

1. Praktikum 1 : Penyimpanan barang dalam gudang.

```
/iew
     Go
                   \leftarrow \rightarrow

∠ Algoritma_dan_Struktur_Data_1G_01

 J Barang01.java U X J Gudang01.java U J Utama01.java 1, U

J PencarianBankAdani

 package P8;
       public class Barang01 {
           int kode;
           String nama;
           String kategori;
           Barang01(int kode, String nama, String kategori) {
           this.kode = kode;
           this.nama = nama;
           this.kategori = kategori;
```

```
on View Go ···
public class Gudang01 {
        Barang01[] tumpukan;
         public Gudang01(int kapasitas) {
           size = kapasitas;
            tumpukan = new Barang01 [size];
            top = -1;
         public boolean cekKosong() {
         public boolean cekPenuh() {
           if (top == size -1) {
         public void tambahBarang(Barang01 brg) {
            if (!cekPenuh()) {
               top++;
               tumpukan[top] = brg;
System.out.println("Barang " + brg.nama + " berhasil ditambahkan ke Gudang");
               System.out.println(x:"Gagal! Tumpukan barang di gudang sudah penuh");
```



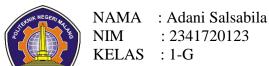
KELAS: 1-G

```
| Note | Section | Section
```

```
### Algoritma_dan_Struktur_Data_1G_01

### Algoritma_dan_Struktur_Data_Id_01

### Algoritma_dan_
```



MATERI: STACK (Jobsheet 7)

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\college\semester \( \frac{1}{2} \) \( \frac{1} \) \( \frac{1} \) \( \frac{1}{2} \) \( \frac{1}{2
      Menu :
       1. Tambah barang
       2. Ambil barang
       3. Tampilkan tumpukan barang
       4. Keluar
      Pilih Operasi :
       Masukkan kode barang :
       Masukkan nama barang :
       Majalah
       Masukkan nama kategori :
       Barang Majalah berhasil ditambahkan ke Gudang
       1. Tambah barang
       2. Ambil barang
       3. Tampilkan tumpukan barang
       4. Keluar
      Pilih Operasi :
       Masukkan kode barang:
       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 :
Masukkan kode barang:
Masukkan nama barang :
Seblak
Masukkan nama kategori :
Barang Seblak berhasil ditambahkan ke Gudang
Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Keluar
Pilih Operasi :
Rincian tumpukan barang di Gudang.
Kode 33: Seblak (Kategori Makanan)
Kode 21: Majalah (Kategori Buku)
```

Pertanyaan!



KELAS: 1-G

MATERI: STACK (Jobsheet 7)

 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.printle(sit...tender berengt);
    System.out.printle(sit...tender berengt);
    System.out.printle(sit...tender berengt);
    System.out.printle(sit...tender berengt);
    System.out.printle(sit...tender berengt berengt);
    System.out.printle(sit...tender berengt treates...);
    System.out.printle(sit...tender);
    System.out.printle(sit...tender berengt in);
    interest berengt berengt berengt in);
    interest berengt berengt berengt in);
    interest berengt berengt berengt in);
    System.out.printle(sit...tender berengt in);
    breat;
    case in general identification ();
    breat;
    case in general identification ();
    breat;
    System.out.printle(sit...tender in);
    System.out.printle(sit...tender in);
    System.out.printle(sit...tender in);
    System.out.printle(sit...tender in);
    System.out.printle(sit.
```

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE 1

PS D:\college\semester 2\Algoritma_dan_S
Program Files\Java\jdk-20\bin\java.exe'
ptionMessages' '-cp' 'D:\college\semester
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.
```



KELAS: 1-G

MATERI: STACK (Jobsheet 7)

2. Praktikum 2 : Konversi Kode Barang ke Biner.



KELAS: 1-G

```
sr > 98 > J SackKonversiDjava > $ SackKonversiDl > @ konversiDesimalKeBiner(int)

package P8;

public class StackKonversiDl {
    int size;
    int[] tumpukanBiner;
    int top;

public StackKonversiDl() {
        tinis.size = 32;
        tumpukanBiner = new int[size];
        top = -1;
    }

public boolean isEmpty() {
        return top == -1;
    }

public boolean isEmpty() {
        return top == -1;
    }

public void push(int data) {
        return top -= size - 1;
    }

public void push(int data) {
        if (isFull()) {
            System.out.println(xi*Stack penuh.*);
        } else {
            top+;
            tumpukanBiner[top] = data;
        }

    public int pop() {
        if (isfampty()) {
            System.out.println(xi*Stack kosong,*);
            return -1;
        } else {
            int data = tumpukanBiner[top];
            top-;
            return data;
    }

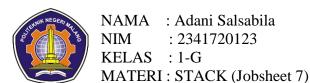
}
```

```
public BarangDl ambilBarang() (

if (litEmby()) (

if (litEmby())
```

```
O PS D:\college\semester 2\Algoritma_dan_Struktur_Data_16_gl> & 'C:\
Program Files\Java\jdx-20\\inin\java.exe' '-XX:*ShowCodeDetailsInExce
ption\text{Program Files\Java.exe Program Files\Java.exe Program
```



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.

```
public String konversiDesimalKeBiner(int kode) {

56

57

StackKonversiOl stack = new StackKonversiOl();

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;

69

}
```

```
Social Street out printle (VOTTUMENDIAN Assembly Corons 1).

PROBLEMS (18) OUTPUT DEBUG CONSOLE TERMINAL PORTS

2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas.
5. Keluar
Pilih Operasi:
1
Rasukkan kode barang:
22
Panci
Pasukkan nama barang:
Panci Pasukkan nama kategori:
Alat Masak
Barang Panci berhasil ditambahkan ke Gudang

Penu:
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas.
5. Keluar
Pilih Operasi:
2
Pilih Operasi:
2
Panci 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
Mesukkan kade barang:
8
Mesukkan nama barang:
9
Mesukkan nama barang:
wijen
Mesukkan 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 binen:
```



KELAS: 1-G

MATERI: STACK (Jobsheet 7)

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

```
public String konversiDesimalKeBiner(int kode) {

56

57

StackKonversiOl stack = new StackKonversiOl();

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;

69

}
```

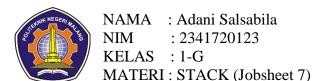
Jawab : Pertama inisialisasi StackKonversi01 untuk menyimpan digit biner sementara. Lalu baris kode while (kode != 0) melakukan konversi biner selama kode masih tidak sama dengan 0. Kemudian while berikutnya untuk mengambil digit biner dari stack dan return biner untuk memberikan hasil konversi biner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

3. Praktikum 3



KELAS: 1-G

```
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> []
```



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.

```
public int derajat (char c) {
    switch (c) {
        // pertanyaan 1 : merubah nilai prioritas tiap operator
        case '^:
        return 4;
        case '%':
        return 3;
        case '/':
        return 3;
        case '':
        return 3;
        case '-':
        return 3;
        case '-':
        return 2;
        case '-':
        return 1;
        default :
        return 0;

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS

'd:\college\semester 2\Algoritma_dan_Struktur_Data_16_01'; & 'C:\Program Files\
ster 2\Algoritma_dan_Struktur_Data_16_01'; & 'C:\Program Files\
Masukkan ekspresi matematika (infix) :
        (a*b+c)/d-e^f
Postfix : ab*c+d/ef^-
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_16_01>
```

2. Jelaskan alur kerja method konversi!

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



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

```
public class Gudang01 {

public Barang01 lihatBarangTerbawah() {

if (!cekKosong()) {

Barang01 barangTerbawah = tumpukan[0];

System.out.println("Barang terbawah : " + barangTerbawah.nama);

return barangTerbawah;

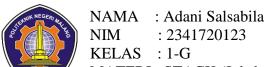
} else {

System.out.println(x:"Tumpukan barang kosong.");

return null;

}

}
```



MATERI: STACK (Jobsheet 7)

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



KELAS: 1-G

```
PROBLEMS (19) OUTPOT DEBUG CONSOLE TERMENAL PORTS

OPS D:\college\semester 2\Algoritma_dan_Struktur_Data_16_01> d:; cd 'd:\college\semester 7\eptionMessages' '-cp' 'D:\college\semester 2\Algoritma_dan_Struktur_Data_16_01\bin 'P8.Bar ta_16_01\bin 'P8.Bar ta_16_01\bin 'P9.BarangGudang_Utama01' ;cle04769-009d_48b8-90f9-398f1ccc42/5ta_16_01\bin S

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

Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang terbawah
6. Cari barang berdasarkan kode
7. Cari barang berdasarkan nama
8. Keluar
7. Lihat barang terbawah
6. Cari barang berdasarkan nama
8. Keluar
7. Lihat barang berdasarkan nama
8. Keluar
8. Keluar
9. Masukkan nama barang :
9. Masukkan barhasil ditambahkan ke Gudang
```

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

```
Menu :

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