



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

Percobaan 1 : Penyimpanan Tumpukan Barang dalam Gudang

```
1 package P8;  
2  
3 public class barang15 {  
4  
5     int kode;  
6     String nama;  
7     String kategori;  
8  
9     barang15(int kd, String nm, String kg) {  
10         kode = kd;  
11         nama = nm;  
12         kategori = kg;  
13     }  
14  
15 }
```

```
1 package P8;  
2  
3 public class gudang15 {  
4     barang15[] tumpukan;  
5     int size;  
6     int top;  
7  
8     gudang15(int kapasitas) {  
9         size = kapasitas;  
10        tumpukan = new barang15[kapasitas];  
11        top = -1;  
12    }  
13  
14    boolean cekKosong() {  
15        if (top == -1) {  
16            return true;  
17        } else {  
18            return false;  
19        }  
20    }  
21  
22    boolean cekPenuh() {  
23        if (top == size - 1) {  
24            return true;  
25        } else {  
26            return false;  
27        }  
28    }  
29  
30    void tambahBarang(barang15 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    barang15 ambilBarang() {  
41        if (!cekKosong()) {  
42            barang15 delete = tumpukan[top];  
43            top--;  
44            System.out.println("Barang " + delete.nama + " diambil dari Gudang.");  
45            return delete;  
46        } else {  
47            System.out.println("Tumpukan barang kosong.");  
48            return null;  
49        }  
50    }  
51  
52    barang15 lihatBarangTeratas() {  
53        if (!cekKosong()) {  
54            barang15 barangTeratas = tumpukan[top];  
55            System.out.println("Barang teratas: " + barangTeratas.nama);  
56            return barangTeratas;  
57        } else {  
58            System.out.println("Tumpukan barang kosong.");  
59            return null;  
60        }  
61    }  
62  
63    void tampilkanBarang() {  
64        if (!cekKosong()) {  
65            System.out.println("Rincian tumpukan barang di Gudang:");  
66            for (int i = top; i >= 0; i--) {  
67                System.out.printf("Kode %d: %s (Kategori %s)\n", tumpukan[i].kode, tumpukan[i].nama, tumpukan[i].kategori);  
68            }  
69        } else {  
70            System.out.println("Tumpukan barang kosong.");  
71        }  
72    }  
73 }  
74
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

```
1 package p8;  
2  
3 import java.util.Scanner;  
4  
5 public class utama15 {  
6  
7     static Scanner kiak = new Scanner(System.in);  
8     public static void main(String[] args) {  
9  
10         gudang15 gudang = new gudang15(7);  
11  
12         boolean kiaganteng = true;  
13  
14         while (kiaganteng) {  
15             System.out.println("Menu:");  
16             System.out.println("1. Tambah barang");  
17             System.out.println("2. Ambil barang");  
18             System.out.println("3. Tampilkan tumpukan barang");  
19             System.out.println("4. Keluar");  
20             System.out.print("Pilih operasi: ");  
21             int pilihan = kiak.nextInt();  
22             kiak.nextLine();  
23  
24             switch (pilihan) {  
25                 case 1:  
26                     System.out.print("Masukkan kode barang: ");  
27                     int kode = kiak.nextInt();  
28                     kiak.nextLine();  
29                     System.out.print("Masukkan nama barang: ");  
30                     String nama = kiak.nextLine();  
31                     System.out.print("Masukkan nama kategori: ");  
32                     String kategori = kiak.nextLine();  
33                     barang15 barangBaru = new barang15(kode, nama, kategori);  
34                     gudang.tambahBarang(barangBaru);  
35                     System.out.println();  
36                     break;  
37  
38                 case 2:  
39                     gudang.ambilBarang();  
40                     System.out.println();  
41                     break;  
42  
43                 case 3:  
44                     gudang.tampilkanBarang();  
45                     System.out.println();  
46                     break;  
47  
48                 case 4:  
49                     kiaganteng = false;  
50                     break;  
51  
52                 default:  
53                     System.out.println("Pilihan tidak valid. Silahkan coba lagi");  
54                     System.out.println();  
55                     continue;  
56             }  
57  
58         }  
59     }  
60 }  
61
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

```
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:
Kode 33: Pizza (Kategori Makanan)
Kode 21: Majalah (Kategori Buku)
```

```
Menu:
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih operasi: 4
PS D:\Kuliah\Semester 2\Praktek Algoritma Struktur Data\Algoritma_Struktur_Data_1G_15> |
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

Pertanyaan :

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

Jawab :

- a. Sebelumnya adalah !isempty

```
barang15 lihatBarangTeratas() {  
    if (!cekKosong()) {
```

- b. Pada bagian menu, ketika memilih 4 tidak langsung keluar, jadi ditambahkan kode berikut

```
boolean kiaganteng = true;
```

```
case 4:  
    kiaganteng = false;  
    break;
```

- c. Sebelumnya kode nya adalah `for (int i = 0; i <= top; i++)` , diubah menjadi seperti kode berikut

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

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

Jawab : Tujuh data barang

```
gudang15 gudang = new gudang15(kapasitas:7);
```

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

Jawab : Pemeriksaan kondisi tersebut berguna untuk memastikan gudang tidak kosong sebelum menampilkan barang-barang. Jika langkah ini dihilangkan, maka program bisa mengalami kesalahan saat mencoba menampilkan barang dari gudang yang kosong, dan bisa melakukan iterasi yang tidak diperlukan

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

Jawab :



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

```
1 package P8;
2
3 import java.util.Scanner;
4
5 public class utama15 {
6
7     static Scanner kiak = new Scanner(System.in);
8     public static void main(String[] args) {
9
10         System.out.print("Masukkan kapasitas gudang : ");
11         int kapasitas = kiak.nextInt();
12         gudang15 gudang = new gudang15(kapasitas);
13
14         boolean kiaganteng = true;
15
16         while (kiaganteng) {
17             System.out.println("Menu:");
18             System.out.println("1. Tambah barang");
19             System.out.println("2. Ambil barang");
20             System.out.println("3. Tampilkan tumpukan barang");
21             System.out.println("4. Lihat barang teratas");
22             System.out.println("5. Keluar");
23             System.out.print("Pilih operasi: ");
24             int pilihan = kiak.nextInt();
25             kiak.nextLine();
26
27             switch (pilihan) {
28                 case 1:
29                     System.out.print("Masukkan kode barang: ");
30                     int kode = kiak.nextInt();
31                     kiak.nextLine();
32                     System.out.print("Masukkan nama barang: ");
33                     String nama = kiak.nextLine();
34                     System.out.print("Masukkan nama kategori: ");
35                     String kategori = kiak.nextLine();
36                     barang15 barangBaru = new barang15(kode, nama, kategori);
37                     gudang.tambahBarang(barangBaru);
38                     System.out.println();
39                     break;
40
41                 case 2:
42                     gudang.ambilBarang();
43                     System.out.println();
44                     break;
45
46                 case 3:
47                     gudang.tampilkanBarang();
48                     System.out.println();
49                     break;
50
51                 case 4:
52                     gudang.lihatBarangTeratas();
53                     System.out.println();
54                     break;
55
56                 case 5:
57                     kiaganteng = false;
58                     break;
59
60                 default:
61                     System.out.println("Pilihan tidak valid. Silahkan coba lagi");
62                     System.out.println();
63                     continue;
64             }
65         }
66     }
67 }
68
69
```

5. Commit dan push kode program ke Github



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

Percobaan 2 : Konversi Kode Barang ke Biner

```
1 package p8;  
2  
3 public class stackKonversi15 {  
4     int size;  
5     int[] tumpukanBiner;  
6     int top;  
7  
8     stackKonversi15() {  
9         this.size = 32; //asumsi 32 bit  
10        tumpukanBiner = new int[size];  
11        top = -1;  
12    }  
13  
14    boolean isEmpty() {  
15        return top == -1;  
16    }  
17  
18    boolean isFull() {  
19        return top == size - 1;  
20    }  
21  
22    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    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 }  
43
```

```
barang15 ambilBarang() {  
    if (!cekKosong()) {  
        barang15 delete = tumpukan[top];  
        top--;  
        System.out.println("Barang " + delete.nama + " diambil dari Gudang.");  
        System.out.println("Kode unik dalam biner: " + konversiDesimalKeBiner(delete.kode));  
        return delete;  
    } else {  
        System.out.println(x:"Tumpukan barang kosong.");  
        return null;  
    }  
}
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

```
Masukkan kapasitas gudang : 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: Sertrika
Masukkan nama kategori: Elektronik
Barang Sertrika 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 Sertrika 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:\Kuliah\Semester 2\Praktek Algoritma Struktur Data\Algoritma_Struktur_Data_1G_15>
```

Pertanyaan :

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

Jawab : Hasilnya akan tetap sama, akan tetapi yang membedakan adalah, jika (kode != 0) masih akan mengenal angka negatif, sedangkan (kode > 0) tidak mengenal angka negatif.

```
String konversiDesimalKeBiner(int kode) {
    stackKonversi15 stack = new stackKonversi15();
    while (kode != 0) {
        int sisa = kode % 2;
        stack.push(sisa);
        kode = kode / 2;
    }
}
```

```
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 : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

Jawab : Method tersebut mengkonversi bilangan desimal menjadi representasi biner, dengan cara membagi secara terus menerus nilai desimal dengan 2 dalam loop, kemudian sisa pembagian disimpan di stack, selanjutnya diambil dari stack dan disusun menjadi biner.

Percobaan 3: Konversi Notasi Infix ke Postfix

```
1 package PB;  
2  
3 public class postfix15 {  
4  
5     int n;  
6     int top;  
7     char[] stack;  
8  
9     public postfix15(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  
22    public char pop() {  
23        char item = stack[top];  
24        top--;  
25        return item;  
26    }  
27  
28    public boolean isOperand(char c) {  
29        if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || c == '.' || c == ',') {  
30            return true;  
31        } else {  
32            return false;  
33        }  
34    }  
35  
36    public boolean isOperator(char c) {  
37        if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {  
38            return true;  
39        } else {  
40            return false;  
41        }  
42    }  
43  
44    public int derajat(char c) {  
45        switch (c) {  
46            case '^':  
47                return 3;  
48  
49            case '%':  
50                return 2;  
51  
52            case '/':  
53                return 2;  
54  
55            case '*':  
56                return 2;  
57  
58            case '-':  
59                return 1;  
60  
61            case '+':  
62                return 1;  
63  
64            default:  
65                return 0;  
66        }  
67    }  
68  
69    public String konversi(String Q) {  
70        String P = "";  
71        char c;  
72        for (int i = 0; i < n; i++) {  
73            c = Q.charAt(i);  
74            if (isOperand(c)) {  
75                P = P + c;  
76            }  
77            if (c == '(') {  
78                push(c);  
79            }  
80            if (c == ')') {  
81                while (stack[top] != '(') {  
82                    P = P + pop();  
83                }  
84                pop();  
85            }  
86            if (isOperator(c)) {  
87                while (derajat(stack[top]) >= derajat(c)) {  
88                    P = P + pop();  
89                }  
90                push(c);  
91            }  
92        }  
93        return P;  
94    }  
95 }  
96  
97 }  
98  
99 }  
100  
101 }  
102
```




NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : STACK

```
1 package p8;  
2  
3 import java.util.Scanner;  
4  
5 public class postfixmain15 {  
6     public static void main(String[] args) {  
7  
8         Scanner kiak = new Scanner(System.in);  
9         String P, Q;  
10        System.out.println("Masukkan ekspresi matematika (infix) : ");  
11        Q = kiak.nextLine();  
12        Q = Q.trim();  
13        Q = Q + " ";  
14  
15        int total = Q.length();  
16        postfix15 post = new postfix15(total);  
17        P = post.konversi(Q);  
18        System.out.println("Postfix: " + P);  
19  
20        kiak.close();  
21    }  
22 }  
23
```

```
Masukkan ekspresi matematika (infix) :  
a+b*(c+d-e)/f  
Postfix: abcd+e-*f/+
```

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, beberapa case memiliki nilai return yang sama karena operator-operator tersebut memiliki tingkat prioritas yang setara dalam ekspresi matematika. Jika return value diubah dengan nilai berbeda-beda setiap case-nya, maka akan menunjukkan prioritas operasi yang berbeda

2. Jelaskan alur kerja method konversi!

Jawab : Program ini mengonversi ekspresi infix ke postfix menggunakan stack, menambahkan operand langsung ke output, menangani tanda kurung dengan memasukkan ke stack dan mengeluarkan dari stack saat menemukan tanda kurung yang sesuai, serta menangani operator dengan membandingkan prioritasnya untuk memindahkannya ke output.

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

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

Jawab : Potongan kode tersebut berfungsi untuk mengambil karakter dari string Q pada posisi indeks ke-i dan menetapkan karakter tersebut ke dalam variabel c. Dengan kata lain, c akan menyimpan karakter pada posisi indeks ke-i dari string Q saat iterasi dilakukan melalui loop.