**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

**QUIS 1**

****

**NAMA : ALVINO VALERIAN D.R**

**KELAS : 1A**

**NO. ABSEN : 05**

**NIM : 2341720027**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

public class LayananService {

layanan head;

int size;

public LayananService(){

head=null;

size=0;

}

public static int totalPemasukan = 0;

public static void addService(layanan layanan) {

totalPemasukan += layanan.getHarga();

}

public static int getTotalPemasukan() {

return totalPemasukan;

}

public boolean isEmpty(){

return head==null;

}

public void tambahLayanan(int kodeLayanan, String namaLayanan, int harga){

if (isEmpty()) {

head = new layanan(kodeLayanan, namaLayanan, harga, null, null);

} else {

layanan current = head;

while (current.next != null) {

current = current.next;

}

layanan newNode = new layanan(kodeLayanan, namaLayanan, harga, current, null);

current.next = newNode;

}

size++;

}

}

public class Dll05 {

pelanggan head;

int size;

public Dll05(){

head =null;

size=0;

}

public boolean isEmpty(){

return head==null;

}

public void addFirst(String namaPelanggan,String noHp){

if (isEmpty()) {

head = new pelanggan(namaPelanggan, noHp, null, null);

}else{

pelanggan node = new pelanggan(namaPelanggan, noHp, null, head);

head.prev = node;

head = node;

}

size++;

}

public void addLast(String namaPelanggan,String noHp){

if (isEmpty()) {

addFirst(namaPelanggan,noHp);

}else{

pelanggan current = head;

while (current.next != null) {

current = current.next;

}

pelanggan newNode = new pelanggan(namaPelanggan, noHp, current, null);

current.next = newNode;

size++;

}

}

public int size(){

return size;

}

public void clear(){

head = null;

size=0;

}

public void print(){

if (!isEmpty()) {

pelanggan tmp = head;

System.out.println("Nama\t Nomer\t");

while (tmp != null) {

System.out.println(tmp.namaPelanggan+"\t" + tmp.noHp+" \t");

tmp=tmp.next;

}

System.out.println("Sisa Antrian: "+size);

}else{

System.out.println("Tidak ada Antrian");

}

}

public void removeFirst() throws Exception{

if (isEmpty()) {

throw new Exception("tidak ada yang mengantri");

}else if (size == 1) {

removeLast();

}else{

head=head.next;

pelanggan penerima = head.prev;

System.out.println(penerima.namaPelanggan+ " Selesai Perbaikan");

head.prev=null;

size--;

}

}

public void removeLast() throws Exception{

if (isEmpty()) {

throw new Exception("tidak ada yang mengantri");

}else if (head.next == null) {

pelanggan penerima = head;

System.out.println(penerima.namaPelanggan+" Selesai Perbaikan");

head=null;

size--;

return;

}

}

public void removeAndAddService(layanan layanan) throws Exception{

if (!isEmpty()) {

removeFirst();

LayananService.addService(layanan);

} else {

System.out.println("Antrian kosong.");

}

}

}

import java.util.Scanner;

public class DLL05Main {

public static void main(String[] args) throws Exception {

Scanner sc = new Scanner(System.in);

Dll05 antrian05 = new Dll05();

int menu;

do {

System.out.println("---------DAFTAR PENGANTRI-------");

System.out.println(" 1.TAMBAH ANTRIAN ");

System.out.println(" 2.CETAK ANTRIAN ");

System.out.println(" 3.HAPUS ANTRIAN SEKALIGUS TAMBAH LAYANAN ");

System.out.println(" 4.LAPORAN LAYANAN TERDAPAT TOTAL PEMASUKAN ");

System.out.println(" 5.KELUAR ");

menu =sc.nextInt();

sc.nextLine();

switch (menu) {

case 1:

System.out.print("masukkan nama : ");

String nama = sc.nextLine();

System.out.print("masukkan no HP : ");

String noHp =sc.nextLine();

antrian05.addLast(nama, noHp);

break;

case 2:

antrian05.print();

break;

case 3:

if (!antrian05.isEmpty()) {

System.out.println("Masukkan detail layanan:");

System.out.print("Kode Layanan: ");

int kodeLayanan = sc.nextInt();

sc.nextLine();

System.out.print("Nama Layanan: ");

String namaLayanan = sc.nextLine();

System.out.print("Harga Layanan: ");

int hargaLayanan = sc.nextInt();

sc.nextLine();

layanan layanan = new layanan(kodeLayanan, namaLayanan, hargaLayanan, null, null);

antrian05.removeAndAddService(layanan);

System.out.println("Antrian dihapus dan layanan berhasil ditambahkan.");

} else {

System.out.println("Antrian kosong.");

}

break;

case 4:

System.out.println("Total Pemasukan dari Layanan: Rp " + LayananService.getTotalPemasukan());

break;

case 5:

System.exit(0);

break;

default:

System.out.println("input salah!");

break;

}

} while (menu !=5);

sc.close();

}

}

public class pelanggan {

String namaPelanggan;

String noHp;

pelanggan next,prev;

pelanggan(String namaPelanggan,String noHp,pelanggan next,pelanggan prev){

this.namaPelanggan=namaPelanggan;

this.noHp=noHp;

this.next=next;

this.prev=prev;

}

public class LayananService {

private static int totalPemasukan = 0;

public static void addService(layanan layanan) {

totalPemasukan += layanan.getHarga();

}

public static int hasilTotalPemasukan() {

return totalPemasukan;

}

}

}

public class layanan {

int kodeLayanan;

String namaLayanan;

int harga;

layanan next1,prev1;

public layanan next;

layanan(int kodeLayanan,String namaLayanan,int harga,layanan next1,layanan prev1){

this.kodeLayanan=kodeLayanan;

this.namaLayanan=namaLayanan;

this.harga=harga;

this.next1=next1;

this.prev1=prev1;

}

public String hasilNamaLayanan() {

return namaLayanan;

}

public int getHarga() {

return harga;

}

private static int totalPemasukan = 0;

public static void addService(layanan layanan) {

totalPemasukan += layanan.getHarga();

}

public static int hasilTotalPemasukan() {

return totalPemasukan;

}

}

