



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

FILE NAME =ABSEN_NAME _CLASS_P1

* Pertemuan mengikuti pertemuan ke berapa

2. 1. Percobaan 1: Penyimpanan Tumpukan Barang dalam Gudang

* screenshot code program hasil percobaan*

Jika percobaan membutuhkan lebih dari 1 class java, maka screenshot dilakukan per class / per method

Class Gudang();

```
J Barang26.java U PrakASD_1F_26\src\P8\Barang26.java\...
1  package P8;
2
3  public class Barang26 {
4      int kode;
5      String nama;
6      String kategori;
7
8      public Barang26(int kode, String nama, String kategori) {
9          this.kode = kode;
10         this.nama = nama;
11         this.kategori = kategori;
12     }
13 }
```

Class Barang();

```
J Gudang26.java U PrakASD_1F_26\src\P8\Gudang26.java\ Gudang26\ tampilkanBarang()
1  package P8;
2
3  public class Gudang26 {
4      Barang26[] tumpukan;
5      int size;
6      int top;
7
8      public Gudang26(int kapasitas) {
9          size = kapasitas;
10         tumpukan = new Barang26[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 }
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

```
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(Barang26 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 sidah penuh");
37         }
38     }
39
40     public Barang26 ambilBarang() {
41         if (!cekKosong()) {
42             Barang26 delete = tumpukan[top];
43             top--;
44             System.out.println("Barang " + delete.nama + " diambil dari Gudang");
45             return delete;
46         } else {
47             System.out.println(x:"Tumpukan barang Kosong!");
48             return null;
49         }
50     }
51
52     public Barang26 lihatBarangTeratas() {
53         if (!cekKosong()) {
54             Barang26 barangTeratas = tumpukan[top];
55             System.out.println("Barang Teratas : " + barangTeratas.nama);
56             return barangTeratas;
57         } else {
58             System.out.println(x:"Tumpukan barang Kosong!");
59             return null;
60         }
61     }
62 }
```

```
63     public void tampilkanBarang() {
64         if (!cekKosong()) {
65             System.out.println(x:"Rincian Tumpukan Barang di Gudang : ");
66             for (int i = 0; i <= top; i++) {
67                 System.out.printf(format:"Kode %d: %s (Kategori %s)\n", tumpukan[i].kode,
68                                     tumpukan[i].kategori);
69             }
70         } else {
71             System.out.println(x:"Tumpukan barang Kosong!");
72         }
73     }
74 }
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

Class Utama();

```
J Utama26.java 1, U PrakASD_1F_26\src\P8\Utama26.java\...
1  package P8;
2
3  import java.util.Scanner;
4
5  public class Utama26 {
6      Run | Debug
7      public static void main(String[] args) {
8          Gudang26 gudang = new Gudang26(kapasitas:7);
9          Scanner sc26 = 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.print(s:"Pilih opsi : ");
18             int pilihan = sc26.nextInt();
19             sc26.nextLine();
20
21             switch (pilihan) {
22                 case 1:
23                     System.out.print(s:"Masukkan Kode Barang      : ");
24                     int kode = sc26.nextInt();
25                     sc26.nextLine();
26                     System.out.print(s:"Masukkan Nama Barang      : ");
27                     String nama = sc26.nextLine();
28                     System.out.print(s:"Masukkan Kategori Barang : ");
29                     String kategori = sc26.nextLine();
30                     Barang26 barangBaru = new Barang26(kode, nama, kategori);
31                     gudang.tambahBarang(barangBaru);
32                     break;
33                 case 2:
34                     gudang.ambilBarang();
35                     break;
36                 case 3:
37                     gudang.tampilkanBarang();
38                     break;
39                 case 4:
40                     continue;
41                 default:
42                     System.out.println(x:"Pilihan tidak Valid. Silahkan coba lagi!");
43             }
44         }
45     }
46 }
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
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?

Jawab :

Perubahan terjadi pada perulangan :

```
63 public void tampilkanBarang() {  
64     if (!cekKosong()) {  
65         System.out.println(x:"Rincian Tumpukan Barang di Gudang : ");  
66         for (int i = top; i >= 0; i--) {  
67             System.out.printf(format:"Kode %d: %s (Kategori %s)\n", tumpukan[i].kode, tumpukan[i].nama,  
68                 tumpukan[i].kategori);  
69         }  
70     } else {  
71         System.out.println(x:"Tumpukan barang Kosong!");  
72     }  
73 }
```

Output setelah perbaikan :

```
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Keluar  
Pilih opsi : 1  
Masukkan Kode Barang : 21  
Masukkan Nama Barang : Majalah  
Masukkan Kategori Barang : Buku  
Barang Majalah Berhasil ditambahkan ke Gudang  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Keluar  
Pilih opsi : 1  
Masukkan Kode Barang : 26  
Masukkan Nama Barang : Jaket  
Masukkan Kategori Barang : Pakaian  
Barang Jaket Berhasil ditambahkan ke Gudang  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Keluar  
Pilih opsi : 2  
Barang Jaket diambil dari Gudang  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Keluar  
Pilih opsi : 1  
Masukkan Kode Barang : 33  
Masukkan Nama Barang : Pizza  
Masukkan Kategori Barang : Makanan  
Barang Pizza Berhasil ditambahkan ke Gudang
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

```
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Keluar  
Pilih opsi : 3  
Rincian Tumpukan Barang di Gudang :  
Kode 33: Pizza (Kategori Makanan)  
Kode 21: Majalah (Kategori Buku)
```

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

Jawab : Maksimal data yaitu 7

```
7          Gudang26 gudang = new Gudang26(kapasitas:7);
```

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

Jawab : Fungsi cekKosong() untuk memastikan bahwa sebelum menampilkan daftar barang, gudang tidak kosong.

jika kondisi !cekKosong() dihapus, maka method tampilkanBarang() akan tetap mencoba untuk menampilkan daftar barang. Dampaknya adalah program akan menampilkan pesan bahwa tumpukan barang kosong.

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

Jawab :

Modifikasi Program pada Class Utama() :

```
6      public static void main(String[] args) {  
7          Scanner sc26 = new Scanner(System.in);  
8          System.out.print(s:"Masukkan Kapasitas Gudang : ");  
9          int kapasitas = sc26.nextInt();  
10         Gudang26 gudang = new Gudang26(kapasitas);  
  
39         case 4:  
40             gudang.lihatBarangTeratas();  
41             break;
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

Outputnya :

```
Masukkan Kapasitas Gudang : 2

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan tumpukan Barang
4. Tampilkan Barang Teratas
5. Keluar
Pilih opsi : 1
Masukkan Kode Barang : 21
Masukkan Nama Barang : Majalah
Masukkan Kategori Barang : Buku
Barang Majalah Berhasil ditambahkan ke Gudang

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan tumpukan Barang
4. Tampilkan Barang Teratas
5. Keluar
Pilih opsi : 1
Masukkan Kode Barang : 26
Masukkan Nama Barang : Jaket
Masukkan Kategori Barang : Pakaian
Barang Jaket Berhasil ditambahkan ke Gudang

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan tumpukan Barang
4. Tampilkan Barang Teratas
5. Keluar
Pilih opsi : 4
Barang Teratas : Jaket
```

5. Commit dan push kode program ke Github

Jawab :

https://github.com/Satriowish/prakASD_1F_26/tree/main/PrakASD_1F_26/src/P8



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

2.2 Percobaan 2: Konversi Kode Barang ke Biner

** screenshot code program hasil percobaan**

Jika percobaan membutuhkan lebih dari 1 class java, maka screenshot dilakukan per class / per method

Class Baru StackKonversi26 :

```
J StackKonversi26.java U, U PrakASD_1F_26\src\P8\StackKonversi26.java\...
1  package P8;
2
3  public class StackKonversi26 {
4      int size;
5      int[] tumpukanBiner;
6      int top;
7
8      public StackKonversi26() {
9          this.size = 32; // asumsi 32 bit
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(x: "Stack Penuh ");
25         } else {
26             top++;
27             tumpukanBiner[top] = data;
28         }
29     }
30
31     public int pop() {
32         if (isEmpty()) {
33             System.out.println(x: "Stack Kosong ");
34             return -1;
35         } else {
36             int data = tumpukanBiner[top];
37             top--;
38             return data;
39         }
40     }
41 }
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

Penambahan Method dan perubahan pada class Gudang :

```
40 public Barang26 ambilBarang() {  
41     if (!cekKosong()) {  
42         Barang26 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(x:"Tumpukan barang Kosong!");  
49         return null;  
50     }  
51 }
```

```
76 public String konversiDesimalKeBiner(int kode) {  
77     StackKonversi26 stack = new StackKonversi26();  
78     while (kode > 0) {  
79         int sisa = kode % 2;  
80         stack.push(sisa);  
81         kode = kode / 2;  
82     }  
83     String biner = new String();  
84     while (!stack.isEmpty()) {  
85         biner += stack.pop();  
86     }  
87     return biner;  
88 }
```

Output :

```
Masukkan Kapasitas Gudang : 1  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Tampilkan Barang Teratas  
5. Keluar  
Pilih opsi : 1  
Masukkan Kode Barang : 13  
Masukkan Nama Barang : Setrika  
Masukkan Kategori Barang : Elektronik  
Barang Setrika Berhasil ditambahkan ke Gudang  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Tampilkan Barang Teratas  
5. Keluar  
Pilih opsi : 2  
Barang Setrika diambil dari Gudang  
Kode unik dalam Biner : 1101
```




NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

Question :

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

Jawab : akan Menghasilkan hasil yang sama dengan sebelumnya, yaitu 1101. karena konversi desimal ke biner memerlukan proses pengulangan sampai sisa pembagian desimal menjadi 0. Ketika kondisi perulangan menggunakan while (kode $\neq 0$), proses konversi akan terus berlanjut sampai kode menjadi nol, yang menunjukkan bahwa semua digit telah dikonversi.

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

Jawab :

1. **Pembentukan Stack** : Pada objek StackKonversi26 dibuat untuk menyimpan sisa hasil bagi dari operasi pembagian bilangan desimal dengan 2.
2. **Iterasi Perhitungan** : Kemudian, dilakukan iterasi selama nilai desimal yang akan dikonversi masih lebih besar dari 0. Didalamnya terdapat Langkah berikut :
Menghitung sisa pembagian bilangan desimal dengan 2.
Menambahkan sisa tersebut ke dalam stack.
Membagi bilangan desimal dengan 2.
3. **Pengambilan dari Stack** : Setelah iterasi selesai, langkah berikutnya adalah mengambil angka-angka biner dari stack secara berurutan.dilakukan menggunakan operasi pop() dari stack. Setiap angka yang diambil akan menjadi biner dari bilangan desimal.
4. **Penggabungan Angka** : Angka-angka biner yang sudah diambil dari stack kemudian digabungkan menjadi satu string biner.
5. **Return Hasil** : String biner yang terbentuk akan menjadi hasilnya kemudian akan dikembalikan ke pemanggil method.



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

2.2 Percobaan 3: Konversi Notasi Infix ke Postfix

** screenshot code program hasil percobaan**

Jika percobaan membutuhkan lebih dari 1 class java, maka screenshot dilakukan per class / per method

Class Postfix() :

```
J Posfix26.java U, U PrakASD_1F_20\src\P8\Posfix26.java Posfix26
1  package P8;
2
3  public class Posfix26 {
4      int n;
5      int top;
6      char[] stack;
7
8      public Posfix26(int total) {
9          n = total;
10         top = -1;
11         stack = new char[n];
12         push('(');
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 isOperand(char c) {
27         if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || c == '.' || c == '-') {
28             return true;
29         } else {
30             return false;
31         }
32     }
33
34     public boolean isOperator(char c) {
35         if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {
36             return true;
37         } else {
38             return false;
39         }
40     }
41 }
```

```
41
42     public int derajat(char c) {
43         switch (c) {
44             case '^':
45                 return 3;
46             case '%':
47                 return 2;
48             case '/':
49                 return 2;
50             case '*':
51                 return 2;
52             case '-':
53                 return 1;
54             case '+':
55                 return 1;
56             default:
57                 return 0;
58         }
59     }
60 }
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

```
60
61     public String konversi(String Q) {
62         String P = "";
63         char c;
64         for (int i = 0; i < n; i++) {
65             c = Q.charAt(i);
66             if (isOperand(c)) {
67                 P = P + c;
68             }
69             if (c == '(') {
70                 push(c);
71             }
72             if (c == ')') {
73                 while (stack[top] != '(') {
74                     P = P + pop();
75                 }
76                 pop();
77             }
78             if (isOperator(c)) {
79                 while (derajat(stack[top]) >= derajat(c)) {
80                     P = P + pop();
81                 }
82                 push(c);
83             }
84         }
85         return P;
86     }
87 }
```

Class PostfixMain() :

```
J PostfixMain.java 1, U, U PrakASD_1F_26\src\P8\PostfixMain.java\...
1     package P8;
2
3     import java.util.Scanner;
4
5     public class PostfixMain {
6         Run | Debug
7         public static void main(String[] args) {
8             Scanner sc = new Scanner(System.in);
9             String P, Q;
10            System.out.println("Masukkan ekspresi matematika (infix) : ");
11            Q = sc.nextLine();
12            Q = Q.trim();
13            Q = Q + ")";
14
15            int total = Q.length();
16            Posfix26 post = new Posfix26(total);
17            P = post.konversi(Q);
18            System.out.println("Postfix : " + P);
19        }
20    }
```

Output :

```
Masukkan ekspresi matematika (infix) :
a+b*(c+d-e)/f
Postfix : abcd+e-*f/+
PS D:\Praktikum Algoritma dan Struktur Data> □
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

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?

Jawab : karena beberapa operator memiliki tingkat prioritas yang sama dalam ekspresi matematika. Seperti operator * yang sama dengan operator / dan % . operator - memiliki nilai sama dengan operator +. Jika return value diubah dengan nilai berbeda-beda setiap case-nya maka akan memengaruhi prioritas operator. Hal ini akan mengakibatkan perubahan dari urutan operator yang dipindahkan dari stack ke dalam ekspresi postfix, sehingga memengaruhi hasil akhir dari konversi infix ke postfix.

2. Jelaskan alur kerja method konversi!

Jawab :

- Method menerima ekspresi infix Q sebagai input.
 - Sebuah string kosong P dibuat sebagai output yang akan berisi ekspresi postfix.
 - Setiap karakter dalam ekspresi infix Q diperiksa satu per satu.
 - Jika karakter merupakan operand, maka langsung dimasukkan ke dalam string P.
 - Jika karakter adalah kurung buka (, maka kurung tersebut dimasukkan ke dalam stack.
 - Jika karakter adalah kurung tutup), maka karakter-karakter dari stack akan dimasukkan ke dalam string P hingga menemukan kurung buka yang sesuai.
 - Jika karakter adalah operator, maka akan dilakukan pengecekan prioritas operator pada stack. Operator-operator pada stack yang memiliki prioritas lebih tinggi atau sama dengan operator saat ini akan dipindahkan ke dalam string P, kemudian operator saat ini dimasukkan ke dalam stack.
 - Setelah semua karakter dalam ekspresi infix diperiksa, karakter-karakter yang tersisa di dalam stack (jika ada) akan dimasukkan ke dalam string P.
3. Pada method konversi, apa fungsi dari potongan kode berikut?

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

Jawab : untuk mengambil atau memotong karakter pada indeks i dari string Q (ekspresi infix) dan menyimpannya dalam variabel c. Ini dilakukan untuk memeriksa setiap karakter dari ekspresi infix satu per satu saat melakukan konversi ke dalam ekspresi postfix.



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

2.4 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

Penambahan Method Pada class Gudang26(); :

```
90     public Barang26 lihatBarangTerbawah() {  
91         if (!cekKosong()) {  
92             Barang26 barangTerbawah = tumpukan[0];  
93             System.out.println("Barang Terbawah : " + barangTerbawah.nama);  
94             return barangTerbawah;  
95         } else {  
96             System.out.println(x:"Tumpukan barang Kosong!");  
97             return null;  
98         }  
99     }
```

Pemanggilan Method ke Class Utama(); :

```
12     while (true) {  
13         System.out.println(x:"\nMenu :");  
14         System.out.println(x:"1. Tambah Barang");  
15         System.out.println(x:"2. Ambil Barang");  
16         System.out.println(x:"3. Tampilkan tumpukan Barang");  
17         System.out.println(x:"4. Tampilkan Barang Teratas");  
18         System.out.println(x:"5. Tampilkan Barang Terbawah");  
19         System.out.println(x:"6. Keluar");  
20         System.out.print(s:"Pilih opsi : ");  
21         int pilihan = sc26.nextInt();  
22         sc26.nextLine();
```

```
45         case 5:  
46             gudang.lihatBarangTerbawah();  
47             break;
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

Outputnya :

```
Masukkan Kapasitas Gudang : 2

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan tumpukan Barang
4. Tampilkan Barang Teratas
5. Tampilkan Barang Terbawah
6. Keluar
Pilih opsi : 1
Masukkan Kode Barang : 13
Masukkan Nama Barang : Setrika
Masukkan Kategori Barang : Elektronik
Barang Setrika Berhasil ditambahkan ke Gudang

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan tumpukan Barang
4. Tampilkan Barang Teratas
5. Tampilkan Barang Terbawah
6. Keluar
Pilih opsi : 1
Masukkan Kode Barang : 26
Masukkan Nama Barang : Jaket
Masukkan Kategori Barang : Pakaian
Barang Jaket Berhasil ditambahkan ke Gudang

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan tumpukan Barang
4. Tampilkan Barang Teratas
5. Tampilkan Barang Terbawah
6. Keluar
Pilih opsi : 5
Barang Terbawah : Setrika
```

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

Penambahan Method Pada class Gudang26(); :

```
101     public int cariBarangbyNama(String cari) {
102         for (int i = 0; i <= top; i++) {
103             if (tumpukan[i].nama.equalsIgnoreCase(cari)) {
104                 System.out.println("Barang dengan nama '" + cari + "' ditemukan pada indeks " + i);
105                 return i;
106             }
107         }
108         System.out.println("Barang dengan nama '" + cari + "' tidak ditemukan.");
109         return -1;
110     }
```



NAMA : SATRIO WISNU ADI PRATAMA
NIM : 2341720219
NO ABSEN : 26
KELAS : 1F
MATERI : Stack

Pemanggilan Method ke Class Utama(); :

```
12         while (true) {  
13             System.out.println(x:"\nMenu :");  
14             System.out.println(x:"1. Tambah Barang");  
15             System.out.println(x:"2. Ambil Barang");  
16             System.out.println(x:"3. Tampilkan tumpukan Barang");  
17             System.out.println(x:"4. Tampilkan Barang Teratas");  
18             System.out.println(x:"5. Tampilkan Barang Terbawah");  
19             System.out.println(x:"6. Cari Barang berdasarkan Nama");  
20             System.out.println(x:"7. Keluar");  
21             System.out.print(s:"Pilih opsi : ");  
22             int pilihan = sc26.nextInt();  
23             sc26.nextLine();
```

```
49         case 6:  
50             System.out.print(s:"Masukkan Nama Barang yang ingin dicari : ");  
51             String namaBarangCari = sc26.nextLine();  
52             gudang.cariBarangbyNama(namaBarangCari);
```

Outputnya :

```
Masukkan Kapasitas Gudang : 2  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Tampilkan Barang Teratas  
5. Tampilkan Barang Terbawah  
6. Cari Barang berdasarkan Nama  
7. Keluar  
Pilih opsi : 1  
Masukkan Kode Barang : 13  
Masukkan Nama Barang : Setrika  
Masukkan Kategori Barang : Elektronik  
Barang Setrika Berhasil ditambahkan ke Gudang  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Tampilkan Barang Teratas  
5. Tampilkan Barang Terbawah  
6. Cari Barang berdasarkan Nama  
7. Keluar  
Pilih opsi : 1  
Masukkan Kode Barang : 26  
Masukkan Nama Barang : Jaket  
Masukkan Kategori Barang : Pakaian  
Barang Jaket Berhasil ditambahkan ke Gudang  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan tumpukan Barang  
4. Tampilkan Barang Teratas  
5. Tampilkan Barang Terbawah  
6. Cari Barang berdasarkan Nama  
7. Keluar  
Pilih opsi : 6  
Masukkan Nama Barang yang ingin dicari : Jaket  
Barang dengan nama 'Jaket' ditemukan pada indeks 1  
PS D:\Praktikum Algoritma dan Struktur Data>
```