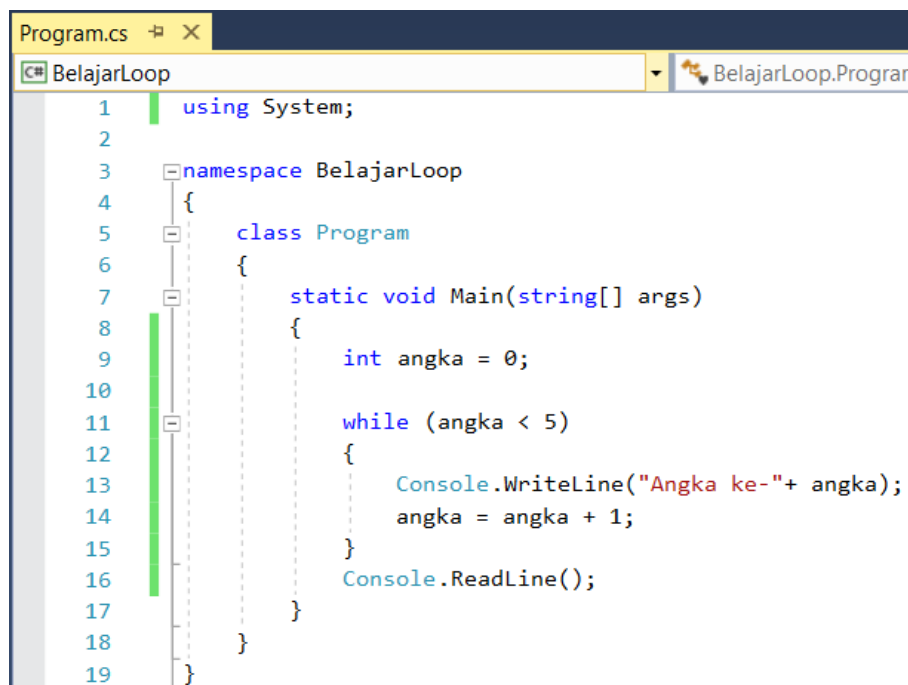


Loops

Kemampuan untuk menjalankan sekumpulan kode berulang kali juga sangat dibutuhkan. Pada C#, terdapat empat macam cara untuk menulis perulangan. Pada modul ini, kita akan mencoba model tersebut satu persatu.

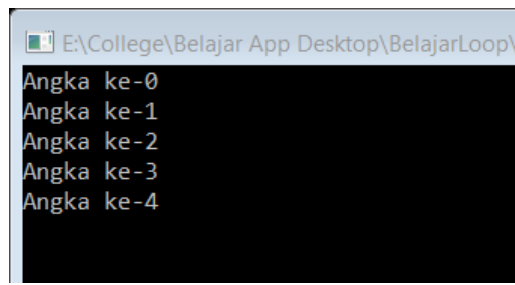
1. While Loop

While loop akan melakukan eksekusi kode program selama kondisi yang diberikan bernilai true. Silahkan bentuk sebuah proyek baru, beri nama „BelajarLoop“.



```
Program.cs  X
C# BelajarLoop BelajarLoop.Program
1 using System;
2
3 namespace BelajarLoop
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int angka = 0;
10
11             while (angka < 5)
12             {
13                 Console.WriteLine("Angka ke-" + angka);
14                 angka = angka + 1;
15             }
16             Console.ReadLine();
17         }
18     }
19 }
```

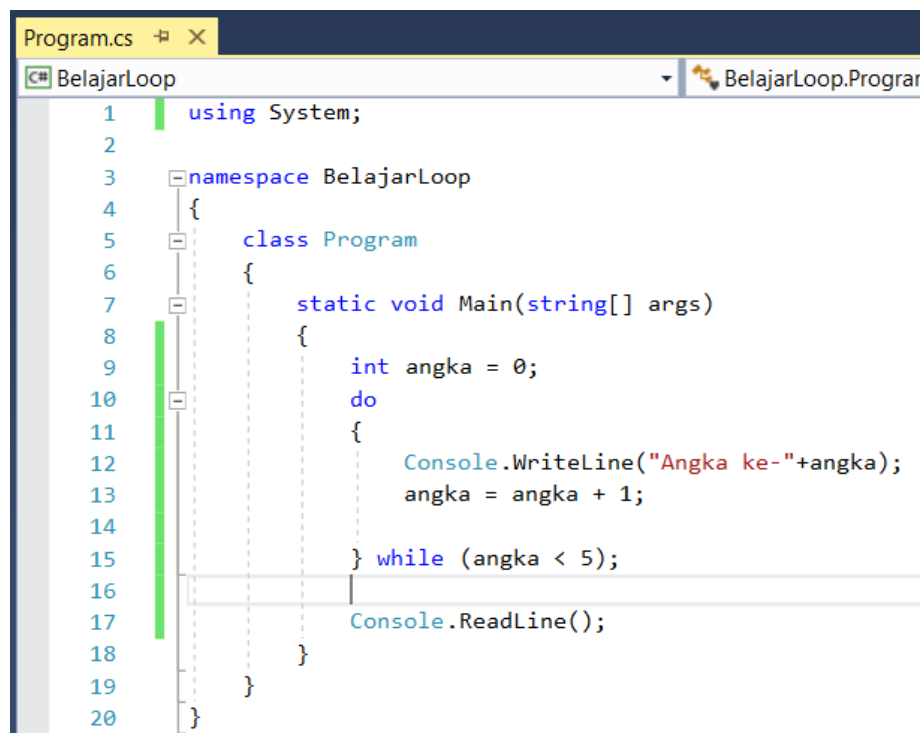
Pada contoh program di atas, kita akan melakukan looping sebanyak 5 kali dengan nilai awal yang dimulai dari 0. Maka aplikasi akan menghasilkan tampilan seperti gambar di bawah ketika di jalankan.



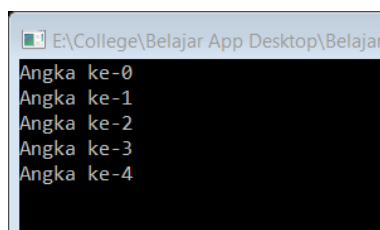
```
E:\College\Belajar App Desktop\BelajarLoop\  
Angka ke-0  
Angka ke-1  
Angka ke-2  
Angka ke-3  
Angka ke-4
```

2. Do While

Model looping kedua adalah dengan menggunakan Do-While. Dimana pada model kali ini, blok kode akan dieksekusi minimal sekali. Kemudian akan dilanjutkan dengan pengecekan kondisi sesuai dengan yang ditetapkan.



```
Program.cs  X  
C# BelajarLoop BelajarLoop.Program  
1 using System;  
2  
3 namespace BelajarLoop  
4 {  
5     class Program  
6     {  
7         static void Main(string[] args)  
8         {  
9             int angka = 0;  
10            do  
11            {  
12                Console.WriteLine("Angka ke-"+angka);  
13                angka = angka + 1;  
14            } while (angka < 5);  
15  
16            Console.ReadLine();  
17  
18        }  
19    }  
20 }
```

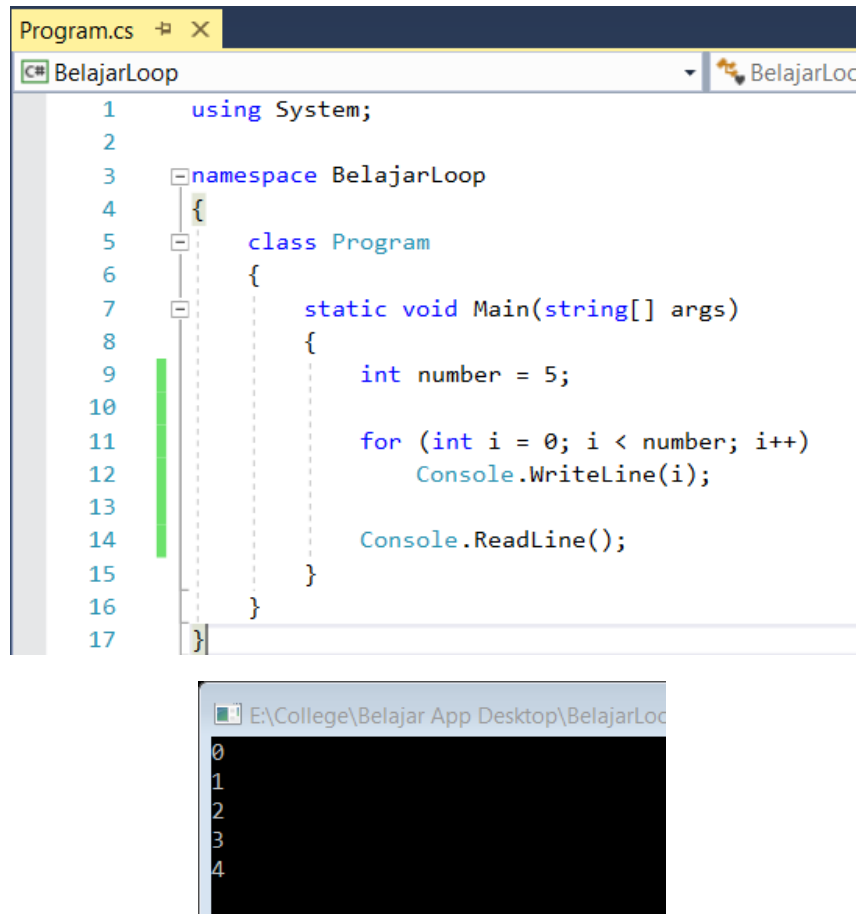


```
E:\College\Belajar App Desktop\BelajarL  
Angka ke-0  
Angka ke-1  
Angka ke-2  
Angka ke-3  
Angka ke-4
```

Seperti pada hasil dari model looping pertama, hasil program kali ini juga sama. Perbedaannya hanya terdapat pada alur eksekusi kode program.

3. For Loop

Pada model looping For Loop juga serupa dengan for loop pada Bahasa pemrograman Java. Inisialisasi, kondisi dan proses counting terjadi dalam satu baris. Silahkan ketikkan kode program berikut untuk melihat hasil program tersebut.



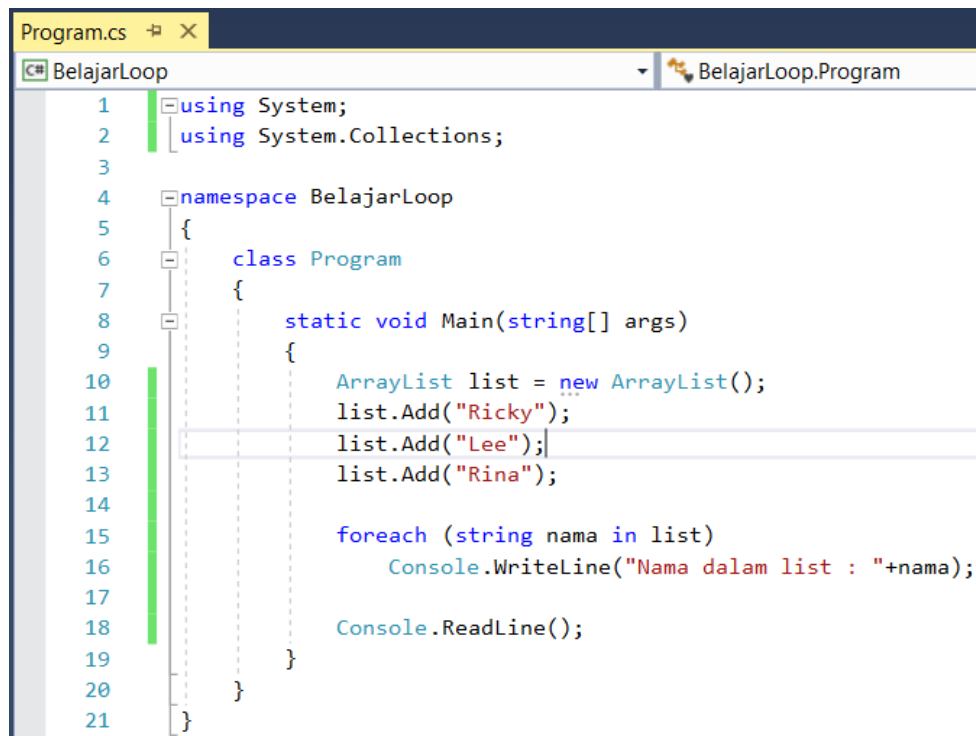
The image shows a C# program in a code editor and its output in a console window. The code defines a namespace 'BelajarLoop' and a class 'Program' with a static 'Main' method. Inside 'Main', a variable 'number' is set to 5, and a 'for' loop iterates from 'i = 0' to 'i < number', printing each value of 'i' to the console. The console output shows the numbers 0, 1, 2, 3, and 4 on separate lines.

```
1 using System;
2
3 namespace BelajarLoop
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int number = 5;
10
11             for (int i = 0; i < number; i++)
12                 Console.WriteLine(i);
13
14             Console.ReadLine();
15         }
16     }
17 }
```

E:\College\Belajar App Desktop\BelajarLoc
0
1
2
3
4

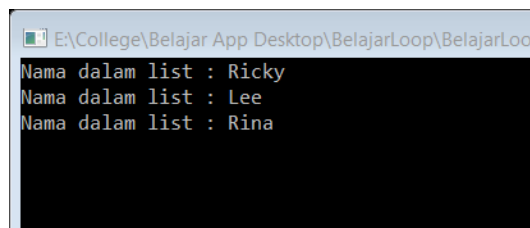
4. Foreach

Model looping yang terakhir adalah Foreach. Pada looping jenis ini, perulangan akan dilakukan terhadap sekumpulan items seperti array maupun list. Pada contoh kali ini, kita akan menggunakan ArrayList.



```
1 using System;
2 using System.Collections;
3
4 namespace BelajarLoop
5 {
6     class Program
7     {
8         static void Main(string[] args)
9         {
10             ArrayList list = new ArrayList();
11             list.Add("Ricky");
12             list.Add("Lee");
13             list.Add("Rina");
14
15             foreach (string nama in list)
16                 Console.WriteLine("Nama dalam list : "+nama);
17
18             Console.ReadLine();
19         }
20     }
21 }
```

Seperti yang dapat terlihat dari kode di atas, penulisan perulangan hampir serupa dengan for loop. Akan tetapi tidak terlihat proses increment di dalamnya. Dalam hal ini, perintah "foreach" berfungsi sebagai pengganti hal tersebut. Variabel "nama" menjadi penampung dari setiap output perulangan yang dilakukan. Terlihat tipe data variabel "nama" adalah string. Jika dijalankan, maka output program akan terlihat seperti berikut.



```
E:\College\Belajar App Desktop\BelajarLoop\BelajarLoo
Nama dalam list : Ricky
Nama dalam list : Lee
Nama dalam list : Rina
```

Pada umumnya, ketika kita harus bekerja dengan sekumpulan data seperti list atau array, model Foreach adalah model looping yang paling sering digunakan karena terasa lebih sederhana.

Array

Seperti yang telah diketahui, array bekerja sebagai tempat penyimpanan dari sekumpulan item. Adapun cara pendeklarasian Array adalah sebagai berikut:

<type data> [] <nama Variabel>;

Ketika ingin melakukan inisialisasi Panjang Array, maka kita dapat melakukan penulisan kode seperti berikut ini :

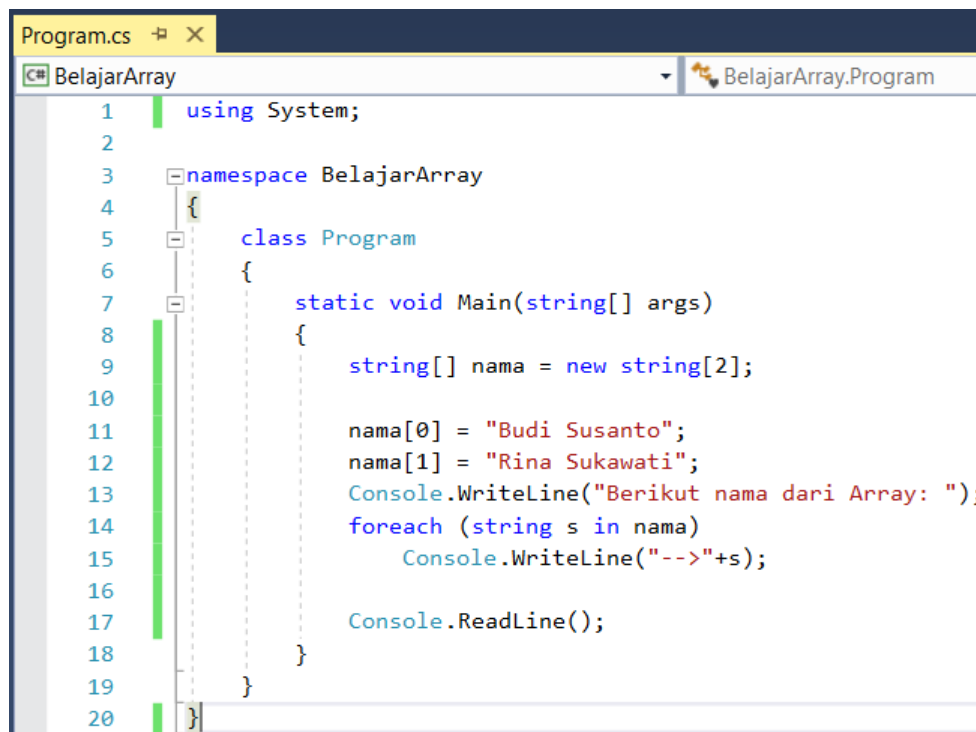
<type data> [] <nama Variabel> = new <type data> [Panjang Array];

Dan untuk melakukan pengisian value array, dapat dilakukan dengan cara:

<nama Variabel> [indeks ke-] = "nilai"; atau

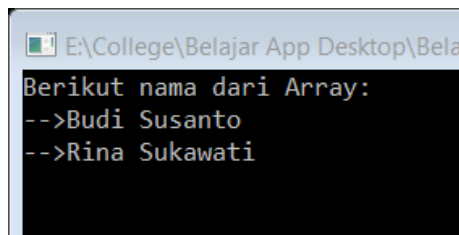
<type data> [] <nama Variabel> = { int1, int2, int3, ...};

Untuk memulai program, silahkan bentuk sebuah proyek baru, beri nama „BelajarArray“. Kemudian silahkan coba kode program di bawah ini.



```
Program.cs
BelajarArray
1  using System;
2
3  namespace BelajarArray
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              string[] nama = new string[2];
10
11              nama[0] = "Budi Susanto";
12              nama[1] = "Rina Sukawati";
13              Console.WriteLine("Berikut nama dari Array: ");
14              foreach (string s in nama)
15                  Console.WriteLine("-->" + s);
16
17              Console.ReadLine();
18          }
19      }
20 }
```

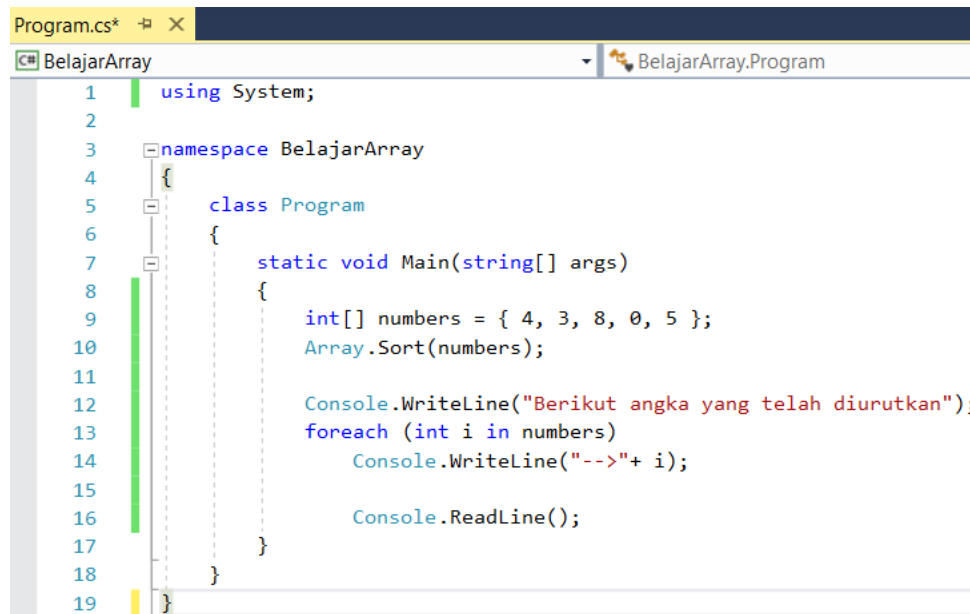
Pada contoh array di atas, kita melakukan pendeklarasian isi array dengan memasukkan nilai sesuai dengan indeks yang diinginkan. Adapun panjang array yang didefinisikan adalah 2, sehingga indeks array yang terisi adalah indeks 0 dan 1.



```
E:\College\Belajar App Desktop\BelajarArray\Program.cs
Berikut nama dari Array:
-->Budi Susanto
-->Rina Sukawati
```

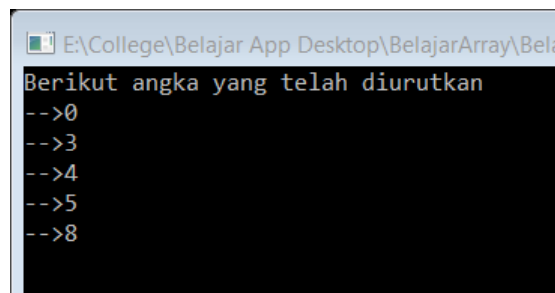
Ketika program dijalankan, maka akan muncul tampilan seperti pada gambar di atas.

Jika pada contoh program di atas menggunakan tipe data String, maka pada contoh berikutnya kita akan menggunakan tipe data Integer. Dalam program kali ini, kita akan memberikan input nilai dengan urutan acak. Output yang diinginkan adalah angka yang telah diurutkan.



```
Program.cs* [X]
BelajarArray BelajarArray.Program
1 using System;
2
3 namespace BelajarArray
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int[] numbers = { 4, 3, 8, 0, 5 };
10            Array.Sort(numbers);
11
12            Console.WriteLine("Berikut angka yang telah diurutkan");
13            foreach (int i in numbers)
14                Console.WriteLine("-->" + i);
15
16            Console.ReadLine();
17        }
18    }
19 }
```

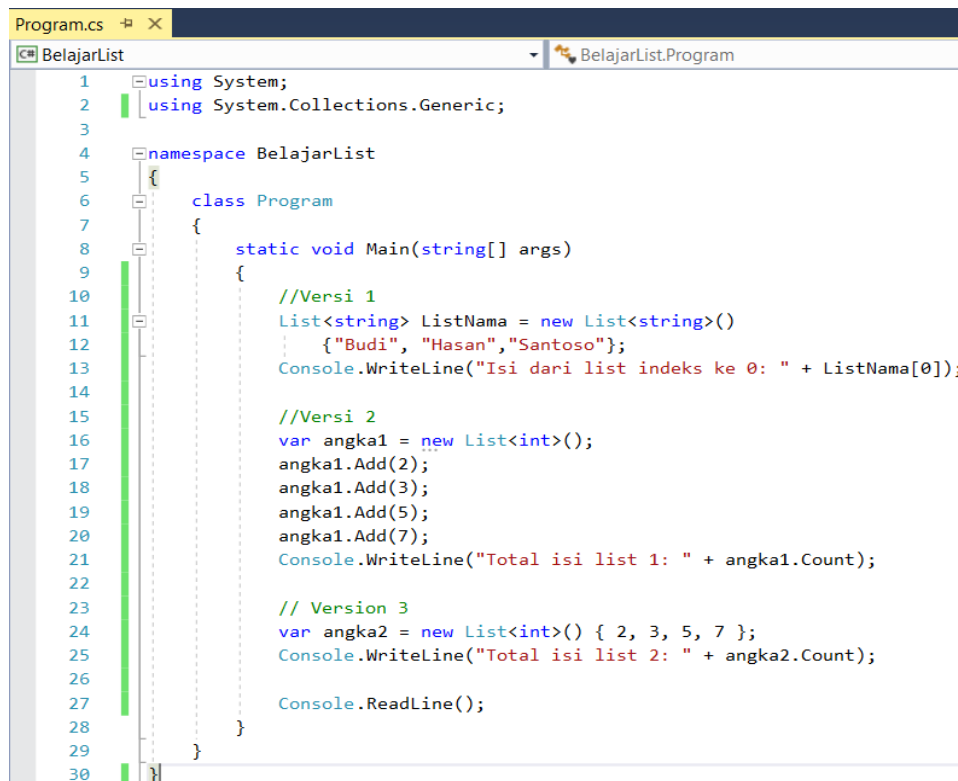
Dapat diperhatikan bahwa untuk melakukan proses pengurutan angka, kita dapat menggunakan fungsi **.sort()** seperti pada contoh di atas. Kemudian jalankan program dan akan menampilkan hasil seperti berikut.



```
E:\College\Belajar App Desktop\BelajarArray\Program.cs
Berikut angka yang telah diurutkan
-->0
-->3
-->4
-->5
-->8
```

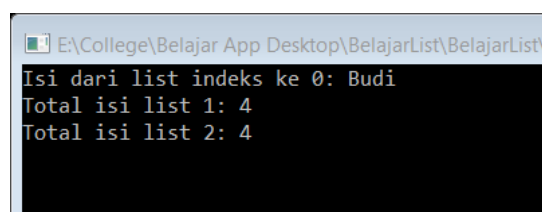
List

List memiliki fungsi yang hampir serupa dengan Array bahkan cara pendeklarasian pun tak jauh berbeda. Akan tetapi, penggunaan List akan terasa lebih penting ketika kita harus melakukan penambahan atau penghapusan data. Sebab akan lebih mudah melakukannya ketika menggunakan List daripada Array. Untuk mencoba program di bawah ini, silahkan bentuk sebuah proyek baru bernama „BelajarList”.



```
1 using System;
2 using System.Collections.Generic;
3
4 namespace BelajarList
5 {
6     class Program
7     {
8         static void Main(string[] args)
9         {
10             //Versi 1
11             List<string> ListNama = new List<string>()
12             { "Budi", "Hasan", "Santoso" };
13             Console.WriteLine("Isi dari list indeks ke 0: " + ListNama[0]);
14
15             //Versi 2
16             var angka1 = new List<int>();
17             angka1.Add(2);
18             angka1.Add(3);
19             angka1.Add(5);
20             angka1.Add(7);
21             Console.WriteLine("Total isi list 1: " + angka1.Count);
22
23             // Version 3
24             var angka2 = new List<int>() { 2, 3, 5, 7 };
25             Console.WriteLine("Total isi list 2: " + angka2.Count);
26
27             Console.ReadLine();
28         }
29     }
30 }
```

Pada contoh program di atas, kita menggunakan tiga versi pendeklarasian list dan cara pengisiannya. Jika dijalankan, maka program akan memberikan output seperti gambar di bawah.



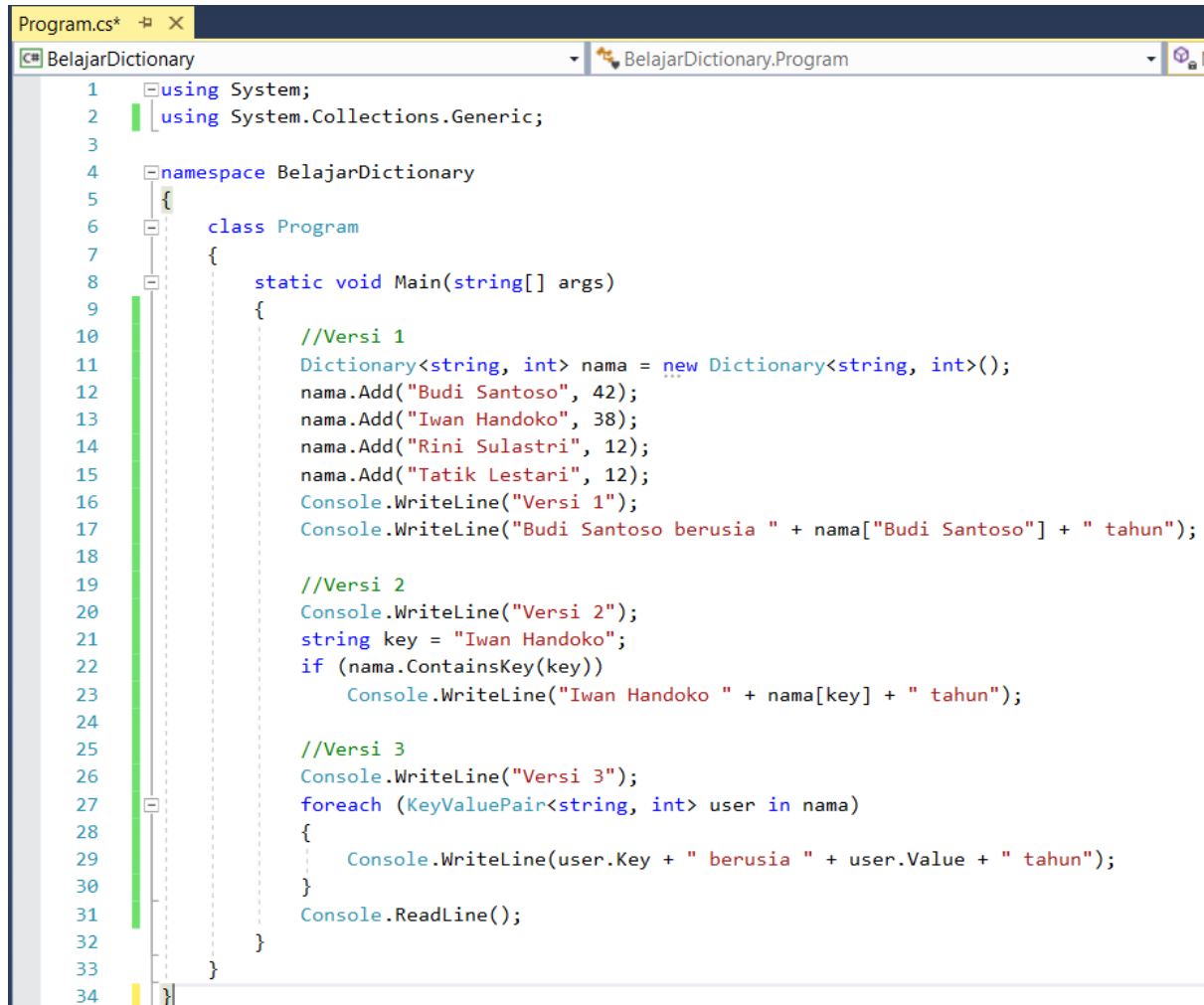
```
E:\College\Belajar App Desktop\BelajarList\BelajarList\
Isi dari list indeks ke 0: Budi
Total isi list 1: 4
Total isi list 2: 4
```

Seperti yang telah disinggung di atas, untuk melakukan penambahan data, kita cukup menggunakan fungsi `namaVar.Add()`. Untuk menghapus, kita dapat menggunakan :

`namaVar.Remove("value")`, `namaVar.RemoveAt(indeks)`, `namaVar.RemoveAll()`

Dictionaries

Berbeda dengan array, Dictionary tidak memiliki urutan yang spesifik. Dengan menggunakan Dictionary, kita bisa merujuk ke konten tanpa menggunakan index. Setiap nilai terkait dengan kata kunci tertentu. Proses pemanggilan bisa dengan memanggil kata kunci atau pengenalan yang telah tersimpan. Ini sama halnya dengan proses pencarian dengan definisi tertentu seperti pada buku kamus.



```
Program.cs* X
BelajarDictionary BelajarDictionary.Program
1 using System;
2 using System.Collections.Generic;
3
4 namespace BelajarDictionary
5 {
6     class Program
7     {
8         static void Main(string[] args)
9         {
10             //Versi 1
11             Dictionary<string, int> nama = new Dictionary<string, int>();
12             nama.Add("Budi Santoso", 42);
13             nama.Add("Iwan Handoko", 38);
14             nama.Add("Rini Sulastrri", 12);
15             nama.Add("Tatik Lestari", 12);
16             Console.WriteLine("Versi 1");
17             Console.WriteLine("Budi Santoso berusia " + nama["Budi Santoso"] + " tahun");
18
19             //Versi 2
20             Console.WriteLine("Versi 2");
21             string key = "Iwan Handoko";
22             if (nama.ContainsKey(key))
23                 Console.WriteLine("Iwan Handoko " + nama[key] + " tahun");
24
25             //Versi 3
26             Console.WriteLine("Versi 3");
27             foreach (KeyValuePair<string, int> user in nama)
28             {
29                 Console.WriteLine(user.Key + " berusia " + user.Value + " tahun");
30             }
31             Console.ReadLine();
32         }
33     }
34 }
```

Dapat dilihat pada Versi pertama dan kedua, pemanggilan value dengan menggunakan kata kunci yang spesifik. Jika kita ingin melakukan proses looping pada koleksi list yang dimiliki untuk mencari sebuah data misalnya, kita bisa menggunakan model seperti pada Versi 3 dengan cara penulisan **KeyValuePair<TKey, TValue>**.

```
E:\College\Belajar App Desktop\BelajarDictionary\BelajarDic
Versi 1
Budi Santoso berusia 42 tahun
Versi 2
Iwan Handoko 38 tahun
Versi 3
Budi Santoso berusia 42 tahun
Iwan Handoko berusia 38 tahun
Rini Sulastri berusia 12 tahun
Tatik Lestari berusia 12 tahun
```

Gambar di atas ini merupakan hasil dari kode program di atas. Silahkan bereksperimen dengan pengetahuan dasar yang kalian miliki tentang bahasa pemrograman C#.

Latihan dan Analisa :

1. Silahkan gunakan operator AND dan OR untuk menyederhanakan pernyataan if else pada kode program 1.
2. Silahkan buat sebuah kode program sederhana yang menerima inputan string berupa nama kamu. Jika inputan bukan nama kamu, maka akan muncul output „Maaf, nama anda tidak dikenali“. Jika benar, maka muncul output „Selamat datang“.

<pre>E:\College\Belajar App Desktop\BelajarIfElse\BelajarIfElse Silahkan masukkan Nama: Maya Selamat Datang, Maya</pre>	<pre>E:\College\Belajar App Desktop\BelajarIfElse\BelajarIfElse Silahkan masukkan Nama: Rara Maaf, nama anda tidak dikenali</pre>
---	---

3. Silahkan tambahkan „default“ pada kode program ke 3 yang memiliki output „Maaf, inputan anda tidak dapat dipahami“.
4. Silahkan buat kode program untuk menampilkan nama bulan berdasarkan urutannya dengan menggunakan switch pada C#. Misal 1=Januari, 2=Februari, 3=...
5. Kapan sebaiknya For Loop digunakan pada pembuatan sebuah program?
6. Apa yang terjadi jika „using.System.Collection;“ pada kode program Loops ke 4 dihapus? Jelaskan alasan/fungsinya.
7. Dengan kode program serupa, silahkan gunakan For Loop pada contoh program Array di atas untuk menampilkan nomor indeks array.

8. Coba tampilkan isi pada program "List" dengan menggunakan „Foreach" loop pada tiap versi.
9. Silahkan tambahkan inputan serta hapus isi dari List dengan menggunakan fungsi .Add dan .Remove yang telah disebutkan di atas.