**BAB VII**

**STACK DAN QUEUE**

1. **TUJUAN**
2. Praktikan mampu memahami dasar penggunaan stack dan queue pada Java,C#, dan Python.
3. Praktikan mampu membedakan konsep dasar stack dan queue dari bahasa pemrograman yang berbeda (Java, C#, Python).
4. Praktikan mampu mengimplementasikan stack dan queue dalam pemrograman Java, C#, Python.
5. **MATERI**
   1. **Pengertian Stack dan Queue**

Stack (tumpukan) sebenarnya secara mudah dapat diartikan sebagai list (urutan) dimana penambahan dan pengambilan elemen hanya dilakukan pada satu sisi yang disebut top (puncak) dari stack.

Stack memiliki aturan LIFO (Last In First Out), yang berarti elemen terakhir yang masuk akan pertama kali diambil atau dilayani. Dalam stack terdapat 2 operasi dasar, yaitu POP dan PUSH.

Queue atau antrian sebenarnya juga merupakan suatu list. Salah satu sifat yang membedakan queue dengan stack adalah bahwa pada queue penambahan elemen dilakukan pada salah satu ujung (ujung depan) dan pengambilan dilakukan pada ujung yang lain (ujung belakang) .

Queue menggunakan prinsip FIFO (First In First Out), yaitu elemen yang pertama masuk akan pertama kali pula dikeluarkan. Sama seperti stack, di queue memiliki operasi dasar yaitu enqueue dan dequeue yang memiliki fungsi sama pada stack pop dan push.

* 1. **Stack dan Queue pada JAVA**

**Stack pada JAVA**

|  |
| --- |
| **Class stack :**  import java.util.Scanner;  import java.util.Stack;  /\*\*  \*  \* @author acer  \*/  public class JavaApplication2 {  /\*\*  \* @param args the command line arguments  \*/  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  }  } |

Program stack diatas terdapat 5 menu utama yaitu, push, pop, lihat daftar data ,lihat data teratas dan keluar. Push berfungsi untuk menambahkan data ke dalam tumpukan, pop untuk mengurangi data, lihat daftar untuk melihat daftar data dan lihat data teratas untuk melihat daftar data yang ada di urutan atas. Di dalam program terdapat perulangan do while yang berfungsi untuk selalu menampilkan menu utama setelah pengguna memberikan instruksi pada program. Jika data antrian telah kosong setelah dihapus semua, maka program akan berhenti.

**Queue pada Java**

|  |
| --- |
| **Class queue :**  import java.util.Arrays;  import java.util.LinkedList;  import java.util.Queue;  import java.util.Scanner;  /\*\*  \*  \* @author acer  \*/  public class JavaApplication3 {  /\*\*  \* @param args the command line arguments  \*/  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  }  } |

Diatas merupakan hasil dari program queue menggunakan bahasa Java. Program diatas terdapat 5 menu utama. Enqueue untuk menambahkan ke antrian. Dequeue untuk mengurangi data dari antrian. Karna prinsip dari Queue adalah first in first out, maka ketika melakukan dequeue maka data yang terhapus adalah data yang pertama dimasukkan.

* 1. **Stack dan Queue pada C#**

**Stack pada C#**

Stack dapat diilustrasikan sebagai tumpukan data dimana data pertama yang masuk maka data itu berada di paling bawah, untuk memasukkan data dapat digunakan perintah push, misal kita melakukan push 1, push 2, push 3, push 4, maka yang akan dikeluarkan nanti yaitu 1,2,3,4 berurutan dari bawah. Dari data push tadi, jika ingin mengeluarkan data 1, maka proses yang dilakukan pop 4, pop 3, pop 2, pop 1. Untuk lebih jelas berikut contoh list code stack dan queue pada c#.

|  |
| --- |
| using System;  using System.Collections;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApplication2  {  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  }  }  } |

**Queue pada C#**

Berbeda dengan stack, konsep dasar queue yaitu antrian jadi yang pertama dimasukan berada di paling atas. Bila dalam queue apabila kita menuliskan enqueue 1, enqueue 2, enqueue 3, enqueue 4. Dari enqueue tadi maka hasilnya yaitu urut 1,2,3,4 dari atas. Sedangkan fungsi dequeue pada queue sama seperti fungsi pop pada stack. Apabila melakukan dequeue, maka hasilnya 2,3,4.

|  |
| --- |
| using System;  using System.Collections;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApplication2  {  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  }  }  } |

* 1. **Stack dan Queue pada Python**

**Stack pada Python**

|  |
| --- |
| 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 ("") |

Program stack diatas terdapat 4 menu utama yaitu, push, pop, lihat daftar data, dan lihat data teratas. Push berfungsi untuk menambahkan data ke dalam tumpukan, pop untuk mengurangi data teratas, lihat daftar untuk melihat daftar data dan lihat data teratas untuk melihat daftar data yang ada di urutan atas. Di dalam program terdapat while yang berfungsi untuk selalu menampilkan menu utama setelah pengguna memberikan instruksi pada program.

**Queue pada Python**

|  |
| --- |
| 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 ("") |

Diatas merupakan hasil dari program queue menggunakan bahasa Python. Program diatas terdapat 4 menu utama. Enqueue untuk menambahkan ke antrian. Dequeue untuk mengurangi data dari antrian. Karna prinsip dari Queue adalah first in first out, maka ketika melakukan dequeue maka data yang terhapus adalah data yang pertama dimasukkan. lihat daftar untuk melihat daftar data dan lihat data teratas untuk melihat daftar data yang ada di urutan atas. Di dalam program terdapat while yang berfungsi untuk selalu menampilkan menu utama setelah pengguna memberikan instruksi pada program.