



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

8.1. Percobaan 1: Penyimpanan Tumpukan Barang dalam Gudang

```
P8 > J Barang25.java > Barang25 > Barang25(int, String, String)
1 package P8;
2
3 public class Barang25 {
4
5     //atribut
6     int kode;
7     String nama;
8     String kategori;
9
10    public Barang25(int kode, String nama, String kategori){
11        this.kode = kode;
12        this.nama = nama;
13        this.kategori = kategori;
14    }
15 }
16

P8 > J Gudang25.java > ...
1 package P8;
2
3 public class Gudang25 {
4
5     Barang25[] tumpukan;
6     int size;
7     int top;
8
9     public Gudang25(int kapasitas){
10        size = kapasitas;
11        tumpukan = new Barang25[size];
12        top = -1;
13    }
14
15    public boolean cekKosong(){
16        if(top == -1){
17            return true;
18        }else {
19            return false;
20        }
21    }
22
23    public boolean cekPenuh(){
24        if(top == size -1){
25            return true;
26        }else {
27            return false;
28        }
29    }
30
31    public void tambahBarang(Barang25 brg){
32        if(!cekPenuh()){
33            top++;
34            tumpukan[top] = brg;
35            System.out.println("Barang " + brg.nama + " berhasil ditambahkan ke Gudang");
36        }else {
37            System.out.println("Gagal! Tumpukan barang di Gudang sudah penuh");
38        }
39    }
40
41    public Barang25 ambilBarang(){
42        if(!cekKosong()){
43            Barang25 delete = tumpukan[top];
44            top--;
45            System.out.println("Barang " + delete.nama + " diambil dari Gudang.");
46            return delete;
47        }else {
48            System.out.println("Tumpukan barang kosong");
49            return null;
50        }
51    }
52
53    public Barang25 lihatBarangTeratas(){
54        if(!cekKosong()){
55            Barang25 barangTeratas = tumpukan[top];
56            System.out.println("Barang teratas: " + barangTeratas.nama);
57            return barangTeratas;
58        }else {
59            System.out.println("Tumpukan barang kosong");
60            return null;
61        }
62    }
63
64    public void tampilkanBarang(){
65        if(!cekKosong()){
66            System.out.println("Rincian tumpukan barang di Gudang");
67            for(int i = 0; i < top; i++){
68                System.out.printf("KodeKd: %s (Kategori %s)\n", tumpukan[i].kode, tumpukan[i].kategori);
69            }
70        }else {
71            System.out.println("Tumpukan barang kosong");
72        }
73    }
74 }
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

```
PS > J Utama25.java > Utama25 > main(String[])
1 package P8;
2
3 import java.util.Scanner;
4
5 public class Utama25 {
6     Run [Debug]
7     public static void main(String[] args) {
8
9         Gudang2b gudang = new Gudang25(kapasitas:7);
10
11         Scanner scanner = new Scanner(System.in);
12
13         while(true){
14             System.out.println(x:"\nMenu");
15             System.out.println(x:"1. Tambah barang");
16             System.out.println(x:"2. Ambil barang");
17             System.out.println(x:"3. Tampilkan tumpukan barang");
18             System.out.println(x:"4. Keluar");
19             System.out.print(s:"Pilih operasi: ");
20             int pilihan = scanner.nextInt();
21             scanner.nextLine();
22
23             switch (pilihan){
24                 case 1:
25                     System.out.print(s:"Masukkan kode barang: ");
26                     int kode = scanner.nextInt();
27                     scanner.nextLine();
28                     System.out.print(s:"Masukkan nama barang: ");
29                     String nama = scanner.nextLine();
30                     System.out.print(s:"Masukkan nama kategori: ");
31                     String kategori = scanner.nextLine();
32                     Barang2b barangBaru = new Barang25(kode, nama, kategori);
33                     gudang.tambahBarang(barangBaru);
34                     break;
35                 case 2:
36                     gudang.ambilBarang();
37                     break;
38                 case 3:
39                     gudang.tampilkanBarang();
40                     break;
41                 case 4:
42                     break;
43                 default:
44                     System.out.println(x:"Pilihan tidak valid. Silakan coba lagi");
45             }
46         }
47     }
```

8.1.2 Verifikasi Hasil Percobaan

```
PS D:\PrakASD_1F_25> & "C:\Program Files\Java\
Utama25"

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 1
Masukkan kode barang: 21
Masukkan nama barang: Majalah
Masukkan nama kategori: Buku
Barang Majalah berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 1
Masukkan kode barang: 26
Masukkan nama barang: Jaket
Masukkan nama kategori: Pakaian
Barang Jaket berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 2
Barang Jaket diambil dari Gudang.

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 1
Masukkan kode barang: 33
Masukkan nama barang: Pizza
Masukkan nama kategori: Makanan
Barang Pizza berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 3
Rincian tumpukan barang di Gudang
Kode21: Majalah (Kategori Buku)
Kode33: Pizza (Kategori Makanan)
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

Question :

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana saja yang perlu diperbaiki

Answer :

- a. Pada method lihatBarangTeratas if(!isEmpty()) diganti dengan if(!cekKosong())

```
public Barang25 lihatBarangTeratas(){  
    // if(!isEmpty()) { kode lama  
    if(!cekKosong()){ //kode baru  
        Barang25 barangTeratas = tumpukan[top];  
        System.out.println("Barang teratas: " + barangTeratas.nama);  
        return barangTeratas;  
    } else {  
        System.out.println(x:"Tumpukan barang kosong");  
        return null;  
    }  
}
```

- b. Pada method tampilkanBarang() kode for(int i = 0; i<=top; i++) diganti menjadi

```
public void tampilkanBarang(){  
    if(!cekKosong()){  
        System.out.println(x:"Rincian tumpukan barang di Gudang");  
  
        //for(int i = 0; i<=top; i++){ kode lama  
        for(int i = top; i >= 0; i--){ // kode baru  
            System.out.printf(format:"Kode%d: %s (Kategori %s)\n", tumpukan[i].kode, tumpukan[i].nama, tumpukan[i].kategori);  
        }  
    } else {  
        System.out.println(x:"Tumpukan barang kosong");  
    }  
}
```

- c. Pada class utama, seharusnya dideklarasikan Boolean supaya perulangan bisa berhenti jika kondisi tertentu.

```
public static void main(String[] args) {  
    Gudang25 gudang = new Gudang25(kapasitas:7);  
    Scanner scanner = new Scanner(System.in);  
    boolean konfirmasi = true; //kode baru  
  
    //while(true){ kode lama  
    while(konfirmasi){ //kode baru  
        System.out.println(x:"\nMenu");  
  
        break;  
    }  
    case 4:  
        konfirmasi = false; //kode baru  
        break;  
    default:
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

Hasil Run Sesuai Verifikasi

```
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 1
Masukkan kode barang: 21
Masukkan nama barang: Majalah
Masukkan nama kategori: Buku
Barang Majalah berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 1
Masukkan kode barang: 26
Masukkan nama barang: Jaket
Masukkan nama kategori: Pakaian
Barang Jaket berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 2
Barang Jaket diambil dari Gudang.

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 1
Masukkan kode barang: 33
Masukkan nama barang: Pizza
Masukkan nama kategori: Makanan
Barang Pizza berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 3
Rincian tumpukan barang di Gudang
Kode33: Pizza (Kategori Makanan)
Kode21: Majalah (Kategori Buku)

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 4
PS D:\PrakASD_1F_25> []
```

2. Berapa banyak data barang yang dapat ditampung di dalam tumpukan? Tunjukkan potongan kode programnya!

Answer:

Data yang dapat ditampung dalam tumpukan adalah 7.

```
Run | Debug
public static void main(String[] args) {
    Gudang25 gudang = new Gudang25(kapasitas:7);
}
```

3. Mengapa perlu pengecekan kondisi !cekKosong() pada method tampilkanBarang? Kalau kondisi tersebut dihapus, apa dampaknya?

Answer :



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

Pengecekan kondisi !cekKosong() perlu dilakukan untuk mengecek data dalam stack kosong atau tidak. Apabila dihapus, maka tidak akan terdeteksi barang yang ada dalam stack kosong atau ada isinya.

4. Modifikasi kode program pada class Utama sehingga pengguna juga dapat memilih operasi lihat barang teratas, serta dapat secara bebas menentukan kapasitas Gudang!

Answer:

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print(s:"Tentukan Kapasitas Gudang: "); //modifikasi pertanyaan 4
    int kapasitas = scanner.nextInt();
    Gudang25 gudang = new Gudang25(kapasitas);
    System.out.print("Kapasitas gudang saat ini adalah: " + kapasitas);
    boolean konfirmasi = true; //kode baru

    while(konfirmasi){ //kode baru
        System.out.println(x:"\nMenu");
        System.out.println(x:"1. Tambah barang");
        System.out.println(x:"2. Ambil barang");
        System.out.println(x:"3. Tampilkan tumpukan barang");
        System.out.println(x:"4. Lihat Barang Teratas"); //modifikasi pertanyaan 4
        System.out.println(x:"5. Keluar");
        System.out.print(s:"Pilih operasi: ");
        int pilihan = scanner.nextInt();
        scanner.nextLine();

        break;
    case 2:
        gudang.ambilBarang();
        break;
    case 3:
        gudang.tampilkanBarang();
        break;
    case 4:
        gudang.lihatBarangTeratas(); //modifikasi pertanyaann 4
        break;
    case 5:

```

```
\79c876eb0fa3185450a0db594819182e:redhat.java\j
Tentukan Kapasitas Gudang: 3
Kapasitas gudang saat ini adalah: 3
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 1
Masukkan kode barang: 21
Masukkan nama barang: Majalah
Masukkan nama kategori: Buku
Barang Majalah berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 1
Masukkan kode barang: 33
Masukkan nama barang: Pizza
Masukkan nama kategori: Makanan
Barang Pizza berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 4
Barang teratas: Pizza

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 5
PS D:\PrakASD 1F 25>
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

5. Commit dan push kode program ke Github

Answer :

```
salsabila mahda@DESKTOP-135ONBT MINGW64 /d/PrakASD_1F_25 (master)
$ git add .

salsabila mahda@DESKTOP-135ONBT MINGW64 /d/PrakASD_1F_25 (master)
$ git commit -m "P8_percobaan1_Pertanyaan4_1F_25"
[master f4e3fa8] P8_percobaan1_Pertanyaan4_1F_25
1 file changed, 9 insertions(+), 2 deletions(-)

salsabila mahda@DESKTOP-135ONBT MINGW64 /d/PrakASD_1F_25 (master)
$ git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 564 bytes | 564.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local object
To https://github.com/salsabilamhda/PrakASD_1F_25.git
cceb9db..f4e3fa8 master -> master
branch 'master' set up to track 'origin/master'.

salsabila mahda@DESKTOP-135ONBT MINGW64 /d/PrakASD_1F_25 (master)
$
```

8.2 Percobaan 2 : Konversi Kode Barang ke Biner

```
//percobaan 2
public String konversiDesimalKeBiner(int kode){
    StackKonversi25 stack = new StackKonversi25();
    while(kode>0){
        int sisa = kode % 2;
        stack.push(sisa);
        kode = kode / 2;
    }
    String biner = new String();
    while (!stack.isEmpty()){
        biner += stack.pop();
    }
    return biner;
}

P8 > J StackKonversi25.java > StackKonversi25 > pop()
1 package P8;
2
3 public class StackKonversi25 {
4
5     int size;
6     int[] tumpukanBiner;
7     int top;
8
9     public StackKonversi25(){
10         this.size = 32; //asumsi 32 bit
11         tumpukanBiner = new int[size];
12         top = -1;
13     }
14
15     public boolean isEmpty(){
16         return top == -1;
17     }
18
19     public boolean isFull(){
20         return top == size - 1;
21     }
22
23     public void push(int data){
24         if(isFull()){
25             System.out.println(x:"Stack penuh");
26         }else {
27             top++;
28             tumpukanBiner[top] = data;
29         }
30     }
31
32     public int pop(){
33         if(isEmpty()){
34             System.out.println(x:"Stack kosong");
35             return -1;
36         }else {
37             int data = tumpukanBiner[top];
38             top--;
39             return data;
40         }
41     }
42 }
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

```
public Barang25 ambilBarang(){
    if(!cekKosong()){
        Barang25 delete = tumpukan[top];
        top--;
        System.out.println("Barang " + delete.nama + " diambil dari Gudang.");
        System.out.println("Kode unik dalam biner: " + konversiDesimalKeBiner(delete.kode)); //percobaan 2
        return delete;
    }else {
        System.out.println(x:"Tumpukan barang kosong");
        return null;
    }
}
```

8.2.1 Verifikasi Hasil Percobaan

```
Kapasitas Gudang Saat Ini Gudang: 0
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 1
Masukkan kode barang: 13
Masukkan nama barang: Setrika
Masukkan nama kategori: Elektronik
Barang Setrika berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 2
Barang Setrika diambil dari Gudang.
Kode unik dalam biner: 1101

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 5
PS D:\PrakASD 1F 25>
```

Question :

1. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while(kode!=0), bagaimana hasilnya? Jelaskan alasannya!

Answer:

```
//percobaan 2
public String konversiDesimalKeBiner(int kode){
    StackKonversi25 stack = new StackKonversi25();
    //while(kode > 0){ kode lama
    while(kode!=0){ //kode baru
        int sisa = kode % 2;
        stack.push(sisa);
        kode = kode / 2;
    }
    String biner = new String();
    while (!stack.isEmpty()){
        biner += stack.pop();
    }
    return biner;
}
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

```
PS D:\PrakASD_1F_25> & 'C:\Program Files\Java\
ta\Roaming\Code\User\workspaceStorage\79c876eb
Tentukan Kapasitas Gudang: 3
Kapasitas gudang saat ini adalah: 3
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 1
Masukkan kode barang: 13
Masukkan nama barang: Setrika
Masukkan nama kategori: Elektronik
Barang Setrika berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 2
Barang Setrika diambil dari Gudang.
Kode unik dalam biner: 1101

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 5
PS D:\PrakASD_1F_25> ^C
PS D:\PrakASD_1F_25>
```

Hasilnya sama seperti sebelumnya, karena kondisi tersebut berarti jika kode tidak sama dengan 0, dan kode sebelumnya berarti kode lebih besar dari 0, maka hasilnya akan sama aja. Kecuali yang kode yang diinput berupa angka minus, maka kode uniknya sebagai berikut

```
0.00000000
Tentukan Kapasitas Gudang: 2
Kapasitas gudang saat ini adalah: 2
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 1
Masukkan kode barang: -13
Masukkan nama barang: Setrika
Masukkan nama kategori: Elektronik
Barang Setrika berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 2
Barang Setrika diambil dari Gudang.
Kode unik dalam biner: -1-10-1

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Keluar
Pilih operasi: 1
```

Hal tersebut dikarenakan program tidak bisa mengeksekusi angka yang kurang dari nol atau minus. Dan jika yang diinputkan angka 0, maka tidak ada output kode unik.

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

Answer :



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

1. Metode 'konversiDesimalKeBiner' menerima satu parameter 'kode' yang merupakan bilangan desimal yang ingin dikonversi menjadi bilangan biner. Tipe data parameter adalah 'int'
2. Dalam metode tersebut, sebuah objek 'StackKonversi' dibuat untuk menyimpan sisa pembagian bilangan desimal dengan 2. Ini adalah bagian dari algoritma konversi desimal ke biner dengan menggunakan operasi pembagian dan modulus.
3. Selama nilai 'kode' masih lebih besar dari 0, langkah-langkah berikut diulang:
 - Hitung sisa pembagian ('sisa') dari 'kode' dibagi dengan 2.
 - Sisa tersebut kemudian dimasukkan ke dalam stack menggunakan metode 'push'.
 - Nilai 'kode' diperbarui dengan pembagian bilangan desimal tersebut dengan 2, sehingga pada iterasi berikutnya, kita akan mendapatkan digit berikutnya dari biner.
4. Setelah selesai iterasi pertama, dilakukan iterasi kedua dengan tujuan untuk mengambil nilai dari stack dan membangun string biner.
 - String 'biner' dibuat.
 - Selama stack tidak kosong ('!stack.isEmpty()'), digit dari biner diambil dari stack menggunakan metode 'pop()', dan ditambahkan ke string 'biner'.
5. String 'biner' yang berisi representasi biner dari bilangan desimal tersebut dikembalikan sebagai output dari metode.

8.3 Percobaan 3 : Konversi Notasi Infix ke Postfix

```
1 // Postfix.java - Konversi Infix ke Postfix
2 package konversi;
3
4 public class Postfix {
5     int top;
6     int total;
7     char stack[];
8
9     public Postfix(int total){
10         this.total = total;
11         top = -1;
12         stack = new char[total];
13     }
14
15     public void push(char c){
16         top++;
17         stack[top] = c;
18     }
19
20     public char pop(){
21         char item = stack[top];
22         top--;
23         return item;
24     }
25
26     public boolean isOperator(char c){
27         if ((c == '+' || c == '-' || c == '*' || c == '/') ||
28             (c == '^' || c == '%')) {
29             return true;
30         } else {
31             return false;
32         }
33     }
34
35     public boolean isOperand(char c){
36         if ((c == 'a' || c == 'b' || c == 'c' || c == 'd' || c == 'e' || c == 'f' || c == 'g' || c == 'h' || c == 'i' || c == 'j' || c == 'k' || c == 'l' || c == 'm' || c == 'n' || c == 'o' || c == 'p' || c == 'q' || c == 'r' || c == 's' || c == 't' || c == 'u' || c == 'v' || c == 'w' || c == 'x' || c == 'y' || c == 'z' || c == 'A' || c == 'B' || c == 'C' || c == 'D' || c == 'E' || c == 'F' || c == 'G' || c == 'H' || c == 'I' || c == 'J' || c == 'K' || c == 'L' || c == 'M' || c == 'N' || c == 'O' || c == 'P' || c == 'Q' || c == 'R' || c == 'S' || c == 'T' || c == 'U' || c == 'V' || c == 'W' || c == 'X' || c == 'Y' || c == 'Z')) {
37             return true;
38         } else {
39             return false;
40         }
41     }
42 }
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

```
public int derajat(char c){
    switch (c){
        case '+':
            return 3;
        case '%':
            return 2;
        case '/':
            return 2;
        case '*':
            return 2;
        case '-':
            return 1;
        case '+':
            return 1;
        default:
            return 0;
    }
}

public String konversi(String Q){
    String P = " ";
    char c;
    for(int i = 0; i < n; i++){
        c = Q.charAt(i);

        if(!IsOperand(c)){
            P = P + c;
        }
        if (c == '('){
            push(c);
        }
        if (c == ')'){
            while (stack[top] != '('){
                P = P + pop();
            }
            pop();
        }
        if (IsOperator(c)){
            while (derajat(stack[top]) >= derajat(c)){
                P = P + pop();
            }
            push(c);
        }
    }
    return P;
}
```

```
PS > J PostfixMain25.java > PostfixMain25 > main(String[])
1 package RB;
2
3 import java.util.Scanner;
4
5 public class PostfixMain25 {
6     public static void main(String[] args) {
7         Scanner sc25 = new Scanner(System.in);
8         String P,Q;
9         System.out.println(x:"Masukkan ekspresi matematika (infix) : ");
10        Q = sc25.nextLine();
11        Q = Q.trim();
12        Q = Q + " ";
13
14        int total = Q.length();
15        Postfix25 post = new Postfix25(total);
16        P = post.konversi(Q);
17        System.out.println("Postfix: " + P);
18    }
19 }
```

8.3.2 Verifikasi Hasil Percobaan

```
ta\Roaming\Code\User\workspaceStorage\7
Masukkan ekspresi matematika (infix) :
a+b*(c+d-e)/f
Postfix: abcd+e-*f/+
PS D:\PrakASD_1F_25>
```

Question :

1. Pada method derajat, mengapa return value beberapa case bernilai sama? Apabila return value diubah dengan nilai berbeda-beda setiap case-nya, apa yang terjadi?

Answer:



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

```
public int derajat(char c){  
    switch (c){  
        case '^':  
            return 6;  
        case '%':  
            return 5;  
        case '/':  
            return 4;  
        case '*':  
            return 3;  
        case '-':  
            return 2;  
        case '+':  
            return 1;  
        default:  
            return 0;  
    }  
}
```

```
Users\salsabila mahda\AppData\Roaming\Q  
8.PostfixMain25'  
Masukkan ekspresi matematika (infix) :  
a+b*(c+d-e)/f  
Postfix: abcde-+f/*+  
PS D:\PrakASD_1F_25>
```

Return value beberapa case bernilai sama karena menunjukkan tingkat prioritas dari suatu operasi. Semakin tinggi nilai suatu operasi, maka semakin tinggi pula prioritas operasi tersebut. Dalam method derajat, return value yang memiliki nilai sama berarti memiliki kesetaraan yang sama dengan operasi dengan return value yang sama. Apabila return value memiliki nilai yang berbeda – beda setiap case-nya, maka tingkat prioritas juga berbeda, dan tidak sesuai dengan aturan prioritas suatu operasi.

2. Jelaskan alur kerja method konversi?

Answer:

1. Inisialisasi variable :

- 'P' adalah string yang akan menyimpan ekspresi postfix.
- 'c' adalah variable karakter yang digunakan untuk membaca setiap karakter dari ekspresi input 'Q'

2. Iterasi Melalui Ekspresi Input:

- Iterasi dilakukan melalui setiap karakter dari ekspresi input 'Q'

3. Memeriksa karakter:

- Jika karakter adalah operand, maka karakter tersebut langsung ditambahkan ke dalam String 'P'
- Jika karakter adalah tanda kurung buka, maka tanda kurung tersebut dimasukkan ke dalam stack.
- Jika karakter adalah tanda kurung tutup, maka dilakukan proses pengosongan stack hingga menemukan tanda kurung buka.
- Jika karakter adalah operator, maka dilakukan pengecekan derajat operator pada stack. Jika derajat operator pada stack lebih besar atau sama dengan derajat operator yang sedang di proses, maka operator stack dihapus dan dimasukkan ke dalam string 'P'

4. Pengambilan String Hasil:

- Setelah iterasi selesai, hasil ekspresi postfix disimpan dalam string 'P' dan dikembalikan

5. Fungsi Tambahan

- Fungsi 'IsOperand' digunakan untuk memeriksa apakah suatu karakter adalah operand



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

- Fungsi 'IsOperator' digunakan untuk memeriksa apakah suatu karakter adalah operator.
- Fungsi 'derajat' digunakan untuk menentukan prioritas (derajat) operator.

3. Pada method konversi, apa fungsi dari potongan kode berikut?

```
c = Q.charAt(i);
```

Answer:

Fungsi potongan kode tersebut adalah untuk mengambil karakter pada posisi indeks 'i' dari string 'Q' dan menyimpan ke dalam variable 'c'. dalam konteks fungsi konversi, 'Q' adalah ekspresi matematika dalam bentuk string yang ingin diubah dari infix menjadi postfix. Dengan menggunakan 'charAt(i)', kita dapat mengakses karakter satu per satu dari string tersebut dalam proses iterasi.

8.4 Latihan Praktikum

Perhatikan dan gunakan Kembali kode program pada percobaan 1. Tambahkan dua method berikut pada class Gudang:

- Method lihatBarangTerbawah digunakan untuk mengecek barang pada tumpukan terbawah

Answer :

```
//Latihan Praktikum
public Barang25 lihatBarangTerbawah(){
    if(!cekKosong()){
        Barang25 barangTerbawah = tumpukan[0];
        System.out.println("Barang terbawah: " + barangTerbawah.nama);
        return barangTerbawah;
    }else {
        System.out.println(x:"Tumpukan barang kosong");
        return null;
    }
}

while(konfirmasi){ //kode baru
    System.out.println(x:"\nMenu");
    System.out.println(x:"1. Tambah barang");
    System.out.println(x:"2. Ambil barang");
    System.out.println(x:"3. Tampilkan tumpukan barang");
    System.out.println(x:"4. Lihat Barang Teratas"); //modifikasi pertanyaan 4
    System.out.println(x:"5. Lihat Barang Terbawah"); //Latihan Praktikum 1
    System.out.println(x:"6. Keluar");
    System.out.print(s:"Pilih operasi: ");
    int pilihan = scanner.nextInt();
    scanner.nextLine();

    switch (pilihan){
        case 1:
            System.out.print(s:"Masukkan kode barang: ");
            int kode = scanner.nextInt();
            scanner.nextLine();
            System.out.print(s:"Masukkan nama barang: ");
            String nama = scanner.nextLine();
            System.out.print(s:"Masukkan nama kategori: ");
            String kategori = scanner.nextLine();
            Barang25 barangBaru = new Barang25(kode, nama, kategori);
            gudang.tambahBarang(barangBaru);
            break;
        case 2:
            gudang.ambilBarang();
            break;
        case 3:
            gudang.tampilkanBarang();
            break;
        case 4:
            gudang.lihatBarangTeratas(); //modifikasi pertanyaan 4
            break;
        case 5:
            gudang.lihatBarangTerbawah(); //Latihan Praktikum 1
            break;
        case 6:
            konfirmasi = false; //kode baru
            break;
        default:
    }
}
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

```
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Keluar
Pilih operasi: 1
Masukkan kode barang: 21
Masukkan nama barang: Majalah
Masukkan nama kategori: Buku
Barang Majalah berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Keluar
Pilih operasi: 1
Masukkan kode barang: 26
Masukkan nama barang: Jaket
Masukkan nama kategori: Pakaian
Barang Jaket berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Keluar
Pilih operasi: 1
Masukkan kode barang: 33
Masukkan nama barang: Pizza
Masukkan nama kategori: Makanan
Barang Pizza berhasil ditambahkan ke Gudang

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Keluar
Pilih operasi: 5
Barang terbawah: Majalah

Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Keluar
Pilih operasi: 6
PS D:\PrakASD_1F_25>
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

- Method cariBarang digunakan untuk mencari ada atau tidaknya barang berdasarkan kode barangnya atau nama barangnya.

Answer :

```
PG > J Gudang25.java X J Utama25.java M
3 public class Gudang25 {
//Latihan Praktikum 2
108 public int CariBarangKode(int cari){
109     int posisi = -1;
110     for(int j = 0; j<tumpukan.length; j++){
111         if(tumpukan[j].kode == cari){
112             posisi = j;
113             break;
114         }
115     }
116     return posisi;
117 }
118
119 public void TampilDataKode(int x, int pos){
120     if(pos != -1){
121         System.out.println("Kode Barang : " + tumpukan[pos].kode);
122         System.out.println("Nama Barang : " + tumpukan[pos].nama);
123         System.out.println("Kategori Barang : " + tumpukan[pos].kategori);
124     } else {
125         System.out.println("Data " + x + " tidak ditemukan");
126     }
127 }
128
129 public void TampilPosisiKode(int x, int pos){
130     if(pos != -1){
131         System.out.println("Data : " + x + " ditemukan pada indeks " + pos);
132     } else {
133         System.out.println("Data " + x + " tidak ditemukan");
134     }
135 }
136
137 public int CariBarangNama(String cari2){
138     int posisi = -1;
139     for(int j = 0; j<tumpukan.length; j++){
140         if (tumpukan[j] != null && tumpukan[j].nama.equalsIgnoreCase(cari2)){
141             posisi = j;
142             break;
143         }
144     }
145     return posisi;
146 }
147
148 public void TampilDataNama(String x, int pos){
149     if(pos != -1){
150         System.out.println("Kode Barang : " + tumpukan[pos].kode);
151         System.out.println("Nama Barang : " + tumpukan[pos].nama);
152         System.out.println("Kategori Barang : " + tumpukan[pos].kategori);
153     } else {
154         System.out.println("Data " + x + " tidak ditemukan");
155     }
156 }
157
158 public void TampilPosisiNama(String x, int pos){
159     if(pos != -1){
160         System.out.println("Data : " + x + " ditemukan pada indeks " + pos);
161     } else {
162         System.out.println("Data " + x + " tidak ditemukan");
163     }
164 }
165 }
166
while(konfirmasi){ //kode baru
System.out.println(x:"\nMenu");
System.out.println(x:"1. Tambah barang");
System.out.println(x:"2. Ambil barang");
System.out.println(x:"3. Tampilkan tumpukan barang");
System.out.println(x:"4. Lihat Barang Teratas"); //modifikasi pertanyaan 4
System.out.println(x:"5. Lihat Barang Terbawah"); //Latihan Praktikum 1
System.out.println(x:"6. Cari Barang"); //Latihan Praktikum 2
System.out.println(x:"7. Keluar");
System.out.print(s:"Pilih operasi: ");
int pilihan = scanner.nextInt();
scanner.nextLine();

break;
case 6: //Latihan Praktikum 2
System.out.println(x:"Cari Barang Berdasarkan: ");
System.out.println(x:"1. Kode Barang");
System.out.println(x:"2. Nama barang");
System.out.print(s:"Masukkan Pilihan Anda: ");
int choice = scanner.nextInt();
switch(choice){
case 1:
System.out.println(x:"=====");
System.out.println(x:"Pencarian berdasarkan Kode Barang");
System.out.print(s:"Masukkan kode barang: ");
int cari = scanner.nextInt();
int posisi = gudang.CariBarangKode(cari);
gudang.TampilPosisiKode(cari, posisi);
gudang.TampilDataKode(cari, posisi);
break;
case 2:
System.out.println(x:"=====");
System.out.println(x:"Pencarian berdasarkan Nama Barang");
scanner.nextLine();
System.out.print(s:"Masukkan nama barang: ");
String cari2 = scanner.nextLine();
posisi = gudang.CariBarangNama(cari2);
if (posisi != -1) {
gudang.TampilPosisiNama(cari2, posisi);
gudang.TampilDataNama(cari2, posisi);
} else {
System.out.println("Data " + cari2 + " tidak ditemukan");
}
break;
default:
System.out.println(x:"Pilihan yang kamu masukkan tidak valid.");
break;
}
}
case 7:
konfirmasi = false; //kode baru
break;
default:
System.out.println(x:"Pilihan tidak valid. Silakan coba lagi!");
}
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : STACK

```
Tentukan Kapasitas Gudang: 3
Kapasitas gudang saat ini adalah: 3
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Cari Barang
7. Keluar
Pilih operasi: 1
Masukkan kode barang: 21
Masukkan nama barang: Majalah
Masukkan nama kategori: Buku
Barang Majalah berhasil ditambahkan ke Gudang
```

```
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Cari Barang
7. Keluar
Pilih operasi: 1
Masukkan kode barang: 26
Masukkan nama barang: Jaket
Masukkan nama kategori: Pakaian
Barang Jaket berhasil ditambahkan ke Gudang
```

```
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Cari Barang
7. Keluar
Pilih operasi: 1
Masukkan kode barang: 33
Masukkan nama barang: Pizza
Masukkan nama kategori: Makanan
Barang Pizza berhasil ditambahkan ke Gudang
```

```
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Cari Barang
7. Keluar
Pilih operasi: 6
Cari Barang Berdasarkan:
1. Kode Barang
2. Nama barang
Masukkan Pilihan Anda: 1
=====
Pencarian berdasarkan Kode Barang
Masukkan kode barang: 26
Data: 26 ditemukan pada indeks 1
Kode Barang : 26
Nama Barang : Jaket
Kategori Barang : Pakaian
```

```
Menu
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat Barang Teratas
5. Lihat Barang Terbawah
6. Cari Barang
7. Keluar
Pilih operasi: 6
Cari Barang Berdasarkan:
1. Kode Barang
2. Nama barang
Masukkan Pilihan Anda: 2
=====
Pencarian berdasarkan Nama Barang
Masukkan nama barang: Majalah
Data: Majalah ditemukan pada indeks 0
Kode Barang : 21
Nama Barang : Majalah
Kategori Barang : Buku
```