

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN
STRUKTUR DATA**

JOBSHEET 8



MUHAMMAD AMMAR HAFIZH

(2341720074)

D-IV TEKNIK INFORMATIKA – 1E

Jurusan Teknologi Informasi

Politeknik Negeri Malang

Hasil Praktikum Percobaan 1

```
PS D:\Kuliah\Semester-2\Algoritma Dan Struktur Data\minggu8> & 'C:\Program Files\Victus\Victus.exe' 'C:\Users\VICIUS\AppData\Roaming\Code\User\workspaceStorage\9249c72b3cc1'

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Keluar
Pilih Operasi : 1
Masukan Kode Barang : 21
Masukan nama barang : Majalah
Masukan kategori : Buku
Barang Majalah berhasil ditambahkan ke Gudang

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Keluar
Pilih Operasi : 1
Masukan Kode Barang : 26
Masukan nama barang : Jaket
Masukan 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
Masukan Kode Barang : 33
Masukan nama barang : Pizza
Masukan kategori : Makanan
Barang Pizza berhasil ditambahkan ke Gudang

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Keluar
Pilih Operasi : 1
Masukan Kode Barang : 33
Masukan nama barang : Pizza
Masukan 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 21 : Majalah (Kategori Buku)
Kode 33 : Pizza (Kategori Makanan)

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Keluar
Pilih Operasi : 1
```

Pertanyaan Percobaan 1

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

- Before

```
Codeium: Refactor | Explain | Generate Javadoc | X
public Barang16 lihatBarangTeratas() {
    if (!isEmpty()) {      The method isEmpty() is undefined for the type G
        Barang16 barangTeratas = tumpukan[top];
        System.out.println("Barang teratas : " + barangTeratas.nama);
        return barangTeratas;
    } else {
        System.out.println(x:"Tumpukan barang kosong");
        return null;
    }
}
```

- After

```
Codeium: Refactor | Explain | Generate Javadoc | X
public Barang16 lihatBarangTeratas() {
    if (!cekKosong()) {
        Barang16 barangTeratas = tumpukan[top];
        System.out.println("Barang teratas : " + barangTeratas.nama);
        return barangTeratas;
    } else {
        System.out.println(x:"Tumpukan barang kosong");
        return null;
    }
}
```

- Before

```
public void tampilkanBarang() {
    if (!cekKosong()) {
        System.out.println(x:"Rincian tumpukan barang di Gudang");
        //for (int i = top; i >= 0; i--){
        for (int i = 0; i <= top; i++) {
            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");
    }
}
```

- After

```
public void tampilkanBarang() {
    if (!cekKosong()) {
        System.out.println(x:"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(x:"Tumpukan barang kosong");
    }
}
```

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

- Size pada stack Gudang adalah 7

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    Gudang16 gudang = new Gudang16(kapasitas:7);  
  
    while (true) {  
        System.out.println("Menu: ");
```

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

- Jika tidak melakukan pengecekan kondisi kosong pada stack bisa jadi stack itu sedang di posisi kosong dan akan terjadi error jika ingin mengambil barang karena tidak ada data apapun dalam stack.

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

- Size Gudang

```
Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);    Resource le

    System.out.print(s:"Kapasistas Gudang : ");
    int size = scanner.nextInt();
    Gudang16 gudang = new Gudang16(size);

    while (true) {
        System.out.println(x:"\nMenu : ");
        System.out.print(s:"1. Tambah Bara
```

- Case 4 Barang Teratas

```
while (true) {
    System.out.println(x:"\nMenu : ");
    System.out.print(s:"1. Tambah Barang");
    System.out.print(s:"2. Ambil Barang");
    System.out.print(s:"3. Tampilkan Tumpukan Barang");
    System.out.print(s:"4. Tampilkan Barang Teratas");
    System.out.print(s:"5. Keluar");
    System.out.print(s:"Pilih Operasi : ");
    int pilihan = scanner.nextInt();
    scanner.nextLine();

    switch (pilihan) {
        case 1:
            System.out.print(s:"Masukan Kode Barang : ");
            int kode = scanner.nextInt();
            scanner.nextLine();
            System.out.print(s:"Masukan nama barang : ");
            String nama = scanner.nextLine();
            System.out.print(s:"Masukan kategori : ");
            String kategori = scanner.nextLine();
            Barang16 barangBaru = new Barang16(kode, nama, kategori);
            gudang.tambahBarang(barangBaru);
            break;
        case 2:
            gudang.ambilBarang();
            break;
        case 3:
            gudang.tampilkanBarang();
            break;
        case 4:
            gudang.lihatBarangTeratas();
            break;
        case 5:
            break;
        default:
            break;
    }
}
```

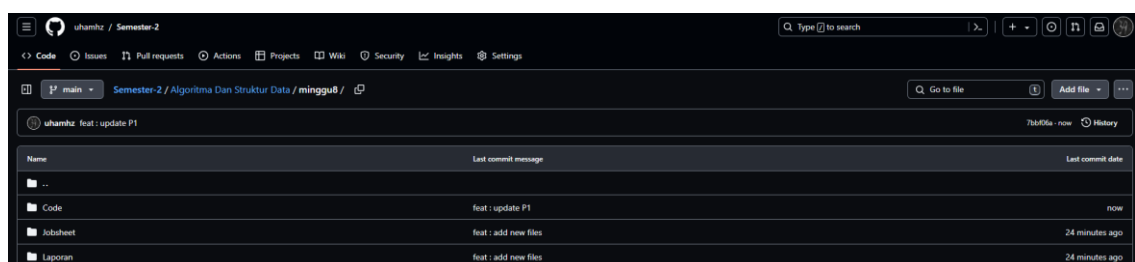
Menu :

1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. Keluar

Pilih Operasi : 4
Barang teratas : Setrika

5. Commit dan push kode program ke Github

- <https://github.com/uhamhz/Semester-2/tree/main/Algoritma%20Dan%20Struktur%20Data/minggu8>



Hasil Praktikum Percobaan 2

Menu :

1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. Keluar

Pilih Operasi : 1

Masukan Kode Barang : 13

Masukan nama barang : Setrika

Masukan kategori : 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 Operasi : 2

Barang Setrika diambil dari Gudang

Kode unik dalam Biner : 1101

Pertanyaan Percobaan 2

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

- Hasilnya sama saja, karena kondisi menginginkan kode tidak sama dengan 0 dan kondisi sebelumnya menginginkan kode lebih dari 0.

```
public String konversiBinerKeDesimal(int kode) {  
    StackKonversi16 stack = new StackKonversi16();  
    while (kode != 0) {  
        int sisa = kode % 2;  
        stack.push(sisa);  
        kode = kode / 2;  
    }  
    String biner = new String();  
    while (!stack.isEmpty()) {  
        biner = biner + stack.pop();  
    }  
    return biner;  
}
```

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

- Sebelum dikonversi kode barang akan dicek terlebih dahulu bukan sama dengan 0 Dan setelah itu kode barang dimodulus 2 dan hasilnya dimasukan ke dalam stack dan setelah itu kode dibagi 2 dan terus melakukan perulangan seperti itu sampai kode sama dengan 0 sehabis kode sisa tadi dimasukan ke dalam stake, stake akan mengeluarkan dari atas dan akan menjadi kode biner.

Hasil Praktikum Percobaan 3

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


Pertanyaan Percobaan 3

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?

- Karena ada beberapa operator yang sederajat dan apabila diubah dengan nilai yang berbeda beda akan menghasilkan postfix yang tidak benar.

2. Jelaskan alur kerja method konversi!

- Pada konversi dari infix ke postfix dibuat kolom string yang masih kosong dan dilakukan perulangan sampai n (total) jika yang ditemukan adalah operand maka akan di pindahkan ke string dan apabila buka kurung yang ditemukan akan dimasukan ke stack jika tutup kurung yang ditemukan maka tumpukan yang di stack akan di keluarkan sampai ketemu buka kurung dan jika yang ditemukan operator maka akan dibandingkan terlebih dahulu dengan stack teratas apabila lebih dari derajat pada stack teratas maka akan dipindahkan ke stack.

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

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

- C sama dengan infix bagian ke-i yang dimasukan dalam main postfix yang sudah ditrim atau dipisah-pisah (dijadikan char)

Latihan Praktikum

- Method lihatBarangTerbawah

```
Codeium: Refactor | Explain | Generate Javadoc | X
public Barang16 lihatBarangTerbawah(){
    if (!cekKosong()) {
        Barang16 barangTerbawah = tumpukan[0];
        System.out.println("Barang terbawah : " + barangTerbawah.nama);
        return barangTerbawah;
    } else {
        System.out.println(x:"Tumpukan barang kosong");
        return null;
    }
}
```

```
public class Utama16 {
    public static void main(String[] args) {

        while (true) {
            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. Tampilkan Barang Teratas");
            System.out.println(x:"5. Tampilkan Barang Terbawah");
            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:"Masukan Kode Barang : ");
                    int kode = scanner.nextInt();
                    scanner.nextLine();
                    System.out.print(s:"Masukan nama barang : ");
                    String nama = scanner.nextLine();
                    System.out.print(s:"Masukan kategori : ");
                    String kategori = scanner.nextLine();
                    Barang16 barangBaru = new Barang16(kode, nama, kategori);
                    gudang.tambahBarang(barangBaru);
                    break;
                case 2:
                    gudang.ambilBarang();
                    break;
                case 3:
                    gudang.tampilkanBarang();
                    break;
                case 4:
                    gudang.lihatBarangTeratas();
                    break;
                case 5:
                    gudang.lihatBarangTerbawah();
                    break;
            }
        }
    }
}
```

Masukan Kode Barang : 5
Masukan nama barang : Setrika
Masukan kategori : 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 Operasi : 1
Masukan Kode Barang : 6
Masukan nama barang : Jaket
Masukan kategori : 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 Operasi : 1
Masukan Kode Barang : 4
Masukan nama barang : Celana
Masukan kategori : Pakaian
Barang Celana 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 Operasi : 5
Barang terbawah : Setrika

- Method cariBarang

```

case 6:
    System.out.println(x:"1. Cari Berdasarkan Kode");
    System.out.println(x:"2. Cari Berdasarkan Nama");
    System.out.print(s:"Pilih Operasi : ");
    pilihan = scanner.nextInt();
    scanner.nextLine();
    switch (pilihan) {
        case 1:
            System.out.print(s:"Masukan Kode Barang yang dicari : ");
            int cariKode = scanner.nextInt();
            gudang.barangKode(cariKode);
            break;
        case 2:
            System.out.print(s:"Masukan Nama Barang yang dicari : ");
            String cariJudul = scanner.nextLine();
            gudang.barangNama(cariJudul);
            break;
        default:
            break;
    }
    break;

```

```

public Barang16 barangNama(String cariNama) {
    for (int i = 0; i < tumpukan.length; i++) {
        if (tumpukan[i].nama.equalsIgnoreCase(cariNama)) {
            System.out.println("Barang dengan kode " + tumpukan[i].kode + " ditemukan adalah : " + tumpukan[i].nama);
            return tumpukan[i];
        } else {
            System.out.println(x:"Barang tidak ditemukan");
        }
    }
    return null;
}

Codeium: Refactor | Explain | Generate Javadoc | X
public Barang16 barangKode(int cariKode) {
    for (int i = 0; i < tumpukan.length; i++) {
        if (tumpukan[i].kode == cariKode) {
            System.out.println("Barang dengan kode" + tumpukan[i].kode + " ditemukan adalah : " + tumpukan[i].nama);
            return tumpukan[i];
        } else {
            System.out.println(x:"Barang tidak ditemukan");
        }
    }
    return null;
}

```

```

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. Tampilkan Barang Terbawah
6. Cari Barang
7. Keluar
Pilih Operasi : 6
1. Cari Berdasarkan Kode
2. Cari Berdasarkan Nama
Pilih Operasi : 1
Masukan Kode Barang yang dicari : 1
Barang dengan kode 1 ditemukan adalah : Indomie

```

```

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. Tampilkan Barang Terbawah
6. Cari Barang
7. Keluar
Pilih Operasi : 6
1. Cari Berdasarkan Kode
2. Cari Berdasarkan Nama
Pilih Operasi : 2
Masukan Nama Barang yang dicari : Aqua
Barang tidak ditemukan
Barang dengan kode 2 ditemukan adalah : Aqua

```