



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

1. Praktikum 1 : Penyimpanan barang dalam gudang.

```
view Go ... ← → Algorithm_dan_Struktur_Data_1G_01
J Barang01.java U X J Gudang01.java U J Utama01.java 1, U J Tugas2Jobsheet1.java 3 J PencarianBankAdani.
src > P8 > J Utama01
1 package P8;
2
3 public class Barang01 {
4
5     int kode;
6     String nama;
7     String kategori;
8
9     Barang01(int kode, String nama, String kategori) {
10         this.kode = kode;
11         this.nama = nama;
12         this.kategori = kategori;
13     }
14 }
15
```

```
on View Go ... ← → Algorithm_dan_Struktur_Data_1G_01
J Barang01.java U X J Gudang01.java U J Utama01.java 1, U J Tugas2Jobsheet1.java 3 J PencarianBankAdani.java U J BankAdani.java C:\...\Rar$D
src > P8 > J Gudang01.java
1 package P8;
2
3 public class Gudang01 {
4     Barang01[] tumpukan;
5     int size;
6     int top;
7
8     public Gudang01(int kapasitas) {
9         size = kapasitas;
10        tumpukan = new Barang01 [size];
11        top = -1;
12    }
13
14    public boolean cekKosong() {
15        if (top == -1) {
16            return true;
17        } else {
18            return false;
19        }
20    }
21
22    public boolean cekPenuh() {
23        if (top == size -1) {
24            return true;
25        } else {
26            return false;
27        }
28    }
29
30    public void tambahBarang(Barang01 brg) {
31        if (!cekPenuh()) {
32            top++;
33            tumpukan[top] = brg;
34            System.out.println("Barang " + brg.nama + " berhasil ditambahkan ke Gudang");
35        } else {
36            System.out.println(x:"Gagal! Tumpukan barang di gudang sudah penuh");
37        }
38    }
39
40}
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

```
Algorithm_dan_Struktur_Data_1G_01

J Barang01.java II  Gudang01.java X  Utama01.java U  Tugas2/Jobsheet1.java 3  PencarianBankAdani.java U  BankAdani.java C:\...\Rar$Dla12348

src > P8 > J Gudang01.java  Utama01 >

3 public class Gudang01 {
4
5     public Barang01 ambilBarang() {
6         if (cekKosong()) {
7             Barang01 delete = tumpukan[top];
8             top--;
9             System.out.println("Barang " + delete.nama + " diambil dari Gudang.");
10            return delete;
11        } else {
12            System.out.println(x:"Tumpukan barang kosong.");
13            return null;
14        }
15    }
16
17    public Barang01 lihatBarangTeratas() {
18        if (cekKosong()) {
19            Barang01 barangTeratas = tumpukan[top];
20            System.out.println("Barang teratas : " + barangTeratas.nama);
21            return barangTeratas;
22        } else {
23            System.out.println(x:"Tumpukan barang kosong.");
24            return null;
25        }
26    }
27
28    public void tampilkanBarang() {
29        if (cekKosong()) {
30            System.out.println(x:"Rincian tumpukan barang di Gudang.");
31            for (int i = top; i >= 0; i--) {
32                System.out.printf(format:"Kode %d: %s (Kategori %s)\n", tumpukan[i].kode, tumpukan[i].nama, tumpukan[i].kategori);
33            }
34        } else {
35            System.out.println(x:"Tumpukan barang kosong.");
36        }
37    }
38
39 }
40
41
42
43
44
45
46
47
```

```
Algorithm_dan_Struktur_Data_1G_01

Java _UTS_Adani  Utama01  Utama01.java U  Gudang01.java U  Utama01.java t.u X  Tugas2/Jobsheet1.java 3  PencarianBankAdani.java U

src > P8 > J Utama01.java > Utama01 > @main(String[])
1 package P8;
2
3 import java.util.Scanner;
4
5 public class Utama01 {
6     Run/Debug
7     public static void main(String[] args) {
8         Gudang01 gudang = new Gudang01(kapasitas:7);
9         Scanner scanner = new Scanner(System.in);
10
11         while (true) {
12             System.out.println(x:"\nMenu : ");
13             System.out.println(x:"1. Tambah barang");
14             System.out.println(x:"2. Ambil barang");
15             System.out.println(x:"3. Tampilkan tumpukan barang");
16             System.out.println(x:"4. Keluar");
17             System.out.println(x:"Pilih Operasi :");
18             int pilihan = scanner.nextInt();
19
20             switch (pilihan) {
21                 case 1:
22                     System.out.println(x:"Masukkan kode barang : ");
23                     int kode = scanner.nextInt();
24                     scanner.nextLine();
25                     System.out.println(x:"Masukkan nama barang : ");
26                     String nama = scanner.nextLine();
27                     System.out.println(x:"Masukkan nama kategori : ");
28                     String kategori = scanner.nextLine();
29                     Barang01 barangBaru = new Barang01(kode, nama, kategori);
30                     gudang.tambahBarang(barangBaru);
31                     break;
32                 case 2:
33                     gudang.ambilBarang();
34                     break;
35                 case 3:
36                     gudang.tampilkanBarang();
37                     break;
38                 case 4:
39                     System.exit(status:0); // Exiting the program
40                     break;
41                 default:
42                     System.out.println(x:"Pilihan tidak valid");
43                     break;
44             }
45         }
46     }
47 }
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01\bin' 'P8.Utama01'

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
```

```
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 :
Seblak
Masukkan nama kategori :
Makanan
Barang Seblak 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.
Kode 33: Seblak (Kategori Makanan)
Kode 21: Majalah (Kategori Buku)
```

Pertanyaan!



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

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

Jawab :

- a. Pada kode di bawah ini, tadinya (!isEmpty())

```
public Barang01 lihatBarangTeratas() {  
    if (!cekKosong()) {
```

- b. Pada method tampilkanBarang()

```
for (int i = top; i >= 0; i--) {
```

- c. Pada switch case (case 4) ditambahkan System.exit.

```
case 4 :  
    System.exit(0);
```

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

Jawab : Tujuh barang.

```
public class Utama01 {  
    public static void main(String[] args) {  
        Gudang01 gudang = new Gudang01(7);
```

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

Jawab : Tanpa pengecekan kondisi tersebut, jika tumpukan barang kosong, program akan mencoba untuk menampilkan barang-barangnya tanpa memeriksa apakah data barang tersedia. Jika kondisi tersebut dihapus, maka program akan mencoba untuk menampilkan barang-barang dalam tumpukan tanpa memeriksa apakah tumpukan kosong dan dapat menyebabkan kesalahan saat menjalankan program, dan program mungkin mencoba untuk mengakses elemen tumpukan yang tidak ada.

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

```
while (true) {  
    System.out.println("\nMenu : ");  
    System.out.println("1. Tambah barang");  
    System.out.println("2. Ambil barang");  
    System.out.println("3. Tampilkan tumpukan barang");  
    System.out.println("4. Lihat barang teratas.");  
    System.out.println("5. Keluar");  
    System.out.println("Pilih Operasi : ");  
    int pilihan = scanner.nextInt();  
  
    switch (pilihan) {  
        case 1 :  
            System.out.println("Masukkan kode barang : ");  
            int kode = scanner.nextInt();  
            scanner.nextLine();  
            System.out.println("Masukkan nama barang : ");  
            String nama = scanner.nextLine();  
            System.out.println("Masukkan nama kategori : ");  
            String kategori = scanner.nextLine();  
            Barang01 barangbaru = new Barang01(kode, nama, kategori);  
            gudang.tambahBarang(barangbaru);  
            break;  
        case 2 :  
            gudang.ambilBarang();  
            break;  
        case 3 :  
            gudang.tampilkanBarang();  
            break;  
        case 4 :  
            gudang.lihatBarangTeratas();  
            break;  
        case 5 :  
            System.exit(0);  
            break;  
        default :  
            System.out.println("Pilihan tidak valid");  
            break;  
    }  
}
```

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE T  
PS D:\college\semester 2\Algoritma dan S  
Program Files\Java\jdk-20\bin\java.exe'  
ptionMessages' '-cp' 'D:\college\semeste  
Data_1G_01\bin' 'P8.Utama01'  
  
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Lihat barang teratas.  
5. Keluar  
Pilih Operasi :  
4  
Tumpukan barang kosong.
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

2. Praktikum 2 : Konversi Kode Barang ke Biner.

```
> PS > J Gudang01.java > % Gudang01
3 public class Gudang01 {
4     Barang01[] tumpukan;
5     int size;
6     int top;
7
8     public Gudang01(int kapasitas) {
9         size = kapasitas;
10        tumpukan = new Barang01 [size];
11        top = -1;
12    }
13
14    public boolean cekKosong() {
15        if (top == -1) {
16            return true;
17        } else {
18            return false;
19        }
20    }
21
22    public boolean cekPenuh() {
23        if (top == size -1) {
24            return true;
25        } else {
26            return false;
27        }
28    }
29
30    public void tambahBarang(Barang01 brg) {
31        if (!cekPenuh()) {
32            top++;
33            tumpukan[top] = brg;
34            System.out.println("Barang " + brg.nama + " berhasil ditambahkan ke Gudang");
35        } else {
36            System.out.println("Gagal! Tumpukan barang di gudang sudah penuh");
37        }
38    }
39
40    public Barang01 ambilBarang() {
41        if (!cekKosong()) {
42            Barang01 delete = tumpukan[top];
43            top--;
44            System.out.println("Barang " + delete.nama + " diambil dari Gudang.");
45            System.out.println("Kode unik dalam biner: " + konversiDesimalKeBiner(delete.kode));
46            return delete;
47        } else {
48            System.out.println("Tumpukan barang kosong.");
49            return null;
50        }
51    }
52 }
```

```
PS > J Gudang01.java > % Gudang01
public class Gudang01 {
    public Barang01 lihatBarangTeratas() {
        if (!cekKosong()) {
            Barang01 barangTeratas = tumpukan[top];
            System.out.println("Barang teratas : " + barangTeratas.nama);
            return barangTeratas;
        } else {
            System.out.println("Tumpukan barang kosong.");
            return null;
        }
    }
    public void tampilkanBarang() {
        if (!cekKosong()) {
            System.out.println("Rincian tumpukan barang di Gudang.");
            for (int i = top; i >= 0; i--) {
                System.out.printf(format:"Kode %d: %s (Kategori %s)\n", tumpukan[i].kode, tumpukan[i].nama, tumpukan[i].kategori);
            }
        } else {
            System.out.println("Tumpukan barang kosong.");
        }
    }
    public String konversiDesimalKeBiner (int kode) {
        StackKonversi01 stack = new StackKonversi01();
        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;
    }
}
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

```
ire > P8 > J StackKonversi01.java > StackKonversi01 > konversiDesimalKeBiner(int)
1 package P8;
2
3 public class StackKonversi01 {
4     int size;
5     int[] tumpukanBiner;
6     int top;
7
8     public StackKonversi01() {
9         this.size = 32;
10        tumpukanBiner = new int[size];
11        top = -1;
12    }
13
14    public boolean isEmpty() {
15        return top == -1;
16    }
17
18    public boolean isFull() {
19        return top == size - 1;
20    }
21
22    public void push(int data) {
23        if (isFull()) {
24            System.out.println("Stack penuh.");
25        } else {
26            top++;
27            tumpukanBiner[top] = data;
28        }
29    }
30
31    public int pop() {
32        if (isEmpty()) {
33            System.out.println("Stack kosong.");
34            return -1;
35        } else {
36            int data = tumpukanBiner[top];
37            top--;
38            return data;
39        }
40    }
41
42 }
```

```
41
42 public Barang01 ambilBarang() {
43     if (isEmpty()) {
44         int kode = pop();
45         String biner = konversiDesimalKeBiner(kode);
46         System.out.println("Barang dengan kode " + kode + " diambil dari Gudang.");
47         System.out.println("Kode unik dalam biner: " + biner);
48         return new Barang01(kode, nama: "Nama Barang", kategori: "Kategori Barang");
49     } else {
50         System.out.println("Tumpukan barang kosong.");
51         return null;
52     }
53 }
54
55 public String konversiDesimalKeBiner(int kode) {
56
57     StackKonversi01 stack = new StackKonversi01();
58     while (kode > 0) {
59         int sisa = kode % 2;
60         stack.push(sisa);
61         kode = kode / 2;
62     }
63     String biner = new String();
64     while (!stack.isEmpty()) {
65         biner += stack.pop();
66     }
67     return biner;
68 }
69 }
```

```
PS D:\college\semester 2\Algoritma dan Struktur Data_16_01> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\college\semester 2\Algoritma dan Struktur Data_16_01\bin' 'P8.Utama01'
```

```
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
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

Pertanyaan!

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

Jawab : proses konversi akan terjadi selama nilai `kode` tidak sama dengan 0. Setiap iterasi akan membagi nilai `kode` dengan 2 dan menyimpan sisa pembagiannya di stack untuk mendapatkan digit biner. Dan kode ini akan terus melakukan pembagian hingga nilai `kode` mencapai 0. Ketika nilai `kode` mencapai 0, tidak ada lagi digit yang perlu dikonversi, dan proses konversi selesai.

```
54  
55  
56  
57 {  
58     StackKonversi01 stack = new StackKonversi01();  
59     while (kode != 0) {  
60         int sisa = kode % 2;  
61         stack.push(sisa);  
62         kode = kode / 2;  
63     }  
64     String biner = new String();  
65     while (!stack.isEmpty()) {  
66         biner += stack.pop();  
67     }  
68     return biner;  
69 }
```

```
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Lihat barang teratas.  
5. Keluar  
Pilih Operasi :  
1  
Masukkan kode barang :  
22  
Masukkan nama barang :  
Pencil  
Masukkan nama kategori :  
Alat Masak  
Barang Pencil 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 Pencil diambil dari Gudang.  
Kode unik dalam biner: 10110
```

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Lihat barang teratas.  
5. Keluar  
Pilih Operasi :  
1  
Masukkan kode barang :  
0  
Masukkan nama barang :  
wijen  
Masukkan nama kategori :  
bahan makanan  
Barang wijen 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 wijen diambil dari Gudang.  
Kode unik dalam biner:
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

```
54  
55 public String konversiDesimalKeBiner(int kode) {  
56  
57     StackKonversi01 stack = new StackKonversi01();  
58     while (kode != 0) {  
59         int sisa = kode % 2;  
60         stack.push(sisa);  
61         kode = kode / 2;  
62     }  
63     String biner = new String();  
64     while (!stack.isEmpty()) {  
65         biner += stack.pop();  
66     }  
67     return biner;  
68 }  
69
```

Jawab : Pertama inialisasi StackKonversi01 untuk menyimpan digit biner sementara. Lalu baris kode while (kode != 0) melakukan konversi biner selama kode masih tidak sama dengan 0. Kemudian while berikutnya untuk mengambil digit biner dari stack dan return biner untuk memberikan hasil konversi biner dari nilai desimal yang diinputkan.

3. Praktikum 3

```
c> PB > Postfix01  
3 public class Postfix01 {  
4  
5     public int n;  
6     public int top;  
7     public char[] stack;  
8  
9     public Postfix01(int total) {  
10         n = total;  
11         top = -1;  
12         stack = new char[n];  
13         push('(');  
14     }  
15  
16     public void push(char c) {  
17         top++;  
18         stack[top] = c;  
19     }  
20  
21     public char pop() {  
22         char item = stack[top];  
23         top--;  
24         return item;  
25     }  
26  
27     public boolean IsOperand (char c) {  
28         if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || (c == '.' || c == ',')) {  
29             return true;  
30         } else {  
31             return false;  
32         }  
33     }  
34  
35     public boolean IsOperator (char c) {  
36         if (c == '*' || c == '%' || c == '/' || c == '** || c == '-' || c == '+') {  
37             return true;  
38         } else {  
39             return false;  
40         }  
41     }  
42 }
```




NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

```
c:\P8> Postfix > J Postfix01.java > StackKonversi01 > push(char)
3 public class Postfix01 {
4
5     public int derajat (char c) {
6         switch (c) {
7             case '^' :
8                 return 3;
9             case '%' :
10                 return 2;
11             case '/' :
12                 return 2;
13             case '*' :
14                 return 2;
15             case '-' :
16                 return 1;
17             case '+' :
18                 return 1;
19             default :
20                 return 0;
21         }
22     }
23
24     public String konversi (String Q) {
25         String P = "";
26         char c;
27         for (int i = 0; i < n; i++) {
28             c = Q.charAt(i);
29             if (IsOperand(c)) {
30                 P = P + c;
31             }
32             if (c == '(') {
33                 push (c);
34             }
35             if (c == ')') {
36                 while (stack[top] != '(') {
37                     P = P + pop ();
38                 }
39                 pop();
40             }
41             if (IsOperator(c)) {
42                 while (derajat(stack[top]) >= derajat(c)) {
43                     P = P + pop();
44                 }
45                 push (c);
46             }
47         }
48         return P;
49     }
50 }
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
```

```
Java J Gudang01.java J StackKonversi01.java J Postfix01.java J PostfixMain01.java 1 X J Utama01
src> P8> Postfix > J Utama01 > main(String[])
1 package P8.Postfix;
2
3 import java.util.Scanner;
4
5 public class PostfixMain01 {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         String P, Q;
10
11         System.out.println(x:"Masukkan ekspresi matematika (infix) : ");
12         Q = sc.nextLine();
13         Q = Q.trim();
14         Q = "(" + Q + ")";
15
16         int total = Q.length();
17
18         Postfix01 post = new Postfix01(total);
19         P = post.konversi(Q);
20         System.out.println("Postfix : " + P);
21     }
22 }
```

```
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01> & 'C:\Program Files\Java\jdk-20\bin\
ester 2\Algoritma_dan_Struktur_Data_1G_01\bin' 'P8.Postfix.PostfixMain01'
Masukkan ekspresi matematika (infix) :
a+b*(c+d-e)/f
Postfix : abcd+e-*f/+
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01> 
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

Pertanyaan!

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?

Jawab : Pada method `derajat`, terdapat beberapa case yang mengembalikan nilai yang sama karena dalam prioritas operator matematika yang diberikan, beberapa operator memiliki tingkat prioritas yang sama. Jika nilai yang dikembalikan diubah untuk setiap case sehingga setiap operator memiliki nilai prioritas yang berbeda-beda, maka hasilnya adalah akan mempengaruhi urutan evaluasi ekspresi matematika saat dikonversi ke postfix. Operator-operator dengan prioritas yang lebih tinggi akan dievaluasi lebih dulu dibandingkan dengan yang lainnya.

```
43 public int derajat (char c) {  
44     switch (c) {  
45         // pertanyaan 1 : merubah nilai prioritas tiap operator  
46         case '^' :  
47             return 4;  
48         case '%' :  
49             return 3;  
50         case '/' :  
51             return 3;  
52         case '*' :  
53             return 3;  
54         case '-' :  
55             return 2;  
56         case '+' :  
57             return 1;  
58         default :  
59             return 0;  
60     }  
61 }
```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS

'd:\college\semester 2\Algoritma dan Struktur Data 1G 01'; & 'C:\Program Files\ster 2\Algoritma dan Struktur Data 1G 01\bin' 'P8.Postfix.PostfixMain01'
Masukkan ekspresi matematika (infix) :
(a*b+c)/d-e^f
Postfix : ab*c+d/ef^~
PS D:\college\semester 2\Algoritma dan Struktur Data 1G 01>

2. Jelaskan alur kerja method konversi!

```
src > P8 > Postfix 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89  
62 public class Postfix01 {  
63     public String konversi (String Q) {  
64         String P = "";  
65         char c;  
66         for (int i = 0; i < Q.length(); i++) {  
67             c = Q.charAt(i);  
68             if (IsOperand(c)) {  
69                 P = P + c;  
70             }  
71             if (c == '(') {  
72                 push(c);  
73             }  
74             if (c == ')') {  
75                 while (stack[top] != '(') {  
76                     P = P + pop();  
77                 }  
78                 pop();  
79             }  
80             if (IsOperator(c)) {  
81                 while (derajat(stack[top]) >= derajat(c)) {  
82                     P = P + pop();  
83                 }  
84                 push(c);  
85             }  
86         }  
87         return P;  
88     }  
89 }
```

Jawab : Pertama deklarasi string `P` untuk menyimpan hasil konversi ekspresi infix ke postfix. Lalu, Iterasi melalui setiap karakter dalam string `Q`, yang merupakan ekspresi



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

matematika dalam bentuk infix. Kemudian setiap karakter diambil satu per satu dan diperiksa :

- Jika karakter adalah operand (bilangan atau variabel), maka karakter tersebut ditambahkan ke dalam string `P`.
- Jika karakter adalah kurung buka '(', maka karakter tersebut dimasukkan ke dalam stack.
- Jika karakter adalah kurung tutup ')', maka dilakukan pengosongan stack hingga menemukan kurung buka yang sesuai. Setiap karakter yang dikeluarkan dari stack dimasukkan ke dalam string `P`.
- Jika karakter adalah operator, maka dilakukan pengosongan stack sesuai dengan aturan prioritas operator. Operasi ini dilakukan hingga stack kosong atau ditemukan operator dengan prioritas lebih rendah.

Terakhir, setelah iterasi selesai, string `P` berisi ekspresi dalam bentuk postfix yang sesuai.

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

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

Jawab : Potongan kode `c = Q.charAt(i);` berfungsi untuk mengambil karakter pada posisi tertentu dalam string `Q` yang merupakan ekspresi matematika dalam bentuk infix.

LATIHAN PRAKTIKUM

Perhatikan dan gunakan kembali kode program pada Percobaan 1.

- a. Tambahkan dua method berikut pada class Gudang:
- Method lihatBarangTerbawah digunakan untuk mengecek barang pada tumpukan terbawah

```
3 public class Gudang01 {
63
64     public Barang01 lihatBarangTerbawah() {
65         if (!cekKosong()) {
66             Barang01 barangTerbawah = tumpukan[0];
67             System.out.println("Barang terbawah : " + barangTerbawah.nama);
68             return barangTerbawah;
69         } else {
70             System.out.println(x:"Tumpukan barang kosong.");
71             return null;
72         }
73     }
74     public Barang01 cariBarang(ObjekTumpukan o) {
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

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

```
public class Gudang01 {  
    public Barang01 cariBarang(Object cari) {  
        if (cari instanceof String) {  
            String namaBarang = (String) cari;  
            return cariBarangDenganNama(namaBarang);  
        } else if (cari instanceof Integer) {  
            int kodeBarang = (int) cari;  
            return cariBarangDenganKode(kodeBarang);  
        } else {  
            System.out.println(x: "Tidak valid");  
            return null;  
        }  
    }  
  
    public Barang01 cariBarangDenganNama(String namaBarang) {  
        if (isEmptyKosong()) {  
            for (int i = top; i >= 0; i--) {  
                if (tumpukan[i].nama.equals(namaBarang)) {  
                    System.out.println("Barang " + namaBarang + " ditemukan di Gudang.");  
                    return tumpukan[i];  
                }  
            }  
            System.out.println("Barang " + namaBarang + " tidak ditemukan di Gudang.");  
            return null;  
        } else {  
            System.out.println(x: "Tumpukan barang kosong.");  
            return null;  
        }  
    }  
  
    public Barang01 cariBarangDenganKode(int kodeBarang) {  
        if (isEmptyKosong()) {  
            for (int i = top; i >= 0; i--) {  
                if (tumpukan[i].kode == kodeBarang) {  
                    System.out.println("Barang dengan kode " + kodeBarang + " ditemukan di Gudang.");  
                    return tumpukan[i];  
                }  
            }  
            System.out.println("Barang dengan kode " + kodeBarang + " tidak ditemukan di Gudang.");  
            return null;  
        } else {  
            System.out.println(x: "Tumpukan barang kosong.");  
            return null;  
        }  
    }  
}
```

```
5 public class Utama01 {  
6     public static void main(String[] args) {  
7         System.out.println(x: "5. Lihat barang terbawah");  
8         System.out.println(x: "6. Cari barang berdasarkan kode");  
9         System.out.println(x: "7. Cari barang berdasarkan nama");  
10        System.out.println(x: "8. Keluar");  
11        System.out.println(x: "Pilih Operasi :");  
12        int pilihan = scanner.nextInt();  
13  
14        switch (pilihan) {  
15            case 1 :  
16                System.out.println(x: "Masukkan kode barang : ");  
17                int kode = scanner.nextInt();  
18                scanner.nextLine();  
19                System.out.println(x: "Masukkan nama barang : ");  
20                String nama = scanner.nextLine();  
21                System.out.println(x: "Masukkan nama kategori : ");  
22                String kategori = scanner.nextLine();  
23                Barang01 barangBaru = new Barang01(kode, nama, kategori);  
24                gudang.tambahBarang(barangBaru);  
25                break;  
26            case 2 :  
27                gudang.ambilBarang();  
28                break;  
29            case 3 :  
30                gudang.tampilkanBarang();  
31                break;  
32            case 4 :  
33                gudang.lihatBarangTeratas();  
34                break;  
35            case 5 :  
36                gudang.lihatBarangTerbawah();  
37                break;  
38            case 6 :  
39                scanner.nextLine();  
40                System.out.println(x: "Masukkan kode barang yang akan dicari : ");  
41                int kodeBarang = scanner.nextInt();  
42                gudang.cariBarangDenganKode(kodeBarang);  
43                break;  
44            case 7 :  
45                scanner.nextLine();  
46                System.out.println(x: "Masukkan nama barang yang akan dicari : ");  
47                String namaBarang = scanner.nextLine();  
48                gudang.cariBarangDenganNama(namaBarang);  
49                break;  
50        }  
51    }  
52 }
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : STACK (Jobsheet 7)

```
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\college\semester 2\Algoritma dan Struktur Data 1G 01> d; cd 'd:\college\semester 2\
eptionMessages' '-cp' 'D:\college\semester 2\Algoritma dan Struktur Data 1G 01\bin' 'P8.Bar
ta 1G 01\srcbin' 'P8.BarangGudang.Utama01' ;c1e04769-009d-48b8-90f9-398f1ccc4275ta 1G 01\src

1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas
5. Lihat barang terbawah
6. Cari barang berdasarkan kode
7. Cari barang berdasarkan nama
8. Keluar
Pilih Operasi :
1
Masukkan kode barang :
1
Masukkan nama barang :
Laptop
Masukkan nama kategori :
Elektronik
Barang Laptop 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 berdasarkan kode
7. Cari barang berdasarkan nama
8. Keluar
Pilih Operasi :
1
Masukkan kode barang :
2
Masukkan nama barang :
Setrika
Masukkan nama kategori :
Elektronik
Barang Setrika berhasil ditambahkan ke Gudang
```

```
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas
5. Lihat barang terbawah
6. Cari barang berdasarkan kode
7. Cari barang berdasarkan nama
8. Keluar
Pilih Operasi :
3
Rincian tumpukan barang di Gudang.
Kode 2: Setrika (Kategori Elektronik)
Kode 1: Laptop (Kategori Elektronik)

Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas
5. Lihat barang terbawah
6. Cari barang berdasarkan kode
7. Cari barang berdasarkan nama
8. Keluar
Pilih Operasi :
5
Barang terbawah : Laptop
```

```
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas
5. Lihat barang terbawah
6. Cari barang berdasarkan kode
7. Cari barang berdasarkan nama
8. Keluar
Pilih Operasi :
6
Masukkan kode barang yang akan dicari :
2
Barang dengan kode 2 ditemukan di Gudang.

Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas
5. Lihat barang terbawah
6. Cari barang berdasarkan kode
7. Cari barang berdasarkan nama
8. Keluar
Pilih Operasi :
7
Masukkan nama barang yang akan dicari :
Laptop
Barang Laptop ditemukan di Gudang.
```