# **BAB VIII**

# **STACK DAN QUEUE**

## **8.1. Tujuan**

1. Untuk memenuhi Tugas Laporan Praktikum Modul 7 Dasar Komputer dan Pemrograman.
2. Mengenal dan mengetahui perbedaan penggunaan beberapa *IDE* atau aplikasi pemrograman dan beberapa jenis bahasa pemrograman.
3. Memahami pengertian dan konsep dari *Stack* dan *Queue*.
4. Mengetahui penggunaan *Stack* dan *Queue* dengan bahasa pemrograman Python, C#, Java, dan PHP.
5. Mengetahui perbedaan *Stack* dan *Queue* dalam bahasa pemrograman Python, C#, Java, dan PHP.
6. Mampu mengimplementasikan *Stack* dan *Queue* dalam bahasa pemrograman Python, C#, Java, dan PHP.
7. Menjalin silaturahmi dengan kakak tingkat dan teman se-angkatan.

## **Dasar Teori**

### ***Stack***

*Stack* atau tumpukan merupakan sebuah koleksi objek yang menggunakan prinsip *LIFO* (*Last In First Out*), yaitu data yang terakhir kali dimasukkan akan pertama kali keluar dari tumpukan tersebut. Tumpukan dapat diimplementasikan sebagai larik. Tumpukan tergolong struktur data linear dan operasi hanya bisa dilakukan di satu ujung struktur yang biasa disebut *top* dari tumpukan.

Ada beberapa metode penting dalam implementasi *stack*, yaitu:

1. *push*(), berfungsi untuk memasukkan data.

|  |
| --- |
| public void push(String value){  stack[++top] = value;  } |

1. *pop*(), berfungsi untuk mengeluarkan data terakhir (atas).

|  |
| --- |
| public String pop() {  return stack[top — ];  } |

1. *peek*(), berfungsi untuk melihat data yang berada pada tumpukan paling atas (akan dikeluarkan).

|  |
| --- |
| public String peek() {  return stack[top];  } |

1. *isEmpty*(), berfungsi untuk menguji apakah *stack* masih kosong.

|  |
| --- |
| public boolean isEmpty() {  return top == -1;  } |

1. *isFull*(), berfungsi untuk menguji apakah *stack* telah penuh.

|  |
| --- |
| public boolean isFull() {  return top == max-1;  } |

Sumber: id.wikipedia.org, medium.com

### ***Queue***

*Queue* adalah bentuk lain dari konsep implementasi *linked list*. *Queue* mempunyai konsep yakni *FIFO* (*First In First Out*). Setiap data yang pertama kali masuk, dialah yang akan keluar duluan. Penambahan itemnya terjadi pada sebuah ujung yang biasa disebut sebagai “ekor” atau (*rear*) dan untuk penghapusannya, terjadi pada ujung yang satunya atau biasa disebut “kepala” atau (*head*). Contoh paling sederhana dalam kehidupan sehari-hari adalah antrian pengunjung bank. Saat masuk, kita akan diberi nomor antrian terlebih dahulu sebelum dipanggil oleh *teller*. Kita akan menunggu sampai urutan kita barulah kita dapat bertransaksi di *teller*.

Beberapa contoh operasi yang ada dalam *Queue* adalah sebagai berikut:

|  |
| --- |
| class Queue:  def \_\_init\_\_(self):  self.items=[]  def isEmpty(self):  return self.items==[]  def enqueue(self,item):  self.items.insert(0,item)  def dequeue(self):  return self.items.pop()  def size(self):  return len(self.items)  def getList(self):  return self.items |

1. *queue*(), membuat suatu antrian baru yang kosong. Tidak memerlukan parameter dan mengembalikan suatu antrian kosong.
2. *enqueue*(item), menambahkan suatu item baru ke ujung saru antrian. Perlu item dan tidak mengembalikan sesuatu.
3. *dequeue*(), menghapus item depan dari antrian. Tidak memerlukan parameter dan mengembalikan itemnya. Antrian termodifikasi.
4. *isEmpty*(), menguji untuk melihat apakah antrian dalam keadaan kosong. Tidak memerlukan parameter dan mengembalian nilai *boolean*.
5. *size*(), mengembalikan jumlah item yang ada di dalam antrian. Tidak memerlukan parameter dan mengembalikan suatu *integer*.

Sumber: mahirkoding.com, mn-belajarpython.blogspot.com

## **Hasil Percobaan dan Analisis**

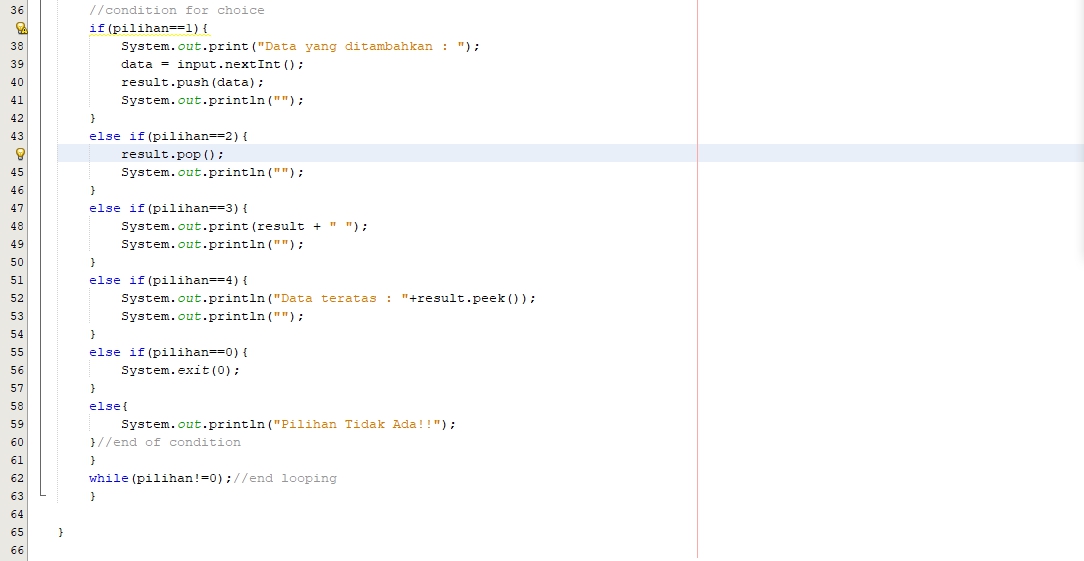
### **Bahasa Pemrograman Java**

***Stack***

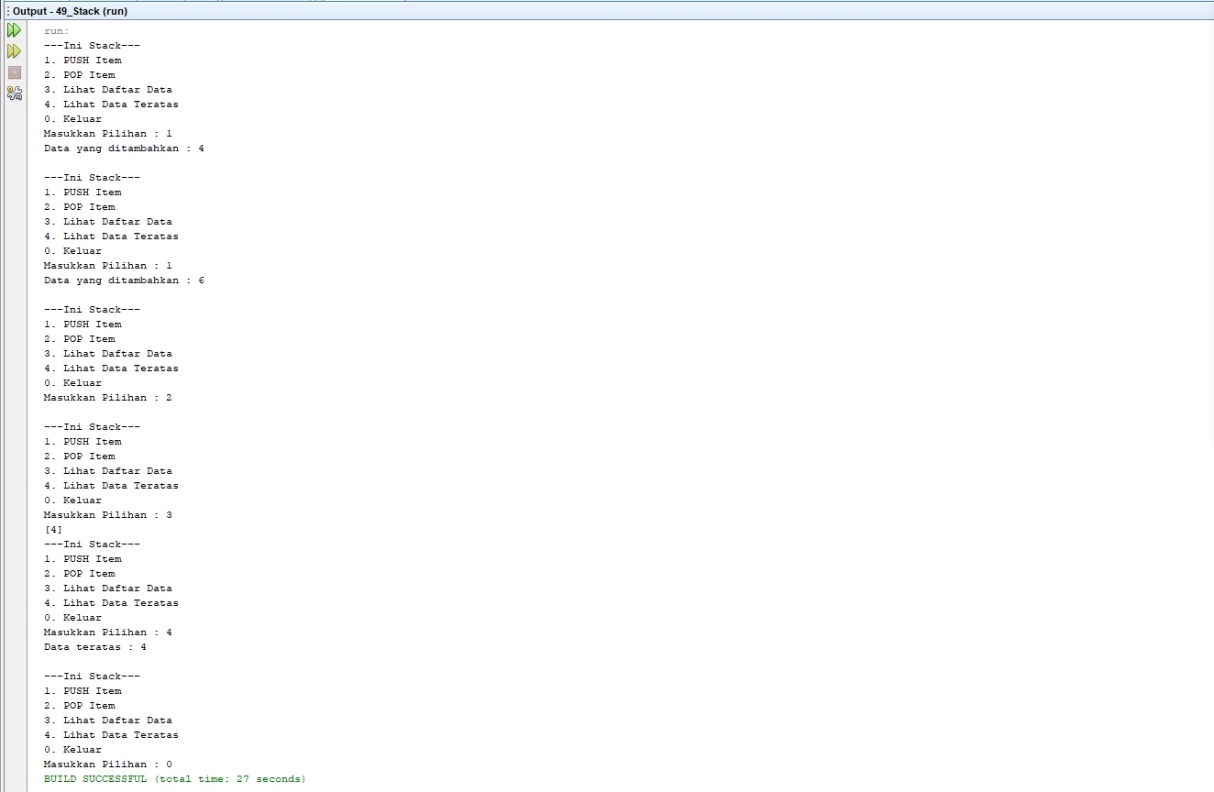
|  |
| --- |
| package pkg49\_stack;  import java.util.Scanner;  import java.util.Stack;  public class Main {  public static void main(String[] args) {  int pilihan;  int data;  Stack result = new Stack();  //looping until false condition  do{  //Displaying Menu  System.out.println("---Ini Stack---");  System.out.println("1. PUSH Item");  System.out.println("2. POP Item");  System.out.println("3. Lihat Daftar Data");  System.out.println("4. Lihat Data Teratas");  System.out.println("0. Keluar");  Scanner input = new Scanner(System.in);  System.out.print("Masukkan Pilihan : ");  pilihan = input.nextInt();  //condition for choice  if(pilihan==1){  System.out.print("Data yang ditambahkan : ");  data = input.nextInt();  result.push(data);  System.out.println("");  }  else if(pilihan==2){  result.pop();  System.out.println("");  }  else if(pilihan==3){  System.out.print(result + " ");  System.out.println("");  }  else if(pilihan==4){  System.out.println("Data teratas : "+result.peek());  System.out.println("");  }  else if(pilihan==0){  System.exit(0);  }  else{  System.out.println("Pilihan Tidak Ada!!");  }//end of condition  }  while(pilihan!=0);//end looping  }    } |



Gambar 8. 1. Source Code Stack dalam Java (1)



Gambar 8. 2. Source Code Stack dalam Java (2)

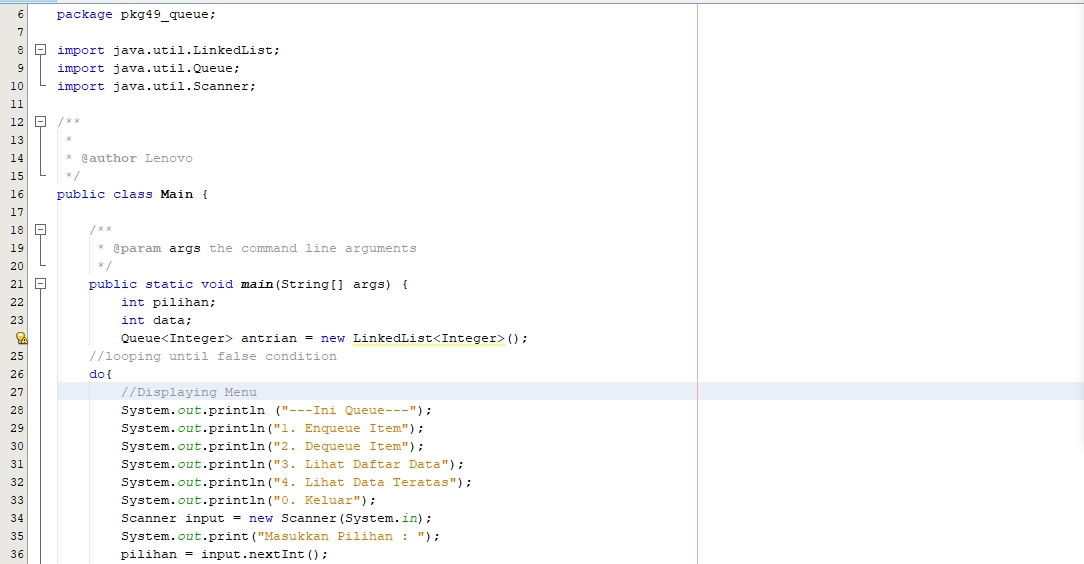


Gambar 8. 3. Output Stack dalam Java

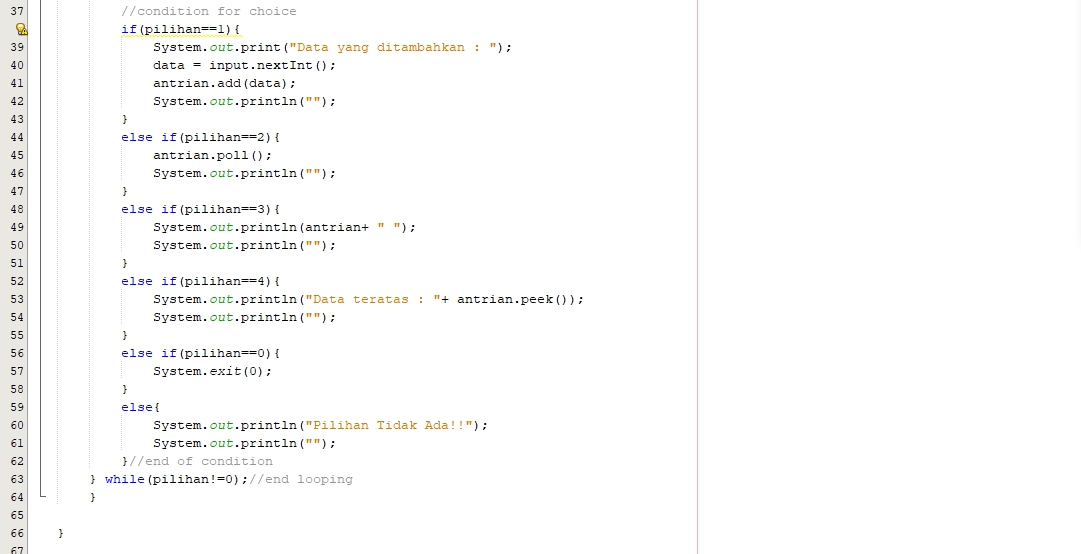
Pertama kita masukkan *package* supaya kita dapat melakukan *stack* serta kita inisialisasikan *stack* kita, selanjutnya kita definisikan variabel-variabel yang akan digunakan dan kita keluarkan fitur-fitur yang tersedia dan berikan *input* supaya pengguna dapat memilih fitur yang diinginkan. Fitur-fitur yang tersedia ini kita pisahkan menggunakan fungsi *if, elif*  dan *else*. Untuk fitur pertama kita yaitu *Push item* kita akan memberikan *input* kepada pengguna untuk menambahkan angka dan selanjutnya angka yang telah dimasukkan akan ditambahkan ke *stack*. Fitur kedua yaitu *POP item*, jika pengguna memilih fitur ini maka kita gunakan *syntax pop* untuk mengeluarkan *item* terakhir. Lalu fitur yang ketiga yaitu mengeluarkan *item item* yang berada pada *stack* ke layar. dan fitur terakhir adalah lihat data teratas, disini kita langsung mengeluarkan data yang teratas.

***Queue***

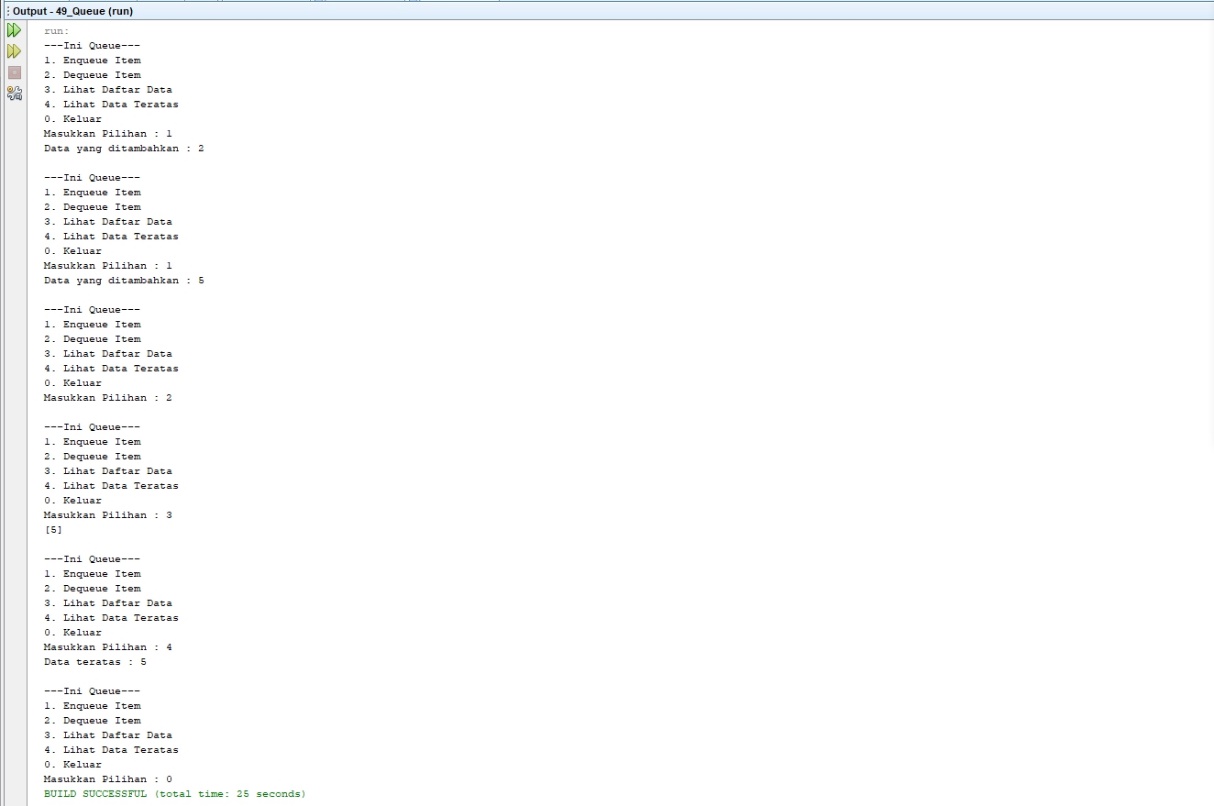
|  |
| --- |
| package pkg49\_queue;  import java.util.LinkedList;  import java.util.Queue;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  int pilihan;  int data;  Queue<Integer> antrian = new LinkedList<Integer>();  //looping until false condition  do{  //Displaying Menu  System.out.println ("---Ini Queue---");  System.out.println("1. Enqueue Item");  System.out.println("2. Dequeue Item");  System.out.println("3. Lihat Daftar Data");  System.out.println("4. Lihat Data Teratas");  System.out.println("0. Keluar");  Scanner input = new Scanner(System.in);  System.out.print("Masukkan Pilihan : ");  pilihan = input.nextInt();  //condition for choice  if(pilihan==1){  System.out.print("Data yang ditambahkan : ");  data = input.nextInt();  antrian.add(data);  System.out.println("");  }  else if(pilihan==2){  antrian.poll();  System.out.println("");  }  else if(pilihan==3){  System.out.println(antrian+ " ");  System.out.println("");  }  else if(pilihan==4){  System.out.println("Data teratas : "+ antrian.peek());  System.out.println("");  }  else if(pilihan==0){  System.exit(0);  }  else{  System.out.println("Pilihan Tidak Ada!!");  System.out.println("");  }//end of condition  } while(pilihan!=0);//end looping  }    } |



Gambar 8. 4. Source Code Queue dalam Java (1)



Gambar 8. 5. Source Code Queue dalam Java (2)



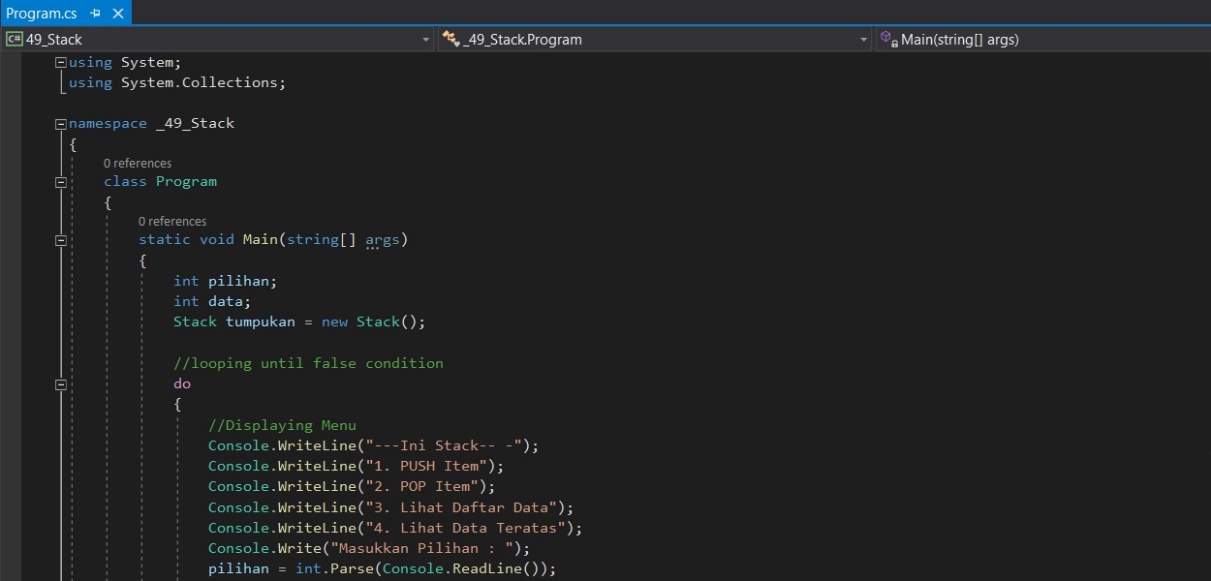
Gambar 8. 6. Output Queue dalam Java

Pertama kita masukkan *package* supaya kita dapat melakukan *queue* serta kita inisialisasikan *queue* kita, selanjutnya kita definisikan variabel-variabel yang akan digunakan dan kita keluarkan fitur-fitur yang tersedia dan berikan *input* supaya pengguna dapat memilih fitur yang diinginkan. Fitur-fitur yang tersedia ini kita pisahkan menggunakan fungsi *if,elif*  dan *else*. Untuk fitur pertama kita yaitu *enqueue Data* kita akan memberikan *input* kepada pengguna untuk menambahkan angka dan selanjutnya angka yang telah dimasukkan akan ditambahkan ke *queue*. Fitur kedua yaitu *dequeue item*, jika pengguna memilih fitur ini maka kita gunakan *syntax poll* untuk mengeluarkan *item* pertama. Lalu fitur yang ketiga yaitu mengeluarkan *item-item* yang berada pada *stack* ke layar. dan fitur terakhir adalah lihat data teratas, disini kita langsung mengeluarkan data yang teratas.

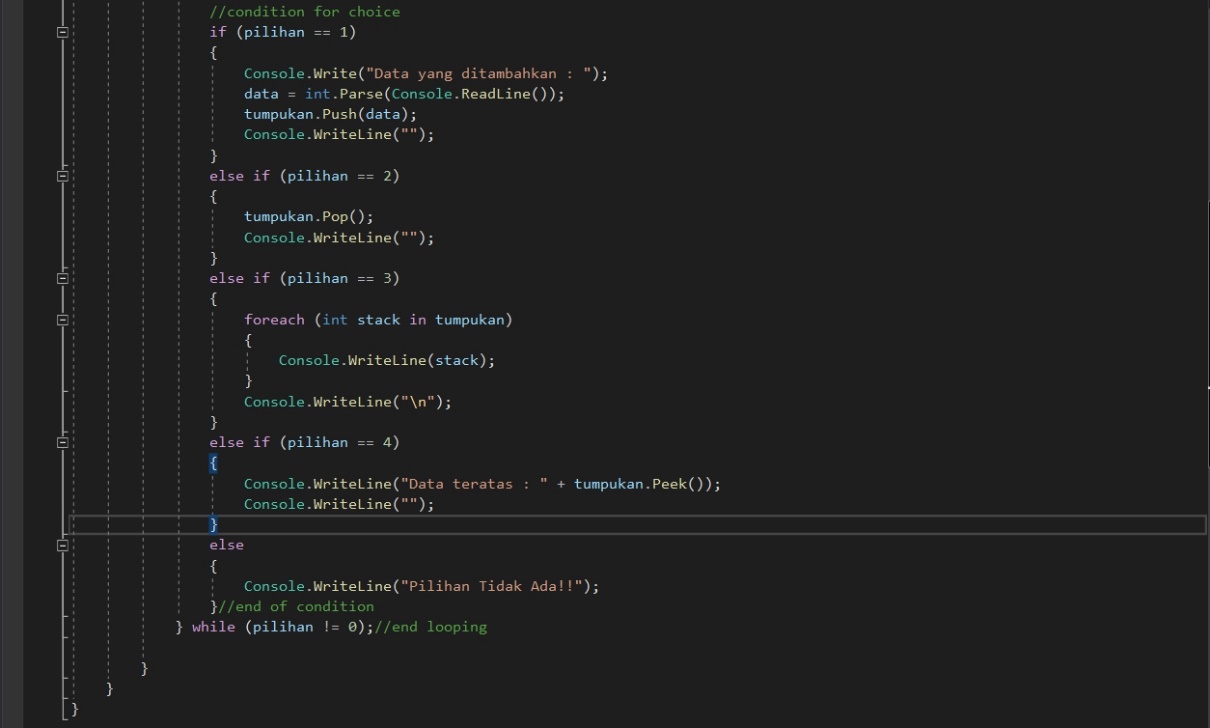
### **Bahasa Pemrograman C#**

***Stack***

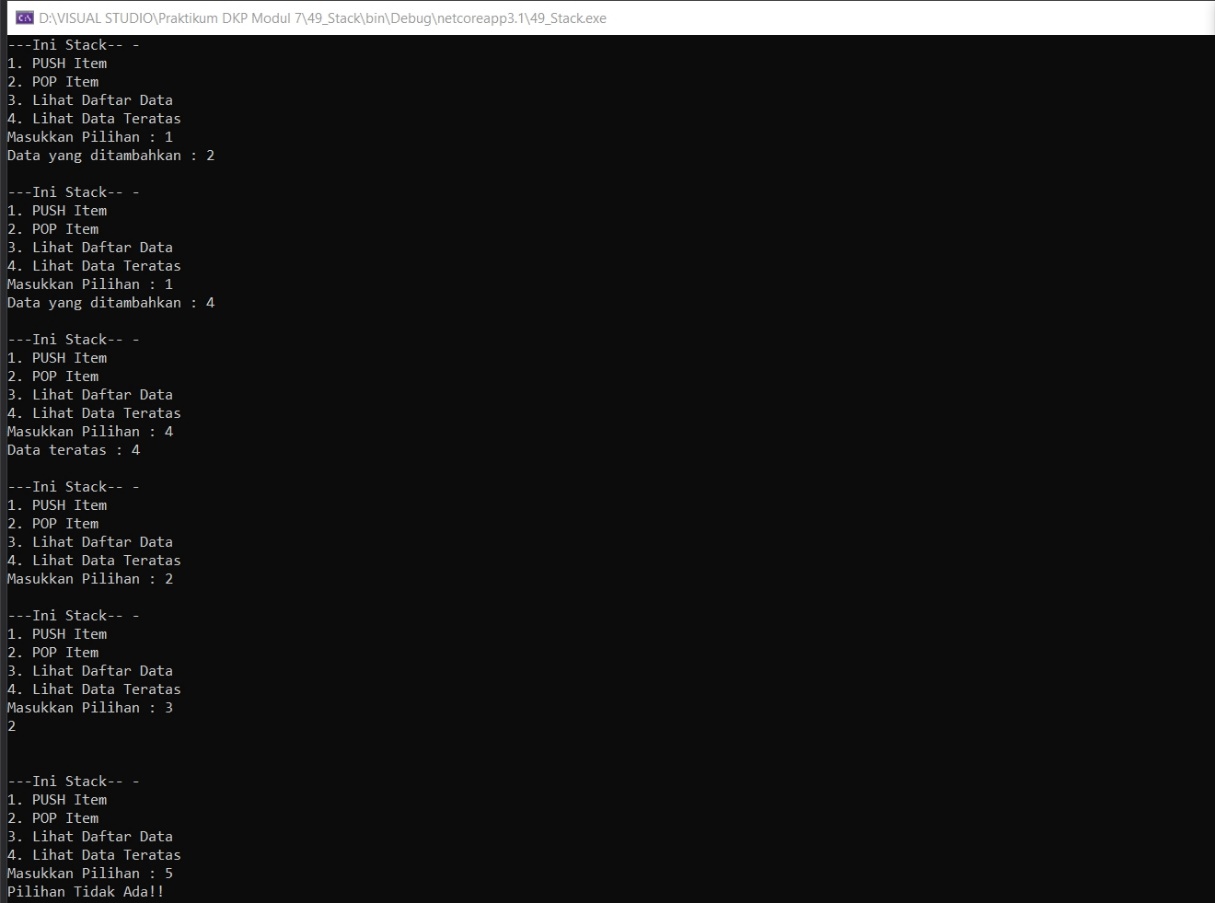
|  |
| --- |
| using System;  using System.Collections;  namespace \_49\_Stack  {  class Program  {  static void Main(string[] args)  {  int pilihan;  int data;  Stack tumpukan = new Stack();  //looping until false condition  do  {  //Displaying Menu  Console.WriteLine("---Ini Stack-- -");  Console.WriteLine("1. PUSH Item");  Console.WriteLine("2. POP Item");  Console.WriteLine("3. Lihat Daftar Data");  Console.WriteLine("4. Lihat Data Teratas");  Console.Write("Masukkan Pilihan : ");  pilihan = int.Parse(Console.ReadLine());  //condition for choice  if (pilihan == 1)  {  Console.Write("Data yang ditambahkan : ");  data = int.Parse(Console.ReadLine());  tumpukan.Push(data);  Console.WriteLine("");  }  else if (pilihan == 2)  {  tumpukan.Pop();  Console.WriteLine("");  }  else if (pilihan == 3)  {  foreach (int stack in tumpukan)  {  Console.WriteLine(stack);  }  Console.WriteLine("\n");  }  else if (pilihan == 4)  {  Console.WriteLine("Data teratas : " + tumpukan.Peek());  Console.WriteLine("");  }  else  {  Console.WriteLine("Pilihan Tidak Ada!!");  }//end of condition  } while (pilihan != 0);//end looping  }  }  } |



Gambar 8. 7. Source Code Stack dalam C# (1)



Gambar 8. 8. Source Code Stack dalam C# (2)

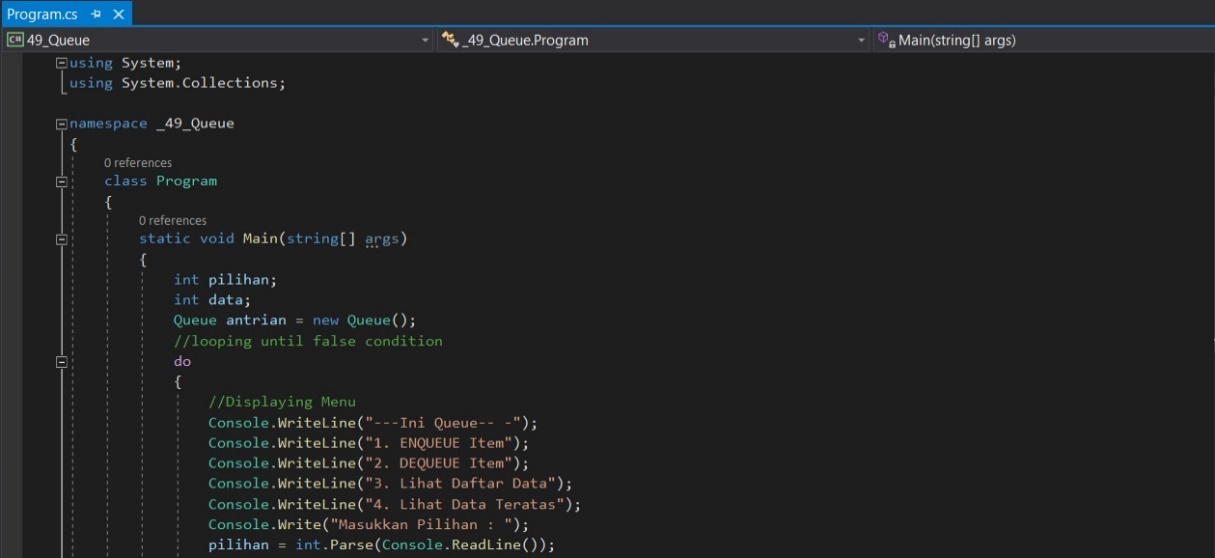


Gambar 8. 9. Output Stack dalam C#

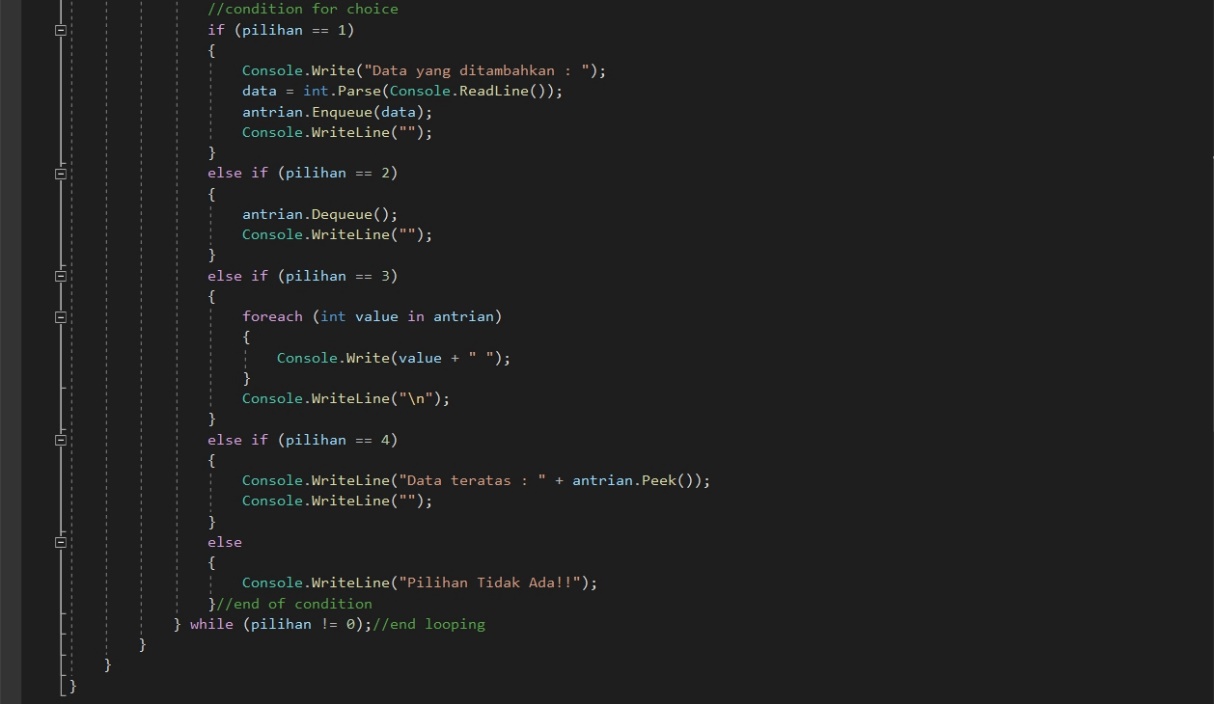
Pertama kita masukkan *package* supaya kita dapat melakukan *stack* serta kita inisialisasikan *stack* kita, selanjutnya kita definisikan variabel-variabel yang akan digunakan dan kita keluarkan fitur-fitur yang tersedia dan berikan *input* supaya pengguna dapat memilih fitur yang diinginkan. Fitur-fitur yang tersedia ini kita pisahkan menggunakan fungsi *if, else if*  dan *else*. Untuk fitur pertama kita yaitu *Push item* kita akan memberikan *input* kepada pengguna untuk menambahkan angka dan selanjutnya angka yang telah dimasukkan akan ditambahkan ke *stack*. Fitur kedua yaitu *POP item*, jika pengguna memilih fitur ini maka kita gunakan *syntax pop* untuk mengeluarkan *item* terakhir. Lalu fitur yang ketiga yaitu mengeluarkan *item item* yang berada pada *stack* ke layar. dan fitur terakhir adalah lihat data teratas, disini kita langsung mengeluarkan data yang teratas.

***Queue***

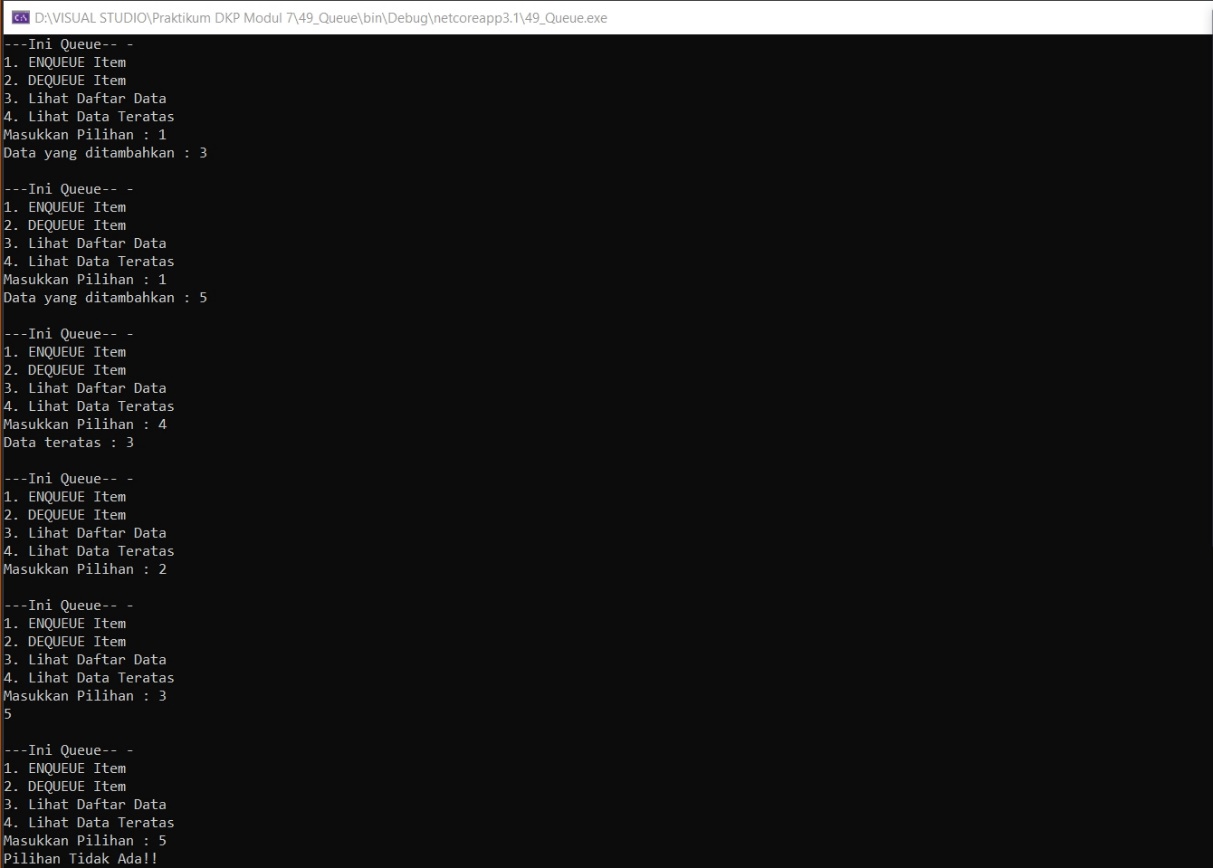
|  |
| --- |
| using System;  using System.Collections;  namespace \_49\_Queue  {  class Program  {  static void Main(string[] args)  {  int pilihan;  int data;  Queue antrian = new Queue();  //looping until false condition  do  {  //Displaying Menu  Console.WriteLine("---Ini Queue-- -");  Console.WriteLine("1. ENQUEUE Item");  Console.WriteLine("2. DEQUEUE Item");  Console.WriteLine("3. Lihat Daftar Data");  Console.WriteLine("4. Lihat Data Teratas");  Console.Write("Masukkan Pilihan : ");  pilihan = int.Parse(Console.ReadLine());  //condition for choice  if (pilihan == 1)  {  Console.Write("Data yang ditambahkan : ");  data = int.Parse(Console.ReadLine());  antrian.Enqueue(data);  Console.WriteLine("");  }  else if (pilihan == 2)  {  antrian.Dequeue();  Console.WriteLine("");  }  else if (pilihan == 3)  {  foreach (int value in antrian)  {  Console.Write(value + " ");  }  Console.WriteLine("\n");  }  else if (pilihan == 4)  {  Console.WriteLine("Data teratas : " + antrian.Peek());  Console.WriteLine("");  }  else  {  Console.WriteLine("Pilihan Tidak Ada!!");  }//end of condition  } while (pilihan != 0);//end looping  }  }  } |



Gambar 8. 10. Source Code Queue dalam C# (1)



Gambar 8. 11. Source Code Queue dalam C# (2)



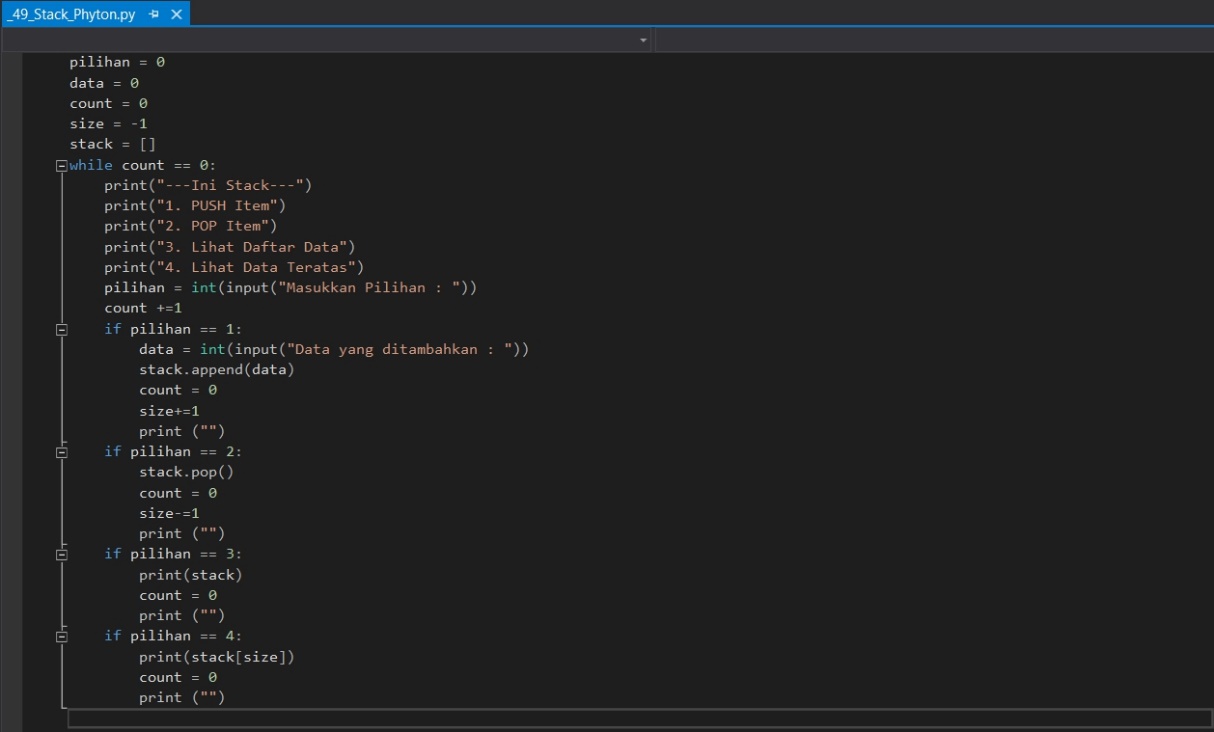
Gambar 8. 12. Output Queue dalam C#

Pertama kita masukkan *package* supaya kita dapat melakukan *queue* serta kita inisialisasikan *queue* kita, selanjutnya kita definisikan variabel-variabel yang akan digunakan dan kita keluarkan fitur-fitur yang tersedia dan berikan *input* supaya pengguna dapat memilih fitur yang diinginkan. Fitur-fitur yang tersedia ini kita pisahkan menggunakan fungsi *if, else if*  dan *else*. Untuk fitur pertama kita yaitu *enqueue Data* kita akan memberikan *input* kepada pengguna untuk menambahkan angka dan selanjutnya angka yang telah dimasukkan akan ditambahkan ke *queue*. Fitur kedua yaitu *dequeue item*, jika pengguna memilih fitur ini maka kita gunakan *syntax dequeue* untuk mengeluarkan *item* pertama. Lalu fitur yang ketiga yaitu mengeluarkan *item item* yang berada pada *stack* ke layar. dan fitur terakhir adalah lihat data teratas, disini kita langsung mengeluarkan data yang teratas.

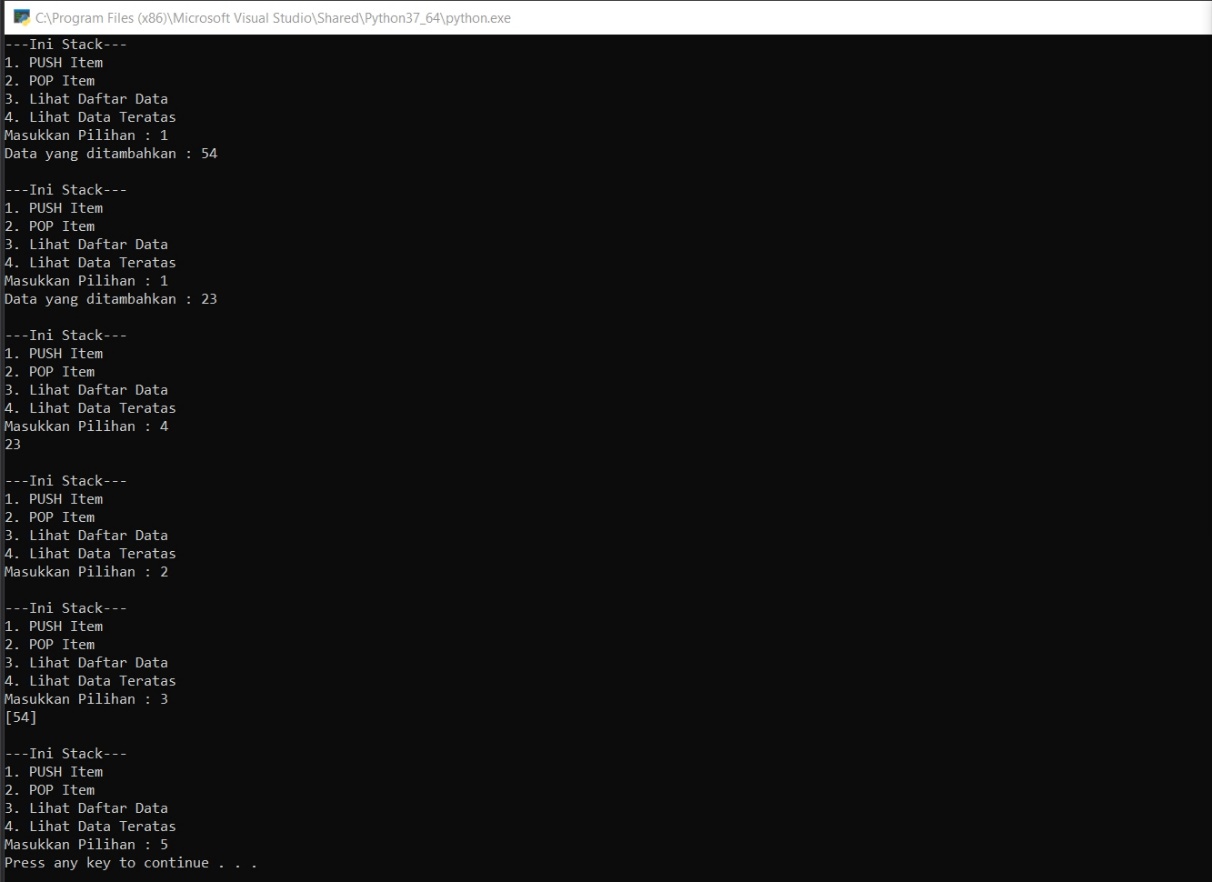
### **Bahasa Pemrograman Python**

***Stack***

|  |
| --- |
| pilihan = 0  data = 0  count = 0  size = -1  stack = []  while count == 0:  print("---Ini Stack---")  print("1. PUSH Item")  print("2. POP Item")  print("3. Lihat Daftar Data")  print("4. Lihat Data Teratas")  pilihan = int(input("Masukkan Pilihan : "))  count +=1  if pilihan == 1:  data = int(input("Data yang ditambahkan : "))  stack.append(data)  count = 0  size+=1  print ("")  if pilihan == 2:  stack.pop()  count = 0  size-=1  print ("")  if pilihan == 3:  print(stack)  count = 0  print ("")  if pilihan == 4:  print(stack[size])  count = 0  print ("") |



Gambar 8. 13. Source Code Stack dalam Python

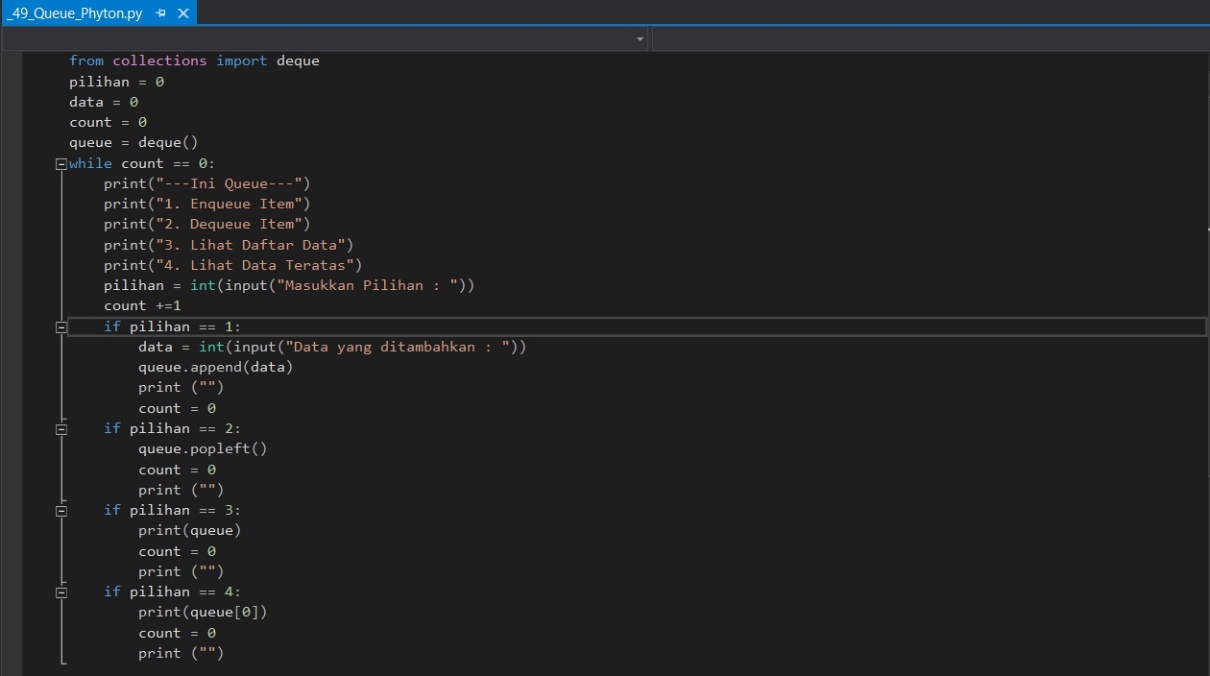


Gambar 8. 14. Output Stack dalam Python

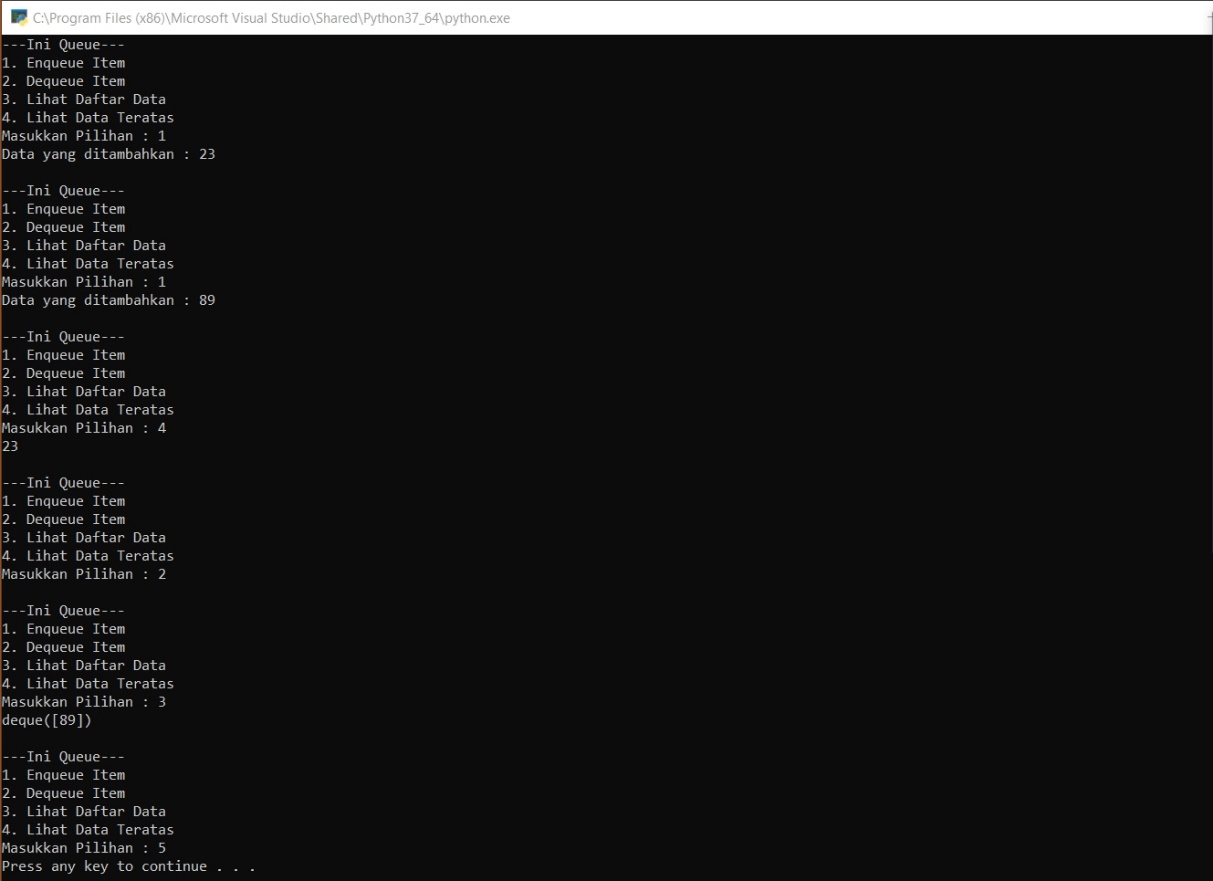
Pertama kita inisialisasikan *stack* kita, selanjutnya kita definisikan variabel-variabel yang akan digunakan dan kita keluarkan fitur-fitur yang tersedia dan berikan *input* supaya pengguna dapat memilih fitur yang diinginkan. Fitur-fitur yang tersedia ini kita pisahkan menggunakan fungsi *if, elif*  dan *else*. Untuk fitur pertama kita yaitu *Push item*  kita akan memberikan *input* kepada pengguna untuk menambahkan angka dan selanjutnya angka yang telah dimasukkan akan ditambahkan ke *stack*. Fitur kedua yaitu *POP item*, jika pengguna memilih fitur ini maka kita gunakan *syntax pop* untuk mengeluarkan *item* terakhir. Lalu fitur yang ketiga yaitu mengeluarkan *item-item* yang berada pada *stack* ke layar. dan fitur terakhir adalah lihat data teratas, disini kita langsung mengeluarkan data yang teratas.

***Queue***

|  |
| --- |
| from collections import deque  pilihan = 0  data = 0  count = 0  queue = deque()  while count == 0:  print("---Ini Queue---")  print("1. Enqueue Item")  print("2. Dequeue Item")  print("3. Lihat Daftar Data")  print("4. Lihat Data Teratas")  pilihan = int(input("Masukkan Pilihan : "))  count +=1  if pilihan == 1:  data = int(input("Data yang ditambahkan : "))  queue.append(data)  print ("")  count = 0  if pilihan == 2:  queue.popleft()  count = 0  print ("")  if pilihan == 3:  print(queue)  count = 0  print ("")  if pilihan == 4:  print(queue[0])  count = 0  print ("") |



Gambar 8. 15. Source Code Queue dalam Python

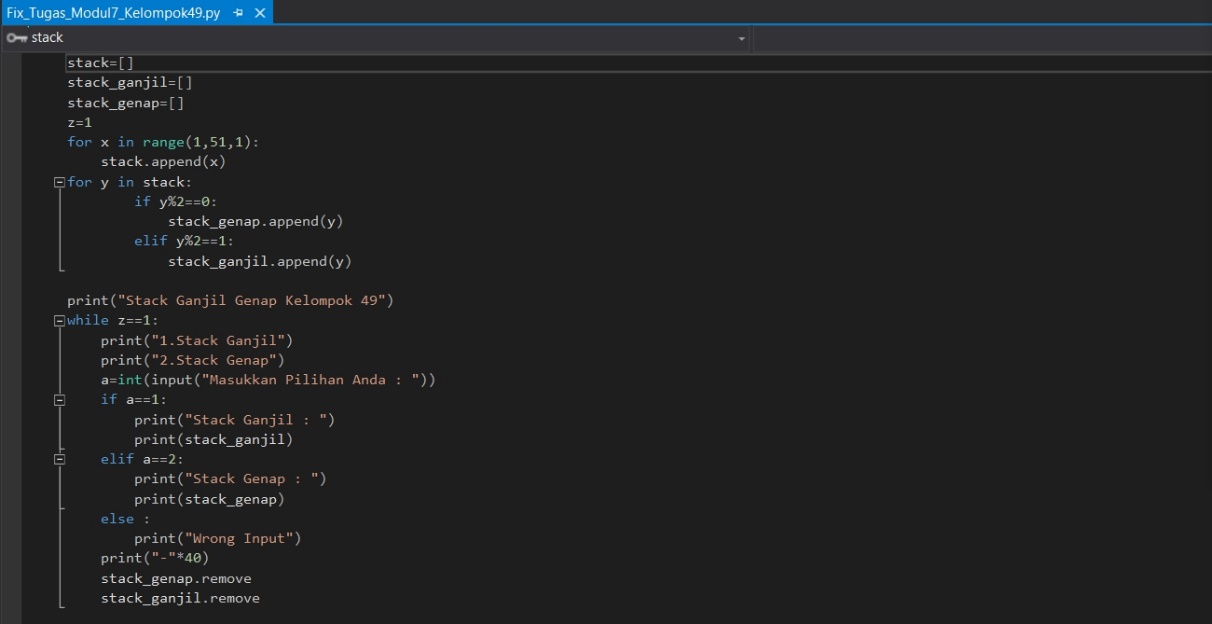


Gambar 8. 16. Output Queue dalam Python

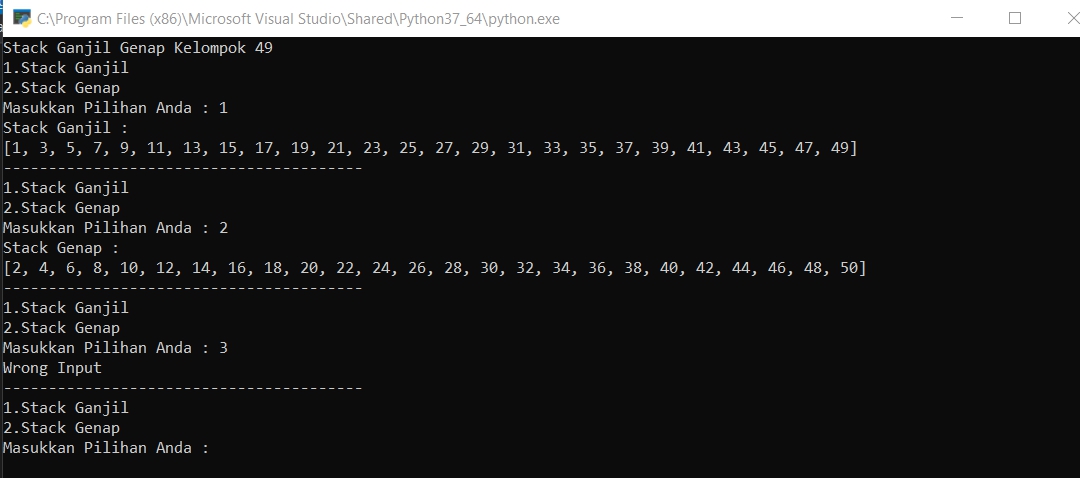
Pertama kita masukkan *package* supaya kita dapat melakukan *queue* serta kita inisialisasikan *queue* kita, selanjutnya kita definisikan variabel-variabel yang akan digunakan dan kita keluarkan fitur-fitur yang tersedia dan berikan *input* supaya pengguna dapat memilih fitur yang diinginkan. Fitur-fitur yang tersedia ini kita pisahkan menggunakan fungsi *if, elif*  dan *else*. Untuk fitur pertama kita yaitu *enqueue Data* kita akan memberikan *input* kepada pengguna untuk menambahkan angka dan selanjutnya angka yang telah dimasukkan akan ditambahkan ke *queue*. Fitur kedua yaitu *dequeue item*, jika pengguna memilih fitur ini maka kita gunakan *syntax dequeue* untuk mengeluarkan *item* pertama. Lalu fitur yang ketiga yaitu mengeluarkan *item-item* yang berada pada *stack* ke layar. dan fitur terakhir adalah lihat data teratas, disini kita langsung mengeluarkan data yang teratas.

## **Tugas**

|  |
| --- |
| stack=[]  stack\_ganjil=[]  stack\_genap=[]  z=1  for x in range(1,51,1):  stack.append(x)  for y in stack:  if y%2==0:  stack\_genap.append(y)  elif y%2==1:  stack\_ganjil.append(y)  print("Stack Ganjil Genap Kelompok 49")  while z==1:  print("1.Stack Ganjil")  print("2.Stack Genap")  a=int(input("Masukkan Pilihan Anda : "))  if a==1:  print("Stack Ganjil : ")  print(stack\_ganjil)  elif a==2:  print("Stack Genap : ")  print(stack\_genap)  else :  print("Wrong Input")  print("-"\*40)  stack\_genap.remove  stack\_ganjil.remove |



Gambar 8. 17. Source Code Tugas dalam Python



Gambar 8. 18. Output Tugas dalam Python

Penjelasan *Output* :

Pertama-tama kita tentukan angka-angka awalnya, disini kami menggunakan angka 1-50. Selanjutnya kita bagi ke *stack* genap dan ganjil dengan cara pengulangan *for* untuk mengambil angka satu-satu lalu kita pisahkan dengan *if* dan *elif*, dengan ketentuan bila angkanya dibagi 2 sisa 0 maka ke *stack*\_genap dan bila dibagi 2 sisa 1 maka ke *stack*\_ganjil. penambahannya menggunakan *syntax append*. Setelah itu kita beri perulangan *while* supaya bisa mengulang dan selanjutnya kita beri pilihan ke pengguna untuk menampilkan *stack* ganjil atau genap. Jika pengguna ingin *stack* genap maka kita keluarkan *stack*\_genap dan sebaliknya bila ingin *stack* ganjil maka kita keluarkan *stack*\_ganjil.

## **Kesimpulan**

* + 1. *Stack* dan *Queue* memiliki hubungan yang saling berkaitan.
    2. *Stack* dan *Queue* memiliki sedikit perbedaan terkait konsep dan perinsipnya.
    3. Penggunaan *Stack* dan *Queue* betujuan agar menghemat ukuran program dan memudahkan dalam mengembangkannya.
    4. Penggunaan *Stack* atau *Queue* dipilih berdasarkan kebutuhan dan urgensinya masing-masing.
    5. Bahasa pemrograman yang keluar dalam waktu dekat, akan memiliki kemampuan yang lebih cepat dan *simple* dari pendahulunya. Seperti phyton lebih *simple* dari C#.
    6. Pemograman mempunyai tiga inti dasar yaitu *input*, proses, dan *output*.
    7. Antara bahasa pemrograman satu dengan yang lainnya kadang terdapat perbedaan-perbendaan tertentu, karena memiliki aturannya masing-masing.
    8. Bahasa pemrogaman tidak dapat dibanding-bandingkan, setiap bahasa pemograman bisa dipakai sesuai kebutuhannya masing-masing.
    9. Diperlukan ketelitian dalam mebuat sebuah program, karena kesalahan sedikit akan membuat program tidak berjalan dengan baik.