TUGAS & EVALUASI

Soal Tugas & Evaluasi

1, Apa yang dimaksud dengan List, Queue, Set dan Map

Jawaban

A. List adalah Struktur data yang menyimpan secara beruntun. Ciri khas list adalah memiliki Elemen disimpan dalam urutan yang sama seperti saat dimasukkan, dan bisa diakses berdasarkan indeks

B. Queue adalah Struktur data yang mengikuti prinsip FIFO (First In, First Out). Ciri" Queue memiliki elemen pertama yang akan di masukan dan juga akan menjadi elemen pertama yang di kelurkan

C. Set adalah Koleksi yang menyimpan elemen unik (tidak ada duplikat).ciri"nya tidak menyimpan elemen secara berurutan dan tidak mengizinkan duplikat.

D. Map Adalah Struktur data yang menyimpan data dalam bentuk pasangan kunci-nilai. Ciri"nya Setiap elemen dalam Map memiliki kunci (key) yang unik yang dipetakan ke nilai (value) tertentu.

Source Code

```
Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class
```

Penjelasan

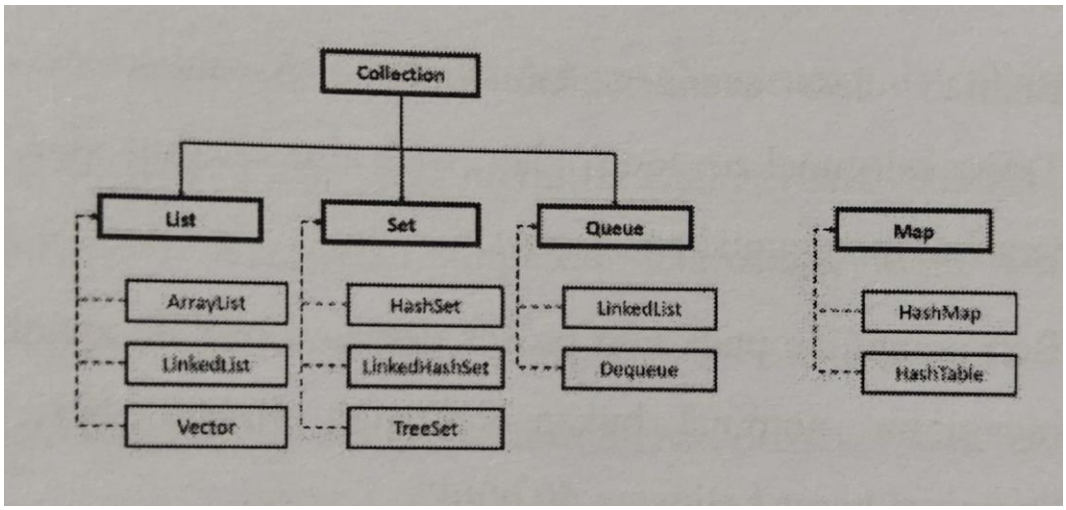
Tulis Penjelasan disini …

Output

Masukan screenshot output disini

TUGAS & EVALUASI

Soal Tugas & Evaluasi



2. Mengapa Map tidak termasuk ke dalam interface Collection?

Jawaban

Map tidak termasuk dalam interface Collection karena struktur data ini berfokus pada penyimpanan pasangan kunci-nilai (key-value pairs), berbeda dengan Collection yang hanya mengelola kumpulan elemen tunggal tanpa asosiasi kunci. Pada Map, elemen diakses berdasarkan kunci, bukan urutan atau indeks, seperti yang ada pada List, Set, atau Queue.

Source Code

```
Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class
```

Penjelasan

Tulis Penjelasan disini …

Output

Masukan screenshot output disini

TUGAS & EVALUASI

Soal Tugas & Evaluasi

3. Jelaskan apa bedanya ArrayList dan LinkedList!

Jawaban

- ArrayList menyimpan elemen dalam array dinamis, sehingga cepat untuk akses berdasarkan indeks, tetapi lambat untuk menambah atau menghapus elemen di tengah, karena perlu menggeser elemen lain.
- LinkedList, sebaliknya, menyimpan elemen dalam node yang saling terhubung, sehingga lebih lambat untuk akses indeks tertentu, tapi cepat untuk menambah atau menghapus elemen di posisi mana pun tanpa perlu menggeser.

Source Code

```
Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class
```

Penjelasan

Tulis Penjelasan disini …

Output

Masukan screenshot output disini

TUGAS & EVALUASI

Soal Tugas & Evaluasi

4. Sebuah bank ingin mengelola antrian pelanggan yang datang untuk mendapatkan layanan. Setiap kali seorang pelanggan selesai dilayani, pelanggan tersebut akan keluar dari antrian, dan pelanggan berikutnya akan dilayani. Bank juga ingin memungkinkan pelanggan VIP untuk masuk ke antrian di posisi tertentu.Data collection apa yang sebaiknya digunakan sesuai studi kasus tersebut?

Jawaban

Data collection yang paling cocok untuk kasus ini adalah `LinkedList` dengan implementasi `Queue`, karena antrian di bank mengikuti pola FIFO (First In, First Out), di mana pelanggan pertama yang datang adalah yang pertama dilayani. Selain itu, `LinkedList` memungkinkan penambahan elemen di posisi tertentu dengan mudah, sehingga memudahkan bank untuk memasukkan pelanggan VIP ke dalam antrian tanpa harus menggeser elemen lain, seperti yang terjadi pada `ArrayList`.

Source Code

```
Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class
```

Penjelasan

Tulis Penjelasan disini …

Output

Masukan screenshot output disini

TUGAS & EVALUASI

Soal Tugas & Evaluasi

5.Buatlah program berdasarkan dari studi kasus soa lsebelumnya!

Jawaban

Ketik jawaban disini …

Source Code

```
import java.util.LinkedList;
import java.util.Queue;

public class Bank {

    private Queue<String> customerQueue = new LinkedList<>();


    public void addCustomer(String customerName) {
        customerQueue.offer(customerName);
        System.out.println(customerName + " masuk antrian.");
    }


    public void serveCustomer() {
        String servedCustomer = customerQueue.poll();
        if (servedCustomer != null) {
            System.out.println(servedCustomer + " sedang
dilayani.");
        } else {
            System.out.println("Antrian kosong, tidak ada
pelanggan yang dilayani.");
        }
    }

    public void addVIPCustomer(String vipCustomerName, int
position) {
        LinkedList<String> tempQueue = new
LinkedList<>(customerQueue);
```

```
        if (position < 0 || position > tempQueue.size()) {
            System.out.println("Posisi tidak valid. Pelanggan
VIP tidak dapat ditambahkan.");
        } else {
            tempQueue.add(position, vipCustomerName);
            customerQueue = tempQueue;
            System.out.println(vipCustomerName + " masuk antrian
sebagai pelanggan VIP di posisi " + position);
        }
    }

    public void showQueue() {
        System.out.println("Antrian saat ini: " +
customerQueue);
    }

    public static void main(String[] args) {
        Bank bankQueue = new Bank();

        bankQueue.addCustomer("Pelanggan A");
        bankQueue.addCustomer("Pelanggan B");
        bankQueue.addCustomer("Pelanggan C");

        bankQueue.showQueue();
        bankQueue.addVIPCustomer("VIP Pelanggan X", 1);

        bankQueue.showQueue();

        bankQueue.serveCustomer();
        bankQueue.serveCustomer();
        bankQueue.showQueue();
    }
}
```

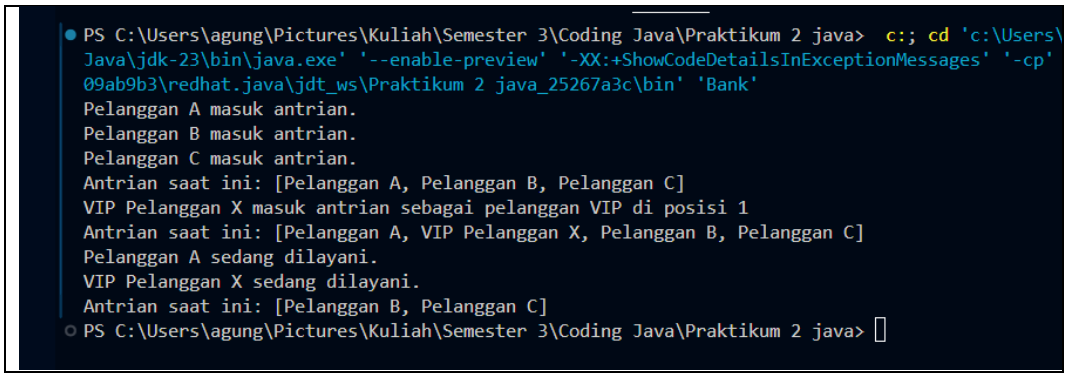
Penjelasan

Program ini mengelola antrian bank dengan dua jenis pelanggan: reguler dan VIP. Pelanggan reguler ditambahkan ke akhir antrian, sedangkan pelanggan VIP dapat ditempatkan di posisi tertentu. Program menggunakan Queue untuk pelanggan reguler dengan implementasi LinkedList, serta menyalin antrian ke

TUGAS & EVALUASI

dalam LinkedList sementara untuk menempatkan pelanggan VIP di posisi yang diinginkan. Fitur lain termasuk melayani pelanggan secara berurutan dari awal antrian dan menampilkan isi antrian saat ini.

Output



TUGAS & EVALUASI

Soal Tugas & Evaluasi

6. Vincent diminta dosennya untuk membuat sebuah sistem manajemen perpustakaan sederhana. Sistem ini harus mampu mengelola data buku, anggota perpustakaan, dan transaksi peminjaman buku. Tidak hanya satu Vincent diwajibkan menggunakan berbagai jenis data collection seperti ArrayList, HashMap, dan Queue. Buatlah program untuk Vincent

Jawaban

Ketik jawaban disini …

Source Code

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

class Buku {
    private String id;
    private String judul;
    private boolean tersedia;

    public Buku(String id, String judul) {
        this.id = id;
        this.judul = judul;
        this.tersedia = true;
    }

    public String getId() {
        return id;
    }

    public String getJudul() {
        return judul;
    }
```

```
    public boolean isTersedia() {
        return tersedia;
    }

    public void setTersedia(boolean tersedia) {
        this.tersedia = tersedia;
    }


    public String deskripsi() {
        return "ID: " + id + ", Judul: " + judul + ", Tersedia:
" + tersedia;
    }
}

class Anggota {
    private String id;
    private String nama;

    public Anggota(String id, String nama) {
        this.id = id;
        this.nama = nama;
    }

    public String getId() {
        return id;
    }

    public String getNama() {
        return nama;
    }


    public String deskripsi() {
        return "ID: " + id + ", Nama: " + nama;
    }
}
```

```
class Perpustakaan {
    private ArrayList<Buku> daftarBuku;
    private HashMap<String, Anggota> daftarAnggota;
    private Queue<String> antreanPeminjaman;

    public Perpustakaan() {
        daftarBuku = new ArrayList<>();
        daftarAnggota = new HashMap<>();
        antreanPeminjaman = new LinkedList<>();
    }

    public void tambahBuku(Buku buku) {
        daftarBuku.add(buku);
        System.out.println("Buku ditambahkan: " +
buku.deskripsi());
    }

    public void tambahAnggota(Anggota anggota) {
        daftarAnggota.put(anggota.getId(), anggota);
        System.out.println("Anggota ditambahkan: " +
anggota.deskripsi());
    }

    public void pinjamBuku(String idBuku, String idAnggota) {
        Buku buku = cariBukuById(idBuku);
        Anggota anggota = daftarAnggota.get(idAnggota);

        if (buku == null) {
            System.out.println("Buku tidak ditemukan!");
            return;
        }
        if (anggota == null) {
            System.out.println("Anggota tidak ditemukan!");
            return;
        }
        if (buku.isTersedia()) {
            buku.setTersedia(false);
```

```
            System.out.println("Buku dipinjam oleh: " +
anggota.getNama());
        } else {
            antreanPeminjaman.add(idBuku);
            System.out.println("Buku sedang dipinjam.
Ditambahkan ke antrean.");
        }
    }

    public void kembalikanBuku(String idBuku) {
        Buku buku = cariBukuById(idBuku);
        if (buku == null) {
            System.out.println("Buku tidak ditemukan!");
            return;
        }
        buku.setTersedia(true);
        System.out.println("Buku dikembalikan: " +
buku.getJudul());

        if (!antreanPeminjaman.isEmpty() &&
antreanPeminjaman.peek().equals(idBuku)) {
            antreanPeminjaman.poll();
            System.out.println("Anggota berikutnya dalam antrean
telah diberi tahu untuk buku: " + buku.getJudul());
        }
    }

    public void tampilkanBuku() {
        System.out.println("Daftar Buku di Perpustakaan:");
        for (Buku buku : daftarBuku) {
            System.out.println(buku.deskripsi());
        }
    }

    public void tampilkanAnggota() {
        System.out.println("Daftar Anggota Perpustakaan:");
        for (Anggota anggota : daftarAnggota.values()) {
            System.out.println(anggota.deskripsi());
        }
```

```
    }

    private Buku cariBukuById(String id) {
        for (Buku buku : daftarBuku) {
            if (buku.getId().equals(id)) {
                return buku;
            }
        }
        return null;
    }
}

public class SistemPerpustakaan {
    public static void main(String[] args) {
        Perpustakaan perpustakaan = new Perpustakaan();
        try (Scanner scanner = new Scanner(System.in)) {
            // Data contoh
            perpustakaan.tambahBuku(new Buku("B1", "Pemrograman
Java"));
            perpustakaan.tambahBuku(new Buku("B2", "Struktur
Data"));
            perpustakaan.tambahAnggota(new Anggota("A1",
"Vincent"));
            perpustakaan.tambahAnggota(new Anggota("A2",
"Emily"));

            int pilihan;
            do {
                System.out.println("\nMenu Perpustakaan:");
                System.out.println("1. Tampilkan Buku");
                System.out.println("2. Tampilkan Anggota");
                System.out.println("3. Pinjam Buku");
                System.out.println("4. Kembalikan Buku");
                System.out.println("5. Keluar");
                System.out.print("Masukkan pilihan: ");
                pilihan = scanner.nextInt();
                scanner.nextLine();

                switch (pilihan) {
```

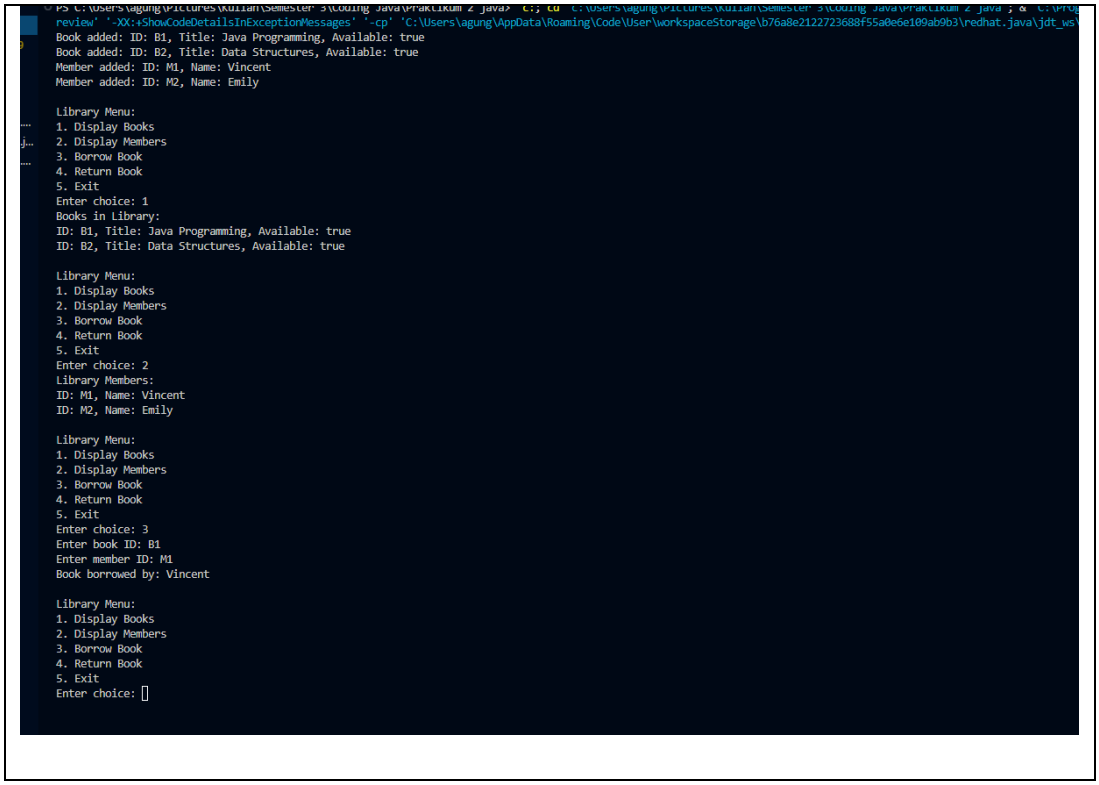
```
                    case 1 -> perpustakaan.tampilkanBuku();
                    case 2 -> perpustakaan.tampilkanAnggota();
                    case 3 -> {
                        System.out.print("Masukkan ID buku: ");
                        String idBuku = scanner.nextLine();
                        System.out.print("Masukkan ID anggota:
");
                        String idAnggota = scanner.nextLine();
                        perpustakaan.pinjamBuku(idBuku,
idAnggota);
                    }
                    case 4 -> {
                        System.out.print("Masukkan ID buku: ");
                        String idBuku = scanner.nextLine();
                        perpustakaan.kembalikanBuku(idBuku);
                    }
                    case 5 -> System.out.println("Keluar...");
                    default -> System.out.println("Pilihan tidak
valid!");
                }
            } while (pilihan != 5);
        }
    }
}
```

Penjelasan

Program ini mengelola sistem perpustakaan sederhana yang memungkinkan penambahan buku dan anggota, peminjaman buku, dan pengembalian buku. Buku disimpan dalam ArrayList, anggota dalam HashMap, dan antrian peminjaman dalam Queue berbasis LinkedList. Jika buku yang dipinjam sedang tidak tersedia, ID buku ditambahkan ke antrian. Fitur lain termasuk menampilkan daftar buku dan anggota yang terdaftar di perpustakaan.

TUGAS & EVALUASI

Output



TUGAS & EVALUASI

Soal Tugas & Evaluasi

Jawaban

Ketik jawaban disini …

Source Code

```
Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class
```

Penjelasan

Tulis Penjelasan disini …

Output

Masukan screenshot output disini