MODUL 11 LAPORAN
PRAKTIKUM STRUKTUR DATA

Disusun oleh :

Nama : Fidelia Ping

NIM : 245410012

Kelas : Informatika 1

PROGRAM STUDI INFORMATIKA

PROGRAM SARJANA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

YOGYAKARTA

2025

MODUL 11
MENGURUTKAN DATA (SORTING) DAN PENCARIAN DATA (SEARCHING) PADA LINKEDLIST

A. TUJUAN PRAKTIKUM

Mahasiswa dapat melakukan pengurutan terhadap suatu data yang terdapat di dalam linkedlist

B. DASAR TEORI

SORTING PADA LINKEDLIST

Pada modul 5 kita yang lalu telah mempelajari bagaimana cara mengurutkan data yang tersimpan dalam sebuah sederet Larik/ array. Pada modul 11 ini kita akan mempelajari bagaimana melakukan cara mengurutkan data pada sebuah linkedlist. Dalam prosesnya, pengurutan data dalam sebuah larik/array sangat mudah dilakukan. Hal ini karena pada struktur penyimpan larik terdapat indeks yang dapat membantu menandai data-data yang hendak dibandingkan/ ditukar. Tidak seperti pada larik, pengurutan data dalam sebuah linkedlist lebih sulit dilakukan. Hal ini karena pada saat kita hendak membandingkan/ menukar data, tidak ada indeks yang dapat membantu untuk menandai data-data tersebut. Untuk itu diperlukan cara lain untuk menggantikan peran indeks tersebut. Pada modul ini kita akan mempelajari bagaimana membuat program untuk mengurutkan data pada struktur penyimpan linkedlist. Metode pengurutan data yang akan kita gunakan adalah bubblesort, selection sort, dan insertion sort.

Sebelum kita mempelajari bagaimana program pengurutan pada linkedlist, perlu kita pahami terlebih dahulu bahwa di dalam banyak metode pengurutan data terdapat dua proses penting yang selalu ada, yaitu :

1. membandingkan dua buah data,
2. menukar kedua data tersebut jika diperlukan.

Pada proses membandingkan dua data, baik untuk struktur penyimpan array/larik maupun linkedlist, cara yang diterapkan cenderung sama, yaitu sama-sama harusditemukan dahulu data yang dimaksud. (pada metode yang berbeda akan berbeda pula cara menemukannya). Dalam proses menukar data, pada array/larik penukaran data hanya dapat dilakukan dengan menukar isi atau nilai (value) dari variabel yang menyimpan data-data tersebut. Sementara pada linkedlist penukaran data dapat dilakukan dengan 2 pendekatan :

1. Menukar isi variabelnya (yaitu dengan cara menukar isi heap namun posisi heap tidak berubah)
2. Menukar posisi heap (tukar heap, yaitu dengan merubah posisi heap/ simpul-simpul dalam linkedlist)

Pada modul ini akan kita pelajari bagaimana melakukan penukaran data dalam linkedlist dengan kedua teknik di atas

Metode Sorting	Single LinkedList		Double LinkedList	
	Tukar Nilai	Tukar Heap	Tukar Nilai	Tukar Heap
Bubble Sort	*ada*	*ada*	*ada*	*ada*
Selection Sort	*ada*	-	*ada*	*ada*
Insertion Sort	-	*ada*	-	*ada*

C. PEMBAHASAN LISTING PRAKTIKUM

Program 11.1

```
import java.util.Scanner;

class Node {
    // Menambahkan atribut sesuai request kode baru (umur, jekel, ipk)
    String nama, alamat, jekel;
    int umur;
    double ipk;
    Node next, prev;

    public Node(String nama, String alamat, int umur, String jekel, double
ipk) {
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.jekel = jekel;
        this.ipk = ipk;
        next = null;
        prev = null;
    }
}

class DoubleLinkedList {
    Node head, tail;

    public DoubleLinkedList() {
        head = tail = null;
    }

    // --- FITUR UTAMA (CRUD) ---

    void tambahDepan(String nama, String alamat, int umur, String jekel,
double ipk) {
        Node baru = new Node(nama, alamat, umur, jekel, ipk);
        if (head == null) {
            head = tail = baru;
        } else {
            baru.next = head;
            head.prev = baru;
            head = baru;
        }
        System.out.println("Data " + nama + " berhasil ditambah di depan.");
    }
```

```
    void tambahBelakang(String nama, String alamat, int umur, String jekel,
double ipk) {
        Node baru = new Node(nama, alamat, umur, jekel, ipk);
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
            tail = baru;
        }
        System.out.println("Data " + nama + " berhasil ditambah di
belakang.");
    }

    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equalsIgnoreCase(nama)) {
            bantu = bantu.next;
        }

        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
            return;
        }

        if (bantu == head && bantu == tail) {
            head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
        System.out.println("Data " + nama + " berhasil dihapus.");
    }

    void cetakMaju() {
        Node bantu = head;
        System.out.println("\n--- CETAK DATA (MAJU) ---");
        if (head == null) System.out.println("Data Kosong");
        while (bantu != null) {
            System.out.printf("%-10s | %-10s | Umur: %d | L/P: %s | IPK:
%.2f\n",
                bantu.nama, bantu.alamat, bantu.umur, bantu.jekel,
bantu.ipk);
            bantu = bantu.next;
        }
    }

    // --- FITUR PENDUKUNG SORTING & SEARCHING (INTEGRASI BARU) ---

    // 1. Method hitung jumlah simpul (diperlukan oleh Bubble Sort)
    public int hitungJumlahSimpul() {
```

```
        int jumlah = 0;
        Node bantu = head;
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }

    // 2. Method Tukar Nilai (Sesuai request Anda)
    public void tukarNilai(Node X, Node Y) {
        // Variabel sementara untuk menampung data X
        String tempNama = X.nama;
        String tempAlamat = X.alamat;
        int tempUmur = X.umur;
        String tempJekel = X.jekel;
        double tempIpk = X.ipk;

        // Pindahkan data Y ke X
        X.nama = Y.nama;
        X.alamat = Y.alamat;
        X.umur = Y.umur;
        X.jekel = Y.jekel;
        X.ipk = Y.ipk;

        // Pindahkan data sementara (X awal) ke Y
        Y.nama = tempNama;
        Y.alamat = tempAlamat;
        Y.umur = tempUmur;
        Y.jekel = tempJekel;
        Y.ipk = tempIpk;
    }

    // 3. Method Bubble Sort (Teknik Tukar Nilai)
    public void mengurutkanDataBubble() {
        if (head == null || head.next == null) {
            System.out.println("Data tidak cukup untuk diurutkan.");
            return;
        }

        System.out.println("\nSedang mengurutkan data (Bubble Sort By
Name)...");
        int N = hitungJumlahSimpul();
        Node A = null;
        Node B = null;
        Node berhenti = null; // Dalam Java LinkedList tail.next adalah null

        for (int i = 1; i <= N - 1; i++) {
            A = head;       // awal
            B = head.next;  // awal.kanan

            while (B != berhenti) {
                // Bandingkan Nama (Ascending A-Z)
                if (A.nama.compareToIgnoreCase(B.nama) > 0) {
                    tukarNilai(A, B);
                }
                A = A.next;
```

```
                B = B.next;
            }
            berhenti = A; // Optimasi Bubble Sort (kurangi range)
        }
        System.out.println("=== PROSES PENGURUTAN SELESAI ===");
    }

    // 4. Method Linear Search (Pencarian Linear)
    public void cariLinearSearch(String keyword) {
        Node bantu = head;
        boolean ditemukan = false;
        int posisi = 1;

        System.out.println("\n--- HASIL PENCARIAN LINEAR ---");
        while (bantu != null) {
            // Mencari berdasarkan kesamaan Nama
            if (bantu.nama.equalsIgnoreCase(keyword)) {
                System.out.println("DITEMUKAN pada posisi ke-" + posisi);
                System.out.println("Nama   : " + bantu.nama);
                System.out.println("Alamat : " + bantu.alamat);
                System.out.println("Umur   : " + bantu.umur);
                System.out.println("IPK    : " + bantu.ipk);
                ditemukan = true;
                break; // Hentikan loop jika ditemukan
            }
            bantu = bantu.next;
            posisi++;
        }

        if (!ditemukan) {
            System.out.println("Data dengan nama '" + keyword + "' tidak
ditemukan.");
        }
    }
}

public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();

        int pilih, umur;
        double ipk;
        String nama, alamat, jekel, dicari;

        do {
            System.out.println("\n=== MENU PROGRAM MASTER ===");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Hapus Data");
            System.out.println("4. Cetak Data (Lihat Hasil)");
            System.out.println("5. Urutkan Data (Bubble Sort)");
            System.out.println("6. Cari Data (Linear Search)");
            System.out.println("7. Keluar");
            System.out.print("Pilih menu: ");
            pilih = sc.nextInt();
            sc.nextLine(); // clear buffer
```

```
            switch (pilih) {
                case 1:
                case 2:
                    System.out.print("Nama    : "); nama = sc.nextLine();
                    System.out.print("Alamat  : "); alamat = sc.nextLine();
                    System.out.print("Umur    : "); umur = sc.nextInt();
                    sc.nextLine(); // clear buffer
                    System.out.print("L/P     : "); jekel = sc.nextLine();
                    System.out.print("IPK     : "); ipk = sc.nextDouble();

                    if (pilih == 1) dll.tambahDepan(nama, alamat, umur,
jekel, ipk);
                    else dll.tambahBelakang(nama, alamat, umur, jekel, ipk);
                    break;

                case 3:
                    System.out.print("Masukkan nama yang dihapus: ");
                    nama = sc.nextLine();
                    dll.hapus(nama);
                    break;

                case 4:
                    dll.cetakMaju();
                    break;

                case 5:
                    dll.mengurutkanDataBubble();
                    // Langsung tampilkan hasil setelah sort
                    dll.cetakMaju();
                    break;

                case 6:
                    System.out.print("Masukkan nama yang dicari: ");
                    dicari = sc.nextLine();
                    dll.cariLinearSearch(dicari);
                    break;

                case 7:
                    System.out.println("Terima kasih.");
                    break;

                default:
                    System.out.println("Pilihan salah.");
            }
        } while (pilih != 7);
    }
}
```

```
=== MENU PROGRAM MASTER ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data (Lihat Hasil)
5. Urutkan Data (Bubble Sort)
6. Cari Data (Linear Search)
7. Keluar
Pilih menu: 1
Nama    : Fidel
Alamat  : Kalimantan
Umur    : 19
L/P     : P
IPK     : 4.0
Data Fidel berhasil ditambah di depan.

=== MENU PROGRAM MASTER ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data (Lihat Hasil)
5. Urutkan Data (Bubble Sort)
6. Cari Data (Linear Search)
7. Keluar
Pilih menu: 1
Nama    : Mian
Alamat  : Sulsel
Umur    : 19
L/P     : P
IPK     : 3.9
Data Mian berhasil ditambah di depan.

=== MENU PROGRAM MASTER ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data (Lihat Hasil)
5. Urutkan Data (Bubble Sort)
6. Cari Data (Linear Search)
7. Keluar
Pilih menu: 1
Nama    : Putra
Alamat  : Kaltara
Umur    : 19
L/P     : L
IPK     : 3.9
Data Putra berhasil ditambah di depan.
```

```
=== MENU PROGRAM MASTER ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data (Lihat Hasil)
5. Urutkan Data (Bubble Sort)
6. Cari Data (Linear Search)
7. Keluar
Pilih menu: 2
Nama    : Melia
Alamat  : Kaltim
Umur    : 19
L/P     : P
IPK     : 4.0
Data Melia berhasil ditambah di belakang.
```

```
=== MENU PROGRAM MASTER ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data (Lihat Hasil)
5. Urutkan Data (Bubble Sort)
6. Cari Data (Linear Search)
7. Keluar
Pilih menu: 5

Sedang mengurutkan data (Bubble Sort By Name)...
=== PROSES PENGURUTAN SELESAI ===

--- CETAK DATA (MAJU) ---
Fidel      | Kalimantan | Umur: 19 | L/P: P | IPK: 4.00
Melia      | Kaltim     | Umur: 19 | L/P: P | IPK: 4.00
Mian       | Sulsel     | Umur: 19 | L/P: P | IPK: 3.90
Putra      | Kaltara    | Umur: 19 | L/P: L | IPK: 3.90

=== MENU PROGRAM MASTER ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data (Lihat Hasil)
5. Urutkan Data (Bubble Sort)
6. Cari Data (Linear Search)
7. Keluar
Pilih menu: 4

--- CETAK DATA (MAJU) ---
Fidel      | Kalimantan | Umur: 19 | L/P: P | IPK: 4.00
Melia      | Kaltim     | Umur: 19 | L/P: P | IPK: 4.00
Mian       | Sulsel     | Umur: 19 | L/P: P | IPK: 3.90
Putra      | Kaltara    | Umur: 19 | L/P: L | IPK: 3.90
```

Pembahasan : Program Java di atas merupakan aplikasi pengelolaan data mahasiswa yang menerapkan konsep **Double Linked List (senarai ganda)** untuk menyimpan dan memanipulasi data berupa nama, alamat, umur, jenis kelamin, dan IPK. Hal ini dapat dibuktikan dari struktur Node yang memiliki dua referensi, yaitu next untuk menunjuk ke simpul berikutnya dan prev untuk menunjuk ke simpul sebelumnya, serta adanya atribut head dan tail pada kelas DoubleLinkedList. Program ini menyediakan fitur CRUD (tambah depan, tambah belakang, hapus, dan cetak data), dilengkapi dengan algoritma **Bubble Sort** untuk mengurutkan data berdasarkan nama menggunakan teknik tukar nilai, serta **Linear Search** untuk mencari data secara berurutan. Dengan menu interaktif berbasis console, program ini menunjukkan implementasi lengkap struktur data double linked list beserta operasi traversal, pengurutan, dan pencarian secara terintegrasi

Program 11.2

```
import java.util.Scanner;

class Node {
    String nama, alamat;
    Node next, prev; // next = kanan, prev = kiri

    public Node(String nama, String alamat) {
        this.nama = nama;
        this.alamat = alamat;
        next = null;
        prev = null;
    }
}

class DoubleLinkedList {
    Node head, tail; // head = awal, tail = akhir

    public DoubleLinkedList() {
        head = tail = null;
    }

    // --- FUNGSI DASAR (Sesuai Program Master Awal) ---

    // Tambah Depan
    void tambahDepan(String nama, String alamat) {
        Node baru = new Node(nama, alamat);
        if (head == null) {
            head = tail = baru;
        } else {
            baru.next = head;
            head.prev = baru;
            head = baru;
        }
    }

    // Tambah Belakang
    void tambahBelakang(String nama, String alamat) {
        Node baru = new Node(nama, alamat);
```

```
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
            tail = baru;
        }
    }

    // Tambah Tengah
    void tambahTengah(String dicari, String nama, String alamat) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(dicari)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data yang dicari tidak ditemukan!");
            return;
        }
        Node baru = new Node(nama, alamat);
        Node sesudah = bantu.next;
        bantu.next = baru;
        baru.prev = bantu;
        if (sesudah != null) {
            baru.next = sesudah;
            sesudah.prev = baru;
        } else {
            tail = baru;
        }
    }

    // Hapus
    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(nama)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
            return;
        }
        if (bantu == head && bantu == tail) {
            head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
```

```
    }

    void cetakMaju() {
        Node bantu = head;
        System.out.println("\nCETAK MAJU :");
        while (bantu != null) {
            System.out.println(bantu.nama + " - " + bantu.alamat);
            bantu = bantu.next;
        }
    }

    void cetakMundur() {
        Node bantu = tail;
        System.out.println("\nCETAK MUNDUR :");
        while (bantu != null) {
            System.out.println(bantu.nama + " - " + bantu.alamat);
            bantu = bantu.prev;
        }
    }

    // --- FUNGSI TAMBAHAN UNTUK SORTING ---

    // Fungsi hitungJumlahSimpul (Dibutuhkan oleh program sort baru Anda)
    public int hitungJumlahSimpul() {
        int jumlah = 0;
        Node bantu = head;
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }

    // --- PROGRAM BARU YANG ANDA MINTA ---
    // Logika tidak diubah, hanya penyesuaian nama variabel (awal->head, kanan-
>next, simpul->Node)
    public void mengurutkanDataBubble_TeknikTukarHeap() {
        int N = hitungJumlahSimpul();
        Node A = null;
        Node B = null;
        Node bantu = null;
        Node berhenti = null; // Di Java, tail.next (akhir.kanan) defaultnya
null

        int nomor;
        System.out.println("Banyaknya simpul = " + hitungJumlahSimpul());

        for (int i = 1; i <= hitungJumlahSimpul() - 1; i++) {
            A = head;       // A = awal
            B = head.next;  // B = awal.kanan
            nomor = 1;

            // Proses banding-tukar, khusus simpul pertama dgn sebelahnya
```

```
            if (A.nama.compareTo(B.nama) > 0) {
                A.next = B.next; // A.kanan = B.kanan
                B.next = A;      // B.kanan = A
                head = B;        // awal = B
            }

            // Proses banding-tukar, simpul kedua dgn sebelahnya, dst
            nomor++;
            bantu = head; // bantu = awal

            // while (bantu.kanan.kanan != berhenti)
            while (bantu.next != null && bantu.next.next != berhenti) {
                A = bantu.next;       // A = bantu.kanan
                B = bantu.next.next;  // B = bantu.kanan.kanan

                if (A.nama.compareTo(B.nama) > 0) {
                    // Tukarkan simpul A dan simpul B
                    A.next = B.next; // A.kanan = B.kanan
                    B.next = A;      // B.kanan = A
                    bantu.next = B;  // bantu.kanan = B

                    if (B == tail) { // if (B == akhir)
                        tail = A;    // akhir = A
                    }
                }
                bantu = bantu.next; // bantu = bantu.kanan
                nomor++;
            }
            berhenti = bantu.next; // berhenti = bantu.kanan
            System.out.println("");
        }
        System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI======");
    }
}

public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();

        int pilih;
        String nama, alamat, dicari;

        do {
            System.out.println("\nMENU:");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Tambah Tengah");
            System.out.println("4. Hapus");
            System.out.println("5. Cetak Maju");
            System.out.println("6. Cetak Mundur");
            System.out.println("7. Urutkan (Bubble Sort - Tukar Heap)");
            System.out.println("8. Keluar");
```

```
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    dll.tambahDepan(nama, alamat);
                    break;
                case 2:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    dll.tambahBelakang(nama, alamat);
                    break;
                case 3:
                    System.out.print("Cari Nama : "); dicari = sc.nextLine();
                    System.out.print("Nama Baru : "); nama = sc.nextLine();
                    System.out.print("Alamat Baru: "); alamat = sc.nextLine();
                    dll.tambahTengah(dicari, nama, alamat);
                    break;
                case 4:
                    System.out.print("Nama Hapus: "); nama = sc.nextLine();
                    dll.hapus(nama);
                    break;
                case 5:
                    dll.cetakMaju();
                    break;
                case 6:
                    dll.cetakMundur();
                    break;
                case 7: // Menu untuk method baru Anda
                    dll.mengurutkanDataBubble_TeknikTukarHeap();
                    dll.cetakMaju();
                    break;
            }

        } while (pilih != 8);
    }
}
```

```
MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 1
Nama   : Fidel
Alamat : Kaltim

MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 1
Nama   : Melia
Alamat : Kaltim

MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 1
Nama   : Mian
Alamat : Sulsel
```

```
MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 2
Nama   : Putra
Alamat : Kaltara

MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 2
Nama   : Arlinda
Alamat : Medan
```

```
MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 5

CETAK MAJU :
Mian - Sulsel
Melia - Kaltim
Fidel - Kaltim
Putra - Kaltara
Arlinda - Medan

MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 6

CETAK MUNDUR :
Arlinda - Medan
Putra - Kaltara
Fidel - Kaltim
Melia - Kaltim
Mian - Sulsel
```

```
MENU:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Heap)
8. Keluar
Pilih: 7
Banyaknya simpul = 5




===PROSES PENGURUTAN BUBBLE SELESAI======

CETAK MAJU :
Arlinda - Medan
Fidel - Kaltim
Melia - Kaltim
Mian - Sulsel
Putra - Kaltara
```

Pembahasan : Program Java di atas merupakan aplikasi pengelolaan data nama dan alamat yang menggunakan struktur data **Double Linked List (senarai ganda)**, yang ditandai dengan adanya dua referensi pada setiap simpul (next dan prev) serta penggunaan penunjuk head dan tail untuk mengakses data dari awal dan akhir. Program ini menyediakan operasi dasar seperti tambah data di depan, belakang, dan tengah, hapus data, serta pencetakan data secara maju dan mundur, sehingga menunjukkan kemampuan traversal dua arah yang menjadi ciri utama double linked list. Selain itu, program juga dilengkapi dengan algoritma **Bubble Sort menggunakan teknik tukar simpul (heap/node swapping)** untuk mengurutkan data berdasarkan nama tanpa menukar isi data, melainkan memanipulasi hubungan antar simpul. Dengan menu interaktif berbasis console, program ini secara keseluruhan memperlihatkan penerapan konsep double linked list yang lengkap, mulai dari manipulasi struktur, traversal dua arah, hingga proses pengurutan data.

Program 11.3

```
import java.util.Scanner;

class Node {
    String nama, alamat;
    Node next, prev; // next = kanan, prev = kiri

    public Node(String nama, String alamat) {
        this.nama = nama;
        this.alamat = alamat;
        next = null;
        prev = null;
    }
}
```

```
class DoubleLinkedList {
    Node head, tail; // head = awal, tail = akhir

    public DoubleLinkedList() {
        head = tail = null;
    }

    // --- FUNGSI CRUD (MASTER PROGRAM AWAL) ---

    // Tambah Depan
    void tambahDepan(String nama, String alamat) {
        Node baru = new Node(nama, alamat);
        if (head == null) {
            head = tail = baru;
        } else {
            baru.next = head;
            head.prev = baru;
            head = baru;
        }
    }

    // Tambah Belakang
    void tambahBelakang(String nama, String alamat) {
        Node baru = new Node(nama, alamat);
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
            tail = baru;
        }
    }

    // Tambah Tengah
    void tambahTengah(String dicari, String nama, String alamat) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(dicari)) {
            bantu = bantu.next;
        }

        if (bantu == null) {
            System.out.println("Data yang dicari tidak ditemukan!");
            return;
        }

        Node baru = new Node(nama, alamat);
        Node sesudah = bantu.next;

        bantu.next = baru;
        baru.prev = bantu;

        if (sesudah != null) {
```

```
            baru.next = sesudah;
            sesudah.prev = baru;
        } else {
            tail = baru;
        }
    }

    // Hapus
    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(nama)) {
            bantu = bantu.next;
        }

        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
            return;
        }

        if (bantu == head && bantu == tail) {
           head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
    }

    // Cetak Maju
    void cetakMaju() {
        Node bantu = head;
        System.out.println("\nCETAK MAJU :");
        while (bantu != null) {
            System.out.println(bantu.nama + " - " + bantu.alamat);
            bantu = bantu.next;
        }
    }

    // Cetak Mundur
    void cetakMundur() {
        Node bantu = tail;
        System.out.println("\nCETAK MUNDUR :");
        while (bantu != null) {
            System.out.println(bantu.nama + " - " + bantu.alamat);
            bantu = bantu.prev;
        }
    }
```

```
    // --- FUNGSI TAMBAHAN (INTEGRASI BARU) ---

    // Fungsi hitungJumlahSimpul (Diperlukan untuk looping sort)
    public int hitungJumlahSimpul() {
        int jumlah = 0;
        Node bantu = head;
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }

    // METHOD BARU: Bubble Sort Teknik Tukar Heap (Pointer)
    // Variabel diterjemahkan: awal->head, akhir->tail, kanan->next, kiri->prev
    public void mengurutkanDataBubble_TeknikTukarHeap() {
        int N = hitungJumlahSimpul();
        Node bantu = head;
        System.out.println("Banyaknya simpul = " + hitungJumlahSimpul());

        for (int i = 1; i <= hitungJumlahSimpul(); i++) {
            // khusus menguji simpul pertama dgn sebelahnya
            if (head != null && head.next != null &&
head.nama.compareTo(head.next.nama) > 0) {
                bantu = head.next;       // bantu = awal.kanan
                head.next = bantu.next;  // awal.kanan = bantu.kanan

                // Cek null untuk menghindari error jika data hanya 2
                if (bantu.next != null) {
                    bantu.next.prev = head; // bantu.kanan.kiri = awal
                }

                bantu.next = head;       // bantu.kanan = awal
                bantu.prev = null;       // bantu.kiri = null
                head.prev = bantu;       // awal.kiri = bantu
                head = bantu;            // awal = bantu
            }

            // khusus menguji simpul kedua dgn sebelahnya, dst
            bantu = head; // bantu = awal

            while (bantu.next != tail && bantu.next != null) { // bantu.kanan !=
akhir
                Node A = bantu.next;      // simpul A
                Node B = bantu.next.next; // simpul B

                if (B != null && A.nama.compareTo(B.nama) > 0) {
                    // tukarkan simpul A dan simpul B
                    A.next = B.next; // A.kanan = B.kanan

                    if (B != tail) {
                        if (A.next != null) A.next.prev = A; // A.kanan.kiri = A
                    }
```

```
                    // Catatan: Baris ini dari kode asli Anda "B.kanan.kiri =
A;"
                    // namun karena B.kanan (sekarang A.next) sudah dihandle di
atas,
                    // dan struktur B akan berubah, kita ikuti alur pointer:
                    // B.next.prev = A (ini sama dengan A.next.prev = A yang
barusan)

                    // Lanjut logika tukar:
                    B.next = A;      // B.kanan = A
                    A.prev = B;      // A.kiri = B
                    bantu.next = B;  // bantu.kanan = B
                    B.prev = bantu;  // B.kiri = bantu

                    if (B == tail) tail = A; // if (B==akhir) akhir = A
                }
                bantu = bantu.next;
            }
            System.out.println("");
        }
        System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI=====");
    }
}

public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();

        int pilih;
        String nama, alamat, dicari;

        do {
            System.out.println("\nMENU MASTER:");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Tambah Tengah");
            System.out.println("4. Hapus");
            System.out.println("5. Cetak Maju");
            System.out.println("6. Cetak Mundur");
            System.out.println("7. Urutkan (Bubble Sort - Tukar Pointer/Heap)");
            System.out.println("8. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    dll.tambahDepan(nama, alamat);
                    break;
```

```
                case 2:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    dll.tambahBelakang(nama, alamat);
                    break;
                case 3:
                    System.out.print("Cari Nama : "); dicari = sc.nextLine();
                    System.out.print("Nama Baru : "); nama = sc.nextLine();
                    System.out.print("Alamat Baru: "); alamat = sc.nextLine();
                    dll.tambahTengah(dicari, nama, alamat);
                    break;
                case 4:
                    System.out.print("Nama Hapus: "); nama = sc.nextLine();
                    dll.hapus(nama);
                    break;
                case 5:
                    dll.cetakMaju();
                    break;
                case 6:
                    dll.cetakMundur();
                    break;
                case 7:
                    // Menjalankan method yang baru saja ditambahkan
                    dll.mengurutkanDataBubble_TeknikTukarHeap();
                    dll.cetakMaju(); // Tampilkan hasil
                    break;
            }

        } while (pilih != 8);
    }
}
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 1
Nama   : Fidel
Alamat : Kaltim

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 1
Nama   : Melia
Alamat : Kaltim

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 1
Nama   : Mian
Alamat : Sulsel
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 1
Nama   : Putra
Alamat : Kaltara

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 2
Nama   : Arlinda
Alamat : Medan

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 2
Nama   : Nadia
Alamat : Jogja
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 5

CETAK MAJU :
Putra - Kaltara
Mian - Sulsel
Melia - Kaltim
Fidel - Kaltim
Arlinda - Medan
Nadia - Jogja

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 4
Nama Hapus: Fidel
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 6

CETAK MUNDUR :
Nadia - Jogja
Arlinda - Medan
Melia - Kaltim
Mian - Sulsel
Putra - Kaltara

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer/Heap)
8. Keluar
Pilih: 7
Banyaknya simpul = 5
```

```
===PROSES PENGURUTAN BUBBLE SELESAI=====

CETAK MAJU :
Arlinda - Medan
Melia - Kaltim
Mian - Sulsel
Nadia - Jogja
Putra - Kaltara
```

Pembahasan : Program Java di atas merupakan aplikasi pengelolaan data nama dan alamat yang menggunakan struktur data **Double Linked List (senarai ganda)**, yang ditandai secara jelas oleh adanya dua pointer pada setiap simpul (next dan prev) serta penggunaan penunjuk head dan tail untuk mengelola data dari awal dan akhir. Program ini menyediakan operasi CRUD lengkap, yaitu tambah data di depan, belakang, dan tengah, hapus data, serta pencetakan data secara maju dan mundur yang menunjukkan kemampuan traversal dua arah—ciri utama double linked list. Selain itu, program mengintegrasikan algoritma **Bubble Sort dengan teknik tukar pointer (heap/node swapping)** untuk mengurutkan data berdasarkan nama tanpa menukar isi data, melainkan dengan memanipulasi hubungan antar simpul. Dengan menu interaktif berbasis console, program ini menunjukkan penerapan konsep double linked list secara menyeluruh, baik dari sisi manipulasi struktur, navigasi dua arah, maupun pengurutan data.

Program 11.4

```
import java.util.Scanner;

class Node {
    String nama, alamat;
    Node next, prev; // next = kanan, prev = kiri

    public Node(String nama, String alamat) {
        this.nama = nama;
        this.alamat = alamat;
        next = null;
        prev = null;
    }
}

class DoubleLinkedList {
    Node head, tail; // head = awal, tail = akhir

    public DoubleLinkedList() {
        head = tail = null;
    }

    // --- FUNGSI CRUD STANDAR ---

    // Tambah Depan
    void tambahDepan(String nama, String alamat) {
        Node baru = new Node(nama, alamat);
        if (head == null) {
            head = tail = baru;
        } else {
```

```
            baru.next = head;
            head.prev = baru;
            head = baru;
        }
    }

    // Tambah Belakang
    void tambahBelakang(String nama, String alamat) {
        Node baru = new Node(nama, alamat);
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
            tail = baru;
        }
    }

    // Tambah Tengah
    void tambahTengah(String dicari, String nama, String alamat) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(dicari)) {
            bantu = bantu.next;
        }

        if (bantu == null) {
            System.out.println("Data yang dicari tidak ditemukan!");
            return;
        }

        Node baru = new Node(nama, alamat);
        Node sesudah = bantu.next;

        bantu.next = baru;
        baru.prev = bantu;

        if (sesudah != null) {
            baru.next = sesudah;
            sesudah.prev = baru;
        } else {
            tail = baru;
        }
    }

    // Hapus
    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(nama)) {
            bantu = bantu.next;
        }

        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
```

```
            return;
        }

        if (bantu == head && bantu == tail) {
           head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
    }

    // Cetak Maju
    void cetakMaju() {
        Node bantu = head;
        System.out.println("\nCETAK MAJU :");
        while (bantu != null) {
            System.out.println(bantu.nama + " - " + bantu.alamat);
            bantu = bantu.next;
        }
    }

    // Cetak Mundur
    void cetakMundur() {
        Node bantu = tail;
        System.out.println("\nCETAK MUNDUR :");
        while (bantu != null) {
            System.out.println(bantu.nama + " - " + bantu.alamat);
            bantu = bantu.prev;
        }
    }

    // Fungsi hitungJumlahSimpul (Pendukung Sorting)
    public int hitungJumlahSimpul() {
        int jumlah = 0;
        Node bantu = head;
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }

    // --- FITUR SORTING (BUBBLE SORT - TUKAR POINTER) ---
    public void mengurutkanDataBubble_TeknikTukarHeap() {
        int N = hitungJumlahSimpul();
        Node bantu = head;
        System.out.println("Banyaknya simpul = " + hitungJumlahSimpul());
```

```
        for (int i = 1; i <= hitungJumlahSimpul(); i++) {
            // Khusus menguji simpul pertama dgn sebelahnya
            if (head != null && head.next != null &&
head.nama.compareTo(head.next.nama) > 0) {
                bantu = head.next;
                head.next = bantu.next;
                if (bantu.next != null) {
                    bantu.next.prev = head;
                }
                bantu.next = head;
                bantu.prev = null;
                head.prev = bantu;
                head = bantu;
            }

            // Khusus menguji simpul kedua dgn sebelahnya, dst
            bantu = head;
            while (bantu.next != tail && bantu.next != null) {
                Node A = bantu.next;
                Node B = bantu.next.next;

                if (B != null && A.nama.compareTo(B.nama) > 0) {
                    // Tukarkan simpul A dan simpul B
                    A.next = B.next;
                    if (B != tail) {
                        if (A.next != null) A.next.prev = A;
                    }
                    B.next = A;
                    A.prev = B;
                    bantu.next = B;
                    B.prev = bantu;
                    if (B == tail) tail = A;
                }
                bantu = bantu.next;
            }
            System.out.println("");
        }
        System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI=====");
    }

    // --- FITUR SEARCHING (LINEAR SEARCH - BARU DITAMBAHKAN) ---
    public void cariLinear() {
        if (head == null) { // jika senarai masih kosong
            System.out.print("....MAAF SENARAI KOSONG....");
        } else { // jika senarai tidak kosong
            Scanner masukan = new Scanner(System.in);
            System.out.print("Silakan masukkan nama yang anda cari : ");
            String NAMACARI = masukan.nextLine();
            boolean statusKetemu = false;
            int i = 0;
            int posisiKetemu = -1;
            Node bantu;
```

```
            bantu = head;
            while (bantu != null) {
                if (NAMACARI.equals(bantu.nama)) {
                    statusKetemu = true;
                    posisiKetemu = i;
                }
                bantu = bantu.next;
                i++;
            }
            System.out.println("Status Ketemu = " + statusKetemu + " di posisi
ke " + posisiKetemu);
        }
    }
}

public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();

        int pilih;
        String nama, alamat, dicari;

        do {
            System.out.println("\nMENU MASTER:");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Tambah Tengah");
            System.out.println("4. Hapus");
            System.out.println("5. Cetak Maju");
            System.out.println("6. Cetak Mundur");
            System.out.println("7. Urutkan (Bubble Sort - Tukar Pointer)");
            System.out.println("8. Cari Data (Linear Search)");
            System.out.println("9. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    dll.tambahDepan(nama, alamat);
                    break;
                case 2:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    dll.tambahBelakang(nama, alamat);
                    break;
                case 3:
                    System.out.print("Cari Nama : "); dicari = sc.nextLine();
                    System.out.print("Nama Baru : "); nama = sc.nextLine();
                    System.out.print("Alamat Baru: "); alamat = sc.nextLine();
```

```
                    dll.tambahTengah(dicari, nama, alamat);
                    break;
                case 4:
                    System.out.print("Nama Hapus: "); nama = sc.nextLine();
                    dll.hapus(nama);
                    break;
                case 5:
                    dll.cetakMaju();
                    break;
                case 6:
                    dll.cetakMundur();
                    break;
                case 7:
                    dll.mengurutkanDataBubble_TeknikTukarHeap();
                    dll.cetakMaju();
                    break;
                case 8: // Menu baru untuk Cari Linear
                    dll.cariLinear();
                    break;
            }

        } while (pilih != 9);
    }
}
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 1
Nama   : Fidel
Alamat : Kaltim

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 2
Nama   : Melia
Alamat : Kaltim

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 1
Nama   : Mian
Alamat : Sulsel
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 2
Nama   : Putra
Alamat : Kaltara

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 2
Nama   : Arlinda
Alamat : Medan

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 3
Cari Nama : Fidel
Nama Baru : Dewi
Alamat Baru: Ambon
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 5

CETAK MAJU :
Mian - Sulsel
Fidel - Kaltim
Dewi - Ambon
Melia - Kaltim
Putra - Kaltara
Arlinda - Medan

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 4
Nama Hapus: Fidel
```

```
MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 6

CETAK MUNDUR :
Arlinda - Medan
Putra - Kaltara
Melia - Kaltim
Dewi - Ambon
Mian - Sulsel

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 7
Banyaknya simpul = 5
```

```
===PROSES PENGURUTAN BUBBLE SELESAI=====

CETAK MAJU :
Arlinda - Medan
Dewi - Ambon
Melia - Kaltim
Mian - Sulsel
Putra - Kaltara

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 8
Silakan masukkan nama yang anda cari : Melia
Status Ketemu = true di posisi ke 2

MENU MASTER:
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan (Bubble Sort - Tukar Pointer)
8. Cari Data (Linear Search)
9. Keluar
Pilih: 9
Press any key to continue . . .
```

Pembahasan : Program Java di atas merupakan aplikasi pengelolaan data nama dan alamat yang menerapkan struktur data **Double Linked List (senarai ganda)**, yang ditunjukkan oleh penggunaan dua pointer pada setiap simpul, yaitu next (kanan) dan prev (kiri), serta adanya penunjuk head dan tail untuk mengakses data dari awal dan akhir. Program ini menyediakan operasi CRUD lengkap seperti tambah data di depan, belakang, dan tengah, hapus data, serta pencetakan data secara maju dan mundur yang membuktikan kemampuan traversal dua arah khas double linked list. Selain itu, program dilengkapi dengan algoritma **Bubble Sort menggunakan teknik tukar pointer (heap/node swapping)** untuk mengurutkan data berdasarkan nama tanpa menukar isi simpul, serta **Linear Search** untuk mencari data secara berurutan. Dengan menu interaktif berbasis console, program ini menunjukkan implementasi yang komprehensif dari konsep double linked list, mencakup manipulasi struktur, pengurutan, dan pencarian data.

LATIHAN

-

D. TUGAS

Memodifikasi program yang ada (praktik)

Mengurutkan data berdasarkan data numerik (umur / ipk) bebas secara ascending maupun descending

1. tukar nilai
2. tukar heap data numerik

Program 11.1

```
import java.util.Scanner;

class Node {
    String nama, alamat, jekel;
    int umur;
    double ipk;
    Node next, prev;

    public Node(String nama, String alamat, int umur, String jekel, double ipk)
{
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.jekel = jekel;
        this.ipk = ipk;
        next = null;
        prev = null;
    }
}

class DoubleLinkedList {
    Node head, tail;

    public DoubleLinkedList() {
        head = tail = null;
    }

    // --- FITUR UTAMA (CRUD) ---

    void tambahDepan(String nama, String alamat, int umur, String jekel, double
ipk) {
        Node baru = new Node(nama, alamat, umur, jekel, ipk);
        if (head == null) {
            head = tail = baru;
        } else {
            baru.next = head;
            head.prev = baru;
            head = baru;
        }
        System.out.println("Data " + nama + " berhasil ditambah di depan.");
    }

    void tambahBelakang(String nama, String alamat, int umur, String jekel, double
ipk) {
        Node baru = new Node(nama, alamat, umur, jekel, ipk);
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
```

```
            tail = baru;
        }
        System.out.println("Data " + nama + " berhasil ditambah di belakang.");
    }

    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equalsIgnoreCase(nama)) {
            bantu = bantu.next;
        }

        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
            return;
        }

        if (bantu == head && bantu == tail) {
            head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
        System.out.println("Data " + nama + " berhasil dihapus.");
    }

    void cetakMaju() {
        Node bantu = head;
        System.out.println("\n--- CETAK DATA (MAJU) ---");
        if (head == null) System.out.println("Data Kosong");
        else {
            System.out.println("-----------------------------------------------
------------------------------");
            System.out.printf("%-15s | %-15s | %-5s | %-5s | %-5s\n", "NAMA",
"ALAMAT", "UMUR", "L/P", "IPK");
            System.out.println("-----------------------------------------------
------------------------------");
            while (bantu != null) {
                System.out.printf("%-15s | %-15s | %-5d | %-5s | %.2f\n",
                    bantu.nama,    bantu.alamat,    bantu.umur,    bantu.jekel,
bantu.ipk);
                bantu = bantu.next;
            }
            System.out.println("-----------------------------------------------
------------------------------");
        }
    }
```

```
    // --- FITUR PENDUKUNG ---

    public int hitungJumlahSimpul() {
        int jumlah = 0;
        Node bantu = head;
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }

    // --- 1. SORTING TUKAR NILAI (IPK) ---

    public void tukarNilai(Node X, Node Y) {
        String tempNama = X.nama;
        String tempAlamat = X.alamat;
        int tempUmur = X.umur;
        String tempJekel = X.jekel;
        double tempIpk = X.ipk;

        X.nama = Y.nama;
        X.alamat = Y.alamat;
        X.umur = Y.umur;
        X.jekel = Y.jekel;
        X.ipk = Y.ipk;

        Y.nama = tempNama;
        Y.alamat = tempAlamat;
        Y.umur = tempUmur;
        Y.jekel = tempJekel;
        Y.ipk = tempIpk;
    }

    // Parameter 'ascending': true untuk kecil->besar, false untuk besar->kecil
    public void urutkanNilai_IPK(boolean ascending) {
        if (head == null || head.next == null) {
            System.out.println("Data tidak cukup untuk diurutkan.");
            return;
        }

        String urutan = ascending ? "Ascending (Kecil ke Besar)" : "Descending
(Besar ke Kecil)";
        System.out.println("\nMengurutkan IPK dengan TUKAR NILAI secara " + urutan
+ "...");

        int N = hitungJumlahSimpul();
        Node A, B;

        for (int i = 0; i < N - 1; i++) {
            A = head;
```

```
            B = head.next;
            while (B != null) {
                boolean kondisiTukar = false;

                if (ascending) {
                    if (A.ipk > B.ipk) kondisiTukar = true;
                } else {
                    if (A.ipk < B.ipk) kondisiTukar = true;
                }

                if (kondisiTukar) {
                    tukarNilai(A, B);
                }
                A = A.next;
                B = B.next;
            }
        }
        System.out.println("=== PENGURUTAN IPK SELESAI ===");
    }

    // --- 2. SORTING TUKAR HEAP/POINTER (UMUR) ---

    public void urutkanHeap_Umur(boolean ascending) {
        if (head == null || head.next == null) {
            System.out.println("Data tidak cukup untuk diurutkan.");
            return;
        }

        String urutan = ascending ? "Ascending (Muda ke Tua)" : "Descending (Tua
ke Muda)";
        System.out.println("\nMengurutkan UMUR dengan TUKAR HEAP/POINTER secara
" + urutan + "...");

        int N = hitungJumlahSimpul();
        Node bantu;

        for (int i = 1; i <= N; i++) {
            // A. Cek Head dengan Next-nya
            boolean kondisiHead = false;
            if (head != null && head.next != null) {
                if (ascending) {
                    if (head.umur > head.next.umur) kondisiHead = true;
                } else {
                    if (head.umur < head.next.umur) kondisiHead = true;
                }
            }

            if (kondisiHead) {
                bantu = head.next;
                head.next = bantu.next;
                if (bantu.next != null) {
                    bantu.next.prev = head;
```

```
                }
                bantu.next = head;
                bantu.prev = null;
                head.prev = bantu;
                head = bantu; // Update Head global
            }

            // B. Cek Node selain Head
            bantu = head;
            while (bantu.next != tail && bantu.next != null) {
                Node A = bantu.next;
                Node B = bantu.next.next;
                boolean kondisiTukar = false;

                if (B != null) {
                    if (ascending) {
                        if (A.umur > B.umur) kondisiTukar = true;
                    } else {
                        if (A.umur < B.umur) kondisiTukar = true;
                    }
                }

                if (kondisiTukar) {
                    // Tukar Pointer A dan B
                    A.next = B.next;
                    if (B != tail) {
                        if (A.next != null) A.next.prev = A;
                    }
                    B.next = A;
                    A.prev = B;
                    bantu.next = B;
                    B.prev = bantu;
                    if (B == tail) tail = A; // Update Tail global
                }
                bantu = bantu.next;
            }
        }
        System.out.println("=== PENGURUTAN UMUR SELESAI ===");
    }

    // --- PENCARIAN ---

    public void cariLinearSearch(String keyword) {
        Node bantu = head;
        boolean ditemukan = false;
        int posisi = 1;

        System.out.println("\n--- HASIL PENCARIAN LINEAR ---");
        while (bantu != null) {
            if (bantu.nama.equalsIgnoreCase(keyword)) {
                System.out.println("DITEMUKAN pada posisi ke-" + posisi);
                System.out.println("Nama   : " + bantu.nama);
```

```
                System.out.println("Alamat : " + bantu.alamat);
                System.out.println("Umur   : " + bantu.umur);
                System.out.println("IPK    : " + bantu.ipk);
                ditemukan = true;
                break;
            }
            bantu = bantu.next;
            posisi++;
        }

        if (!ditemukan) {
            System.out.println("Data dengan nama '" + keyword + "' tidak
ditemukan.");
        }
    }
}

public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();

        int pilih, subPilih, umur;
        double ipk;
        String nama, alamat, jekel, dicari;

        do {
            System.out.println("\n=== MENU PROGRAM MASTER (NUMERIK) ===");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Hapus Data");
            System.out.println("4. Cetak Data");
            System.out.println("5. Urutkan IPK (Tukar Nilai)");
            System.out.println("6. Urutkan UMUR (Tukar Heap/Pointer)");
            System.out.println("7. Cari Data Nama (Linear Search)");
            System.out.println("8. Keluar");
            System.out.print("Pilih menu: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                case 2:
                    System.out.print("Nama    : "); nama = sc.nextLine();
                    System.out.print("Alamat  : "); alamat = sc.nextLine();
                    System.out.print("Umur    : "); umur = sc.nextInt();
                    sc.nextLine();
                    System.out.print("L/P     : "); jekel = sc.nextLine();
                    System.out.print("IPK     : "); ipk = sc.nextDouble();

                    if (pilih == 1) dll.tambahDepan(nama, alamat, umur, jekel,
ipk);
```

```
                else dll.tambahBelakang(nama, alamat, umur, jekel, ipk);
                break;

            case 3:
                System.out.print("Masukkan nama yang dihapus: ");
                nama = sc.nextLine();
                dll.hapus(nama);
                break;

            case 4:
                dll.cetakMaju();
                break;

            case 5: // TUKAR NILAI - IPK
                System.out.println("\n--- Opsi Pengurutan IPK ---");
                System.out.println("1. Ascending (Kecil -> Besar)");
                System.out.println("2. Descending (Besar -> Kecil)");
                System.out.print("Pilih: ");
                subPilih = sc.nextInt();
                if (subPilih == 1) dll.urutkanNilai_IPK(true);
                else if (subPilih == 2) dll.urutkanNilai_IPK(false);
                else System.out.println("Pilihan salah.");

                dll.cetakMaju();
                break;

            case 6: // TUKAR HEAP - UMUR
                System.out.println("\n--- Opsi Pengurutan UMUR ---");
                System.out.println("1. Ascending (Muda -> Tua)");
                System.out.println("2. Descending (Tua -> Muda)");
                System.out.print("Pilih: ");
                subPilih = sc.nextInt();
                if (subPilih == 1) dll.urutkanHeap_Umur(true);
                else if (subPilih == 2) dll.urutkanHeap_Umur(false);
                else System.out.println("Pilihan salah.");

                dll.cetakMaju();
                break;

            case 7:
                System.out.print("Masukkan nama yang dicari: ");
                dicari = sc.nextLine();
                dll.cariLinearSearch(dicari);
                break;

            case 8:
                System.out.println("Terima kasih.");
                break;

            default:
                System.out.println("Pilihan salah.");
        }
```

```
        } while (pilih != 8);
    }
}
```

```
=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 1
Nama    : Fidelia
Alamat  : Kaltim
Umur    : 19
L/P     : P
IPK     : 4.0
Data Fidelia berhasil ditambah di depan.

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 1
Nama    : Melia
Alamat  : kaltim
Umur    : 19
L/P     : P
IPK     : 3.9
Data Melia berhasil ditambah di depan.

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 1
Nama    : Mian
Alamat  : Sulsel
Umur    : 19
L/P     : P
```

```
IPK     : 3.8
Data Mian berhasil ditambah di depan.

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 1
Nama    : Nadia
Alamat  : Jogja
Umur    : 20
L/P     : P
IPK     : 4.0
Data Nadia berhasil ditambah di depan.

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 2
Nama    : Putra
Alamat  : Kaltara
Umur    : 19
L/P     : L
IPK     : 3.7
Data Putra berhasil ditambah di belakang.

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 2
Nama    : Rama
Alamat  : Solo
```

```
Umur    : 19
L/P     : L
IPK     : 3.5
Data Rama berhasil ditambah di belakang.

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 4

--- CETAK DATA (MAJU) ---
--------------------------------------------------------------------------------
NAMA            | ALAMAT          | UMUR  | L/P   | IPK
--------------------------------------------------------------------------------
Nadia           | Jogja           | 20    | P     | 4.00
Mian            | Sulsel          | 19    | P     | 3.80
Melia           | kaltim          | 19    | P     | 3.90
Fidelia         | Kaltim          | 19    | P     | 4.00
Putra           | Kaltara         | 19    | L     | 3.70
Rama            | Solo            | 19    | L     | 3.50
--------------------------------------------------------------------------------

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 3
Masukkan nama yang dihapus: Fidelia
Data Fidelia berhasil dihapus.

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
```

```
8. Keluar
Pilih menu: 5

--- Opsi Pengurutan IPK ---
1. Ascending (Kecil -> Besar)
2. Descending (Besar -> Kecil)
Pilih: 2

Mengurutkan IPK dengan TUKAR NILAI secara Descending (Besar ke Kecil)...
=== PENGURUTAN IPK SELESAI ===

--- CETAK DATA (MAJU) ---
--------------------------------------------------------------------------------
NAMA            | ALAMAT          | UMUR  | L/P   | IPK
--------------------------------------------------------------------------------
Nadia           | Jogja           | 20    | P     | 4.00
Melia           | kaltim          | 19    | P     | 3.90
Mian            | Sulsel          | 19    | P     | 3.80
Putra           | Kaltara         | 19    | L     | 3.70
Rama            | Solo            | 19    | L     | 3.50
--------------------------------------------------------------------------------

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 6

--- Opsi Pengurutan UMUR ---
1. Ascending (Muda -> Tua)
2. Descending (Tua -> Muda)
Pilih: 1

Mengurutkan UMUR dengan TUKAR HEAP/POINTER secara Ascending (Muda ke Tua)...
=== PENGURUTAN UMUR SELESAI ===

--- CETAK DATA (MAJU) ---
--------------------------------------------------------------------------------
NAMA            | ALAMAT          | UMUR  | L/P   | IPK
--------------------------------------------------------------------------------
Melia           | kaltim          | 19    | P     | 3.90
Mian            | Sulsel          | 19    | P     | 3.80
Putra           | Kaltara         | 19    | L     | 3.70
Rama            | Solo            | 19    | L     | 3.50
```

```
Nadia           | Jogja           | 20    | P     | 4.00
--------------------------------------------------------------------------------

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 7
Masukkan nama yang dicari: Rama

--- HASIL PENCARIAN LINEAR ---
DITEMUKAN pada posisi ke-4
Nama   : Rama
Alamat : Solo
Umur   : 19
IPK    : 3.5

=== MENU PROGRAM MASTER (NUMERIK) ===
1. Tambah Depan
2. Tambah Belakang
3. Hapus Data
4. Cetak Data
5. Urutkan IPK (Tukar Nilai)
6. Urutkan UMUR (Tukar Heap/Pointer)
7. Cari Data Nama (Linear Search)
8. Keluar
Pilih menu: 8
Terima kasih.
Press any key to continue . . .
```

Pembahasan : Program ini merupakan aplikasi pengelolaan data mahasiswa yang menggunakan struktur data Double Linked List untuk menyimpan dan mengelola data berupa nama, alamat, umur, jenis kelamin, dan IPK. Setiap data disimpan dalam sebuah node yang memiliki dua penunjuk, yaitu next untuk mengakses data berikutnya dan prev untuk mengakses data sebelumnya, sehingga memungkinkan proses penelusuran data secara dua arah. Program menyediakan fitur CRUD yang meliputi penambahan data di depan dan di belakang, penghapusan data, serta pencetakan data yang ditampilkan dalam bentuk tabel. Seluruh operasi tersebut dikelola melalui kelas DoubleLinkedList dengan bantuan pointer head sebagai penunjuk awal dan tail sebagai penunjuk akhir. Selain itu, program mengimplementasikan dua jenis pengurutan data, yaitu pengurutan IPK menggunakan teknik tukar nilai yang menukar isi data antar node tanpa mengubah posisi node, serta pengurutan umur menggunakan teknik tukar heap atau pointer yang menukar posisi node dengan memanipulasi hubungan antar node. Kedua metode pengurutan tersebut dapat dijalankan secara ascending maupun descending. Program juga dilengkapi dengan fitur pencarian data menggunakan metode linear search yang bekerja dengan menelusuri node satu per satu dari awal hingga data ditemukan. Dengan menu interaktif berbasis console, program ini memperlihatkan penerapan konsep double linked list, sorting, dan searching secara terintegrasi dan mudah dipahami.

Program 11.2

```
import java.util.Scanner;
class Node {
    String nama, alamat;
    int umur;
    double ipk;
    Node next, prev;
```

```
    public Node(String nama, String alamat, int umur, double ipk) {
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.ipk = ipk;
        next = null;
        prev = null;
    }
}
class DoubleLinkedList {
    Node head, tail;
    public DoubleLinkedList() {
        head = tail = null;
    }
    void tambahDepan(String nama, String alamat, int umur, double ipk) {
        Node baru = new Node(nama, alamat, umur, ipk);
        if (head == null) {
            head = tail = baru;
        } else {
            baru.next = head;
            head.prev = baru;
            head = baru;
        }
    }
    void tambahBelakang(String nama, String alamat, int umur, double ipk) {
        Node baru = new Node(nama, alamat, umur, ipk);
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
            tail = baru;
        }
    }
    void tambahTengah(String dicari, String nama, String alamat, int umur, double
ipk) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equalsIgnoreCase(dicari)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data induk tidak ditemukan!");
            return;
        }
        Node baru = new Node(nama, alamat, umur, ipk);
        Node sesudah = bantu.next;
        bantu.next = baru;
        baru.prev = bantu;
        if (sesudah != null) {
            baru.next = sesudah;
            sesudah.prev = baru;
        } else {
```

```
            tail = baru;
        }
    }
    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equalsIgnoreCase(nama)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
            return;
        }
        if (bantu == head && bantu == tail) {
            head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
    }
    void cetakMaju() {
        Node bantu = head;
        System.out.println("\n--- CETAK MAJU ---");
        System.out.printf("%-15s %-15s %-5s %-5s\n", "NAMA", "ALAMAT", "UMUR",
"IPK");
        System.out.println("-------------------------------------------------");
        while (bantu != null) {
            System.out.printf("%-15s   %-15s   %-5d   %-5.2f\n",   bantu.nama,
bantu.alamat, bantu.umur, bantu.ipk);
            bantu = bantu.next;
        }
    }
    void cetakMundur() {
        Node bantu = tail;
        System.out.println("\n--- CETAK MUNDUR ---");
        System.out.printf("%-15s %-15s %-5s %-5s\n", "NAMA", "ALAMAT", "UMUR",
"IPK");
        System.out.println("-------------------------------------------------");
        while (bantu != null) {
            System.out.printf("%-15s   %-15s   %-5d   %-5.2f\n",   bantu.nama,
bantu.alamat, bantu.umur, bantu.ipk);
            bantu = bantu.prev;
        }
    }
    public int hitungJumlahSimpul() {
        int jumlah = 0;
        Node bantu = head;
```

```
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }
    public void sortTukarNilai(boolean byIpk, boolean ascending) {
        int N = hitungJumlahSimpul();
        if (N < 2) return;
        System.out.println("\nProses Sorting (Tukar Nilai)...");
        Node A, B;
        for (int i = 0; i < N - 1; i++) {
            A = head;
            B = head.next;
            while (B != null) {
                boolean tukar = false;
                double valA = byIpk ? A.ipk : A.umur;
                double valB = byIpk ? B.ipk : B.umur;
                if (ascending) {
                    if (valA > valB) tukar = true;
                } else {
                    if (valA < valB) tukar = true;
                }
                if (tukar) {
                    String tNama = A.nama; A.nama = B.nama; B.nama = tNama;
                    String tAlamat = A.alamat; A.alamat = B.alamat; B.alamat =
tAlamat;
                    int tUmur = A.umur; A.umur = B.umur; B.umur = tUmur;
                    double tIpk = A.ipk; A.ipk = B.ipk; B.ipk = tIpk;
                }
                A = A.next;
                B = B.next;
            }
        }
    }
    public void sortTukarHeap(boolean byIpk, boolean ascending) {
        int N = hitungJumlahSimpul();
        if (N < 2) return;
        System.out.println("\nProses Sorting (Tukar Heap/Pointer)...");
        for (int i = 1; i <= N; i++) {
            boolean tukarHead = false;
            if (head != null && head.next != null) {
                double valA = byIpk ? head.ipk : head.umur;
                double valB = byIpk ? head.next.ipk : head.next.umur;
                if (ascending) {
                    if (valA > valB) tukarHead = true;
                } else {
                    if (valA < valB) tukarHead = true;
                }
            }
            if (tukarHead) {
                Node bantu = head.next;
```

```
                head.next = bantu.next;
                if (bantu.next != null) bantu.next.prev = head;
                bantu.next = head;
                bantu.prev = null;
                head.prev = bantu;
                head = bantu;
            }
            Node bantu = head;
            while (bantu.next != tail && bantu.next != null) {
                Node A = bantu.next;
                Node B = bantu.next.next;
                boolean tukarNode = false;
                if (B != null) {
                    double valA = byIpk ? A.ipk : A.umur;
                    double valB = byIpk ? B.ipk : B.umur;
                    if (ascending) {
                        if (valA > valB) tukarNode = true;
                    } else {
                        if (valA < valB) tukarNode = true;
                    }
                }
                if (tukarNode) {
                    A.next = B.next;
                    if (B != tail) {
                        if (A.next != null) A.next.prev = A;
                    }
                    B.next = A;
                    A.prev = B;
                    bantu.next = B;
                    B.prev = bantu;
                    if (B == tail) tail = A;
                }
                bantu = bantu.next;
            }
        }
    }
}
public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();
        int pilih, subPilih, orderPilih, umur;
        double ipk;
        String nama, alamat, dicari;
        boolean isAscending;
        do {
            System.out.println("\n=== MENU UTAMA ===");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Tambah Tengah");
            System.out.println("4. Hapus Data");
            System.out.println("5. Cetak Maju");
```

```
            System.out.println("6. Cetak Mundur");
            System.out.println("7. SORTING (Tukar Nilai & Heap)");
            System.out.println("8. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();
            switch (pilih) {
                case 1:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    System.out.print("Umur   : "); umur = sc.nextInt();
                    System.out.print("IPK    : "); ipk = sc.nextDouble();
                    dll.tambahDepan(nama, alamat, umur, ipk);
                    break;
                case 2:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    System.out.print("Umur   : "); umur = sc.nextInt();
                    System.out.print("IPK    : "); ipk = sc.nextDouble();
                    dll.tambahBelakang(nama, alamat, umur, ipk);
                    break;
                case 3:
                    System.out.print("Cari Nama : "); dicari = sc.nextLine();
                    System.out.print("Nama Baru : "); nama = sc.nextLine();
                    System.out.print("Alamat    : "); alamat = sc.nextLine();
                    System.out.print("Umur Baru : "); umur = sc.nextInt();
                    System.out.print("IPK Baru  : "); ipk = sc.nextDouble();
                    dll.tambahTengah(dicari, nama, alamat, umur, ipk);
                    break;
                case 4:
                    System.out.print("Nama Hapus: "); nama = sc.nextLine();
                    dll.hapus(nama);
                    break;
                case 5:
                    dll.cetakMaju();
                    break;
                case 6:
                    dll.cetakMundur();
                    break;
                case 7:
                    System.out.println("\n--- MENU SORTING ---");
                    System.out.println("Pilih Teknik:");
                    System.out.println("1. Tukar NILAI (Value Swap)");
                    System.out.println("2. Tukar HEAP (Pointer Swap)");
                    System.out.print("Pilih Teknik: ");
                    int teknik = sc.nextInt();
                    System.out.println("\nPilih Kriteria Data:");
                    System.out.println("1. Berdasarkan UMUR");
                    System.out.println("2. Berdasarkan IPK");
                    System.out.print("Pilih Kriteria: ");
                    subPilih = sc.nextInt();
                    boolean byIpk = (subPilih == 2);
```

```
                    System.out.println("\nPilih Urutan:");
                    System.out.println("1. Ascending (Kecil -> Besar)");
                    System.out.println("2. Descending (Besar -> Kecil)");
                    System.out.print("Pilih Urutan: ");
                    orderPilih = sc.nextInt();
                    isAscending = (orderPilih == 1);
                    if (teknik == 1) {
                        dll.sortTukarNilai(byIpk, isAscending);
                    } else if (teknik == 2) {
                        dll.sortTukarHeap(byIpk, isAscending);
                    } else {
                        System.out.println("Pilihan teknik salah!");
                    }
                    dll.cetakMaju();
                    break;
            }
        } while (pilih != 8);
    }
}
```

```
=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 1
Nama   : Fidelia
Alamat : Kaltim
Umur   : 19
IPK    : 4.0

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 1
Nama   : Melia
Alamat : Kaltim
Umur   : 19
IPK    : 3.9

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 1
Nama   : Mian
Alamat : Sulsel
Umur   : 19
```

```
IPK    : 3.9

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 1
Nama   : Mian
Alamat : Sulsel
Umur   : 19
IPK    : 3.9

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 1
Nama   : Putra
Alamat : Kaltara
Umur   : 19
IPK    : 3.8

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 2
Nama   : Rama
Alamat : Solo
```

```
Umur   : 19
IPK    : 3.6

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 2
Nama   : Nadia
Alamat : Jogja
Umur   : 20
IPK    : 4.0

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 2
Nama   : Melia
Alamat : Ambon
Umur   : 19
IPK    : 3.8

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 3
Cari Nama : Putra
```

```
Nama Baru : Dewi
Alamat    : Ambon
Umur Baru : 18
IPK Baru  : 4.0

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 4
Nama Hapus: Melia

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 5

--- CETAK MAJU ---
NAMA            ALAMAT          UMUR  IPK
-------------------------------------------------
Putra           Kaltara         19    3.80
Dewi            Ambon           18    4.00
Mian            Sulsel          19    3.90
Fidelia         Kaltim          19    4.00
Rama            Solo            19    3.60
Nadia           Jogja           20    4.00
Melia           Ambon           19    3.80

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
```

```
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 6

--- CETAK MUNDUR ---
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
Melia           Ambon           19    3.80
Nadia           Jogja           20    4.00
Rama            Solo            19    3.60
Fidelia         Kaltim          19    4.00
Mian            Sulsel          19    3.90
Dewi            Ambon           18    4.00
Putra           Kaltara         19    3.80

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 7

--- MENU SORTING ---
Pilih Teknik:
1. Tukar NILAI (Value Swap)
2. Tukar HEAP (Pointer Swap)
Pilih Teknik: 1

Pilih Kriteria Data:
1. Berdasarkan UMUR
2. Berdasarkan IPK
Pilih Kriteria: 1

Pilih Urutan:
1. Ascending (Kecil -> Besar)
2. Descending (Besar -> Kecil)
Pilih Urutan: 1
```

```
Proses Sorting (Tukar Nilai)...

--- CETAK MAJU ---
NAMA            ALAMAT          UMUR  IPK
-------------------------------------------------
Dewi            Ambon           18    4.00
Putra           Kaltara         19    3.80
Mian            Sulsel          19    3.90
Fidelia         Kaltim          19    4.00
Rama            Solo            19    3.60
Melia           Ambon           19    3.80
Nadia           Jogja           20    4.00

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 7

--- MENU SORTING ---
Pilih Teknik:
1. Tukar NILAI (Value Swap)
2. Tukar HEAP (Pointer Swap)
Pilih Teknik: 2

Pilih Kriteria Data:
1. Berdasarkan UMUR
2. Berdasarkan IPK
Pilih Kriteria: 2

Pilih Urutan:
1. Ascending (Kecil -> Besar)
2. Descending (Besar -> Kecil)
Pilih Urutan: 2

Proses Sorting (Tukar Heap/Pointer)...

--- CETAK MAJU ---
NAMA            ALAMAT          UMUR  IPK
```

```
-----------------------------------------------
Dewi            Ambon           18    4.00
Fidelia         Kaltim          19    4.00
Nadia           Jogja           20    4.00
Mian            Sulsel          19    3.90
Putra           Kaltara         19    3.80
Melia           Ambon           19    3.80
Rama            Solo            19    3.60

=== MENU UTAMA ===
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus Data
5. Cetak Maju
6. Cetak Mundur
7. SORTING (Tukar Nilai & Heap)
8. Keluar
Pilih: 8
Press any key to continue . . .
```

Pembahasan : Program ini merupakan aplikasi pengelolaan data mahasiswa yang menggunakan struktur data double linked list untuk menyimpan informasi berupa nama, alamat, umur, dan IPK. Data disimpan dalam class Node yang memiliki dua pointer, yaitu next dan prev, sehingga setiap simpul dapat diakses secara maju maupun mundur. Class DoubleLinkedList berfungsi sebagai pengelola utama data dengan menyediakan operasi penambahan data di depan, belakang, dan tengah berdasarkan nama tertentu, penghapusan data berdasarkan nama, serta pencetakan data dari awal ke akhir dan sebaliknya. Program juga mengimplementasikan dua metode pengurutan data, yaitu pengurutan dengan cara tukar nilai yang menukar isi atribut antar node tanpa mengubah posisi node, serta pengurutan dengan cara tukar heap atau pointer yang menukar posisi node dengan mengatur ulang hubungan antar node. Pengurutan dapat dilakukan berdasarkan umur atau IPK dengan pilihan urutan menaik atau menurun. Seluruh fitur tersebut dijalankan melalui menu interaktif pada method main yang menerima input pengguna menggunakan Scanner, sehingga pengguna dapat mengelola data mahasiswa secara dinamis sekaligus memahami cara kerja linked list, proses sorting, dan manipulasi pointer dalam satu program yang terintegrasi.

Program 11.2

```
import java.util.Scanner;

class Node {
    String nama, alamat;
    int umur;       // Tambahan atribut numerik
    double ipk;     // Tambahan atribut numerik
    Node next, prev; // next = kanan, prev = kiri

    public Node(String nama, String alamat, int umur, double ipk) {
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.ipk = ipk;
        next = null;
```

```
        prev = null;
    }
}

class DoubleLinkedList {
    Node head, tail; // head = awal, tail = akhir

    public DoubleLinkedList() {
        head = tail = null;
    }

    // --- FUNGSI CRUD (Disesuaikan dengan Umur & IPK) ---

    void tambahDepan(String nama, String alamat, int umur, double ipk) {
        Node baru = new Node(nama, alamat, umur, ipk);
        if (head == null) {
            head = tail = baru;
        } else {
            baru.next = head;
            head.prev = baru;
            head = baru;
        }
    }

    void tambahBelakang(String nama, String alamat, int umur, double ipk) {
        Node baru = new Node(nama, alamat, umur, ipk);
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
            tail = baru;
        }
    }

    void tambahTengah(String dicari, String nama, String alamat, int umur, double
ipk) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(dicari)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data induk tidak ditemukan!");
            return;
        }
        Node baru = new Node(nama, alamat, umur, ipk);
        Node sesudah = bantu.next;
        bantu.next = baru;
        baru.prev = bantu;
        if (sesudah != null) {
            baru.next = sesudah;
            sesudah.prev = baru;
```

```
        } else {
            tail = baru;
        }
    }

    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(nama)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
            return;
        }
        if (bantu == head && bantu == tail) {
            head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
    }

    void cetakMaju() {
        Node bantu = head;
        System.out.println("\nCETAK MAJU :");
        System.out.printf("%-15s %-15s %-5s %-5s\n", "NAMA", "ALAMAT", "UMUR",
"IPK");
        System.out.println("-----------------------------------------------");
        while (bantu != null) {
            System.out.printf("%-15s   %-15s   %-5d   %-5.2f\n",   bantu.nama,
bantu.alamat, bantu.umur, bantu.ipk);
            bantu = bantu.next;
        }
    }

    void cetakMundur() {
        Node bantu = tail;
        System.out.println("\nCETAK MUNDUR :");
        System.out.printf("%-15s %-15s %-5s %-5s\n", "NAMA", "ALAMAT", "UMUR",
"IPK");
        System.out.println("-----------------------------------------------");
        while (bantu != null) {
            System.out.printf("%-15s   %-15s   %-5d   %-5.2f\n",   bantu.nama,
bantu.alamat, bantu.umur, bantu.ipk);
            bantu = bantu.prev;
        }
```

```
    }

    public int hitungJumlahSimpul() {
        int jumlah = 0;
        Node bantu = head;
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }

    // ========================================================================
    // 1. SORTING NUMERIK: TUKAR NILAI (VALUE SWAP)
    // ========================================================================
    // Parameter:
    // jenisData: 1=Umur, 2=IPK
    // urutan: 1=Ascending (Kecil->Besar), 2=Descending (Besar->Kecil)
    public void sortNumerik_TukarNilai(int jenisData, int urutan) {
        int N = hitungJumlahSimpul();
        if (N < 2) return;

        System.out.println("\nProses Sorting Numerik (Tukar Nilai)...");
        Node A, B;

        for (int i = 0; i < N - 1; i++) {
            A = head;
            B = head.next;
            while (B != null) {
                boolean tukar = false;

                // Ambil nilai berdasarkan jenis data
                double valA = (jenisData == 1) ? A.umur : A.ipk;
                double valB = (jenisData == 1) ? B.umur : B.ipk;

                // Cek kondisi berdasarkan urutan (Asc/Desc)
                if (urutan == 1) { // Ascending
                    if (valA > valB) tukar = true;
                } else { // Descending
                    if (valA < valB) tukar = true;
                }

                // Lakukan penukaran data jika kondisi terpenuhi
                if (tukar) {
                    // Tukar Nama
                    String tNama = A.nama; A.nama = B.nama; B.nama = tNama;
                    // Tukar Alamat
                    String tAlamat = A.alamat; A.alamat = B.alamat; B.alamat =
tAlamat;
                    // Tukar Umur
                    int tUmur = A.umur; A.umur = B.umur; B.umur = tUmur;
                    // Tukar IPK
```

```
                    double tIpk = A.ipk; A.ipk = B.ipk; B.ipk = tIpk;
                }
                A = A.next;
                B = B.next;
            }
        }
        System.out.println("Sorting Selesai.");
    }

    // ======================================================================
    // 2. SORTING NUMERIK: TUKAR HEAP / POINTER
    // ======================================================================
    public void sortNumerik_TukarHeap(int jenisData, int urutan) {
        int N = hitungJumlahSimpul();
        if (N < 2) return;

        System.out.println("\nProses Sorting Numerik (Tukar Heap/Pointer)...");
        Node bantu;

        for (int i = 1; i <= N; i++) {
            // A. Khusus menguji simpul pertama (HEAD) dgn sebelahnya
            if (head != null && head.next != null) {
                boolean tukarHead = false;
                double valHead = (jenisData == 1) ? head.umur : head.ipk;
                double valNext = (jenisData == 1) ? head.next.umur :
head.next.ipk;

                if (urutan == 1) { // Ascending
                    if (valHead > valNext) tukarHead = true;
                } else { // Descending
                    if (valHead < valNext) tukarHead = true;
                }

                if (tukarHead) {
                    bantu = head.next;
                    head.next = bantu.next;
                    if (bantu.next != null) {
                        bantu.next.prev = head;
                    }
                    bantu.next = head;
                    bantu.prev = null;
                    head.prev = bantu;
                    head = bantu; // Update Head
                }
            }

            // B. Khusus menguji simpul kedua dst (INNER LOOP)
            bantu = head;
            while (bantu.next != tail && bantu.next != null) {
                Node A = bantu.next;
                Node B = bantu.next.next;
```

```
                if (B != null) {
                    boolean tukarNode = false;
                    double valA = (jenisData == 1) ? A.umur : A.ipk;
                    double valB = (jenisData == 1) ? B.umur : B.ipk;

                    if (urutan == 1) { // Ascending
                        if (valA > valB) tukarNode = true;
                    } else { // Descending
                        if (valA < valB) tukarNode = true;
                    }

                    if (tukarNode) {
                        // Logika Tukar Pointer (Heap)
                        A.next = B.next;
                        if (B != tail) {
                            if (A.next != null) A.next.prev = A;
                        }
                        B.next = A;
                        A.prev = B;
                        bantu.next = B;
                        B.prev = bantu;

                        if (B == tail) tail = A; // Update Tail
                    }
                }
                bantu = bantu.next;
            }
        }
        System.out.println("Sorting Selesai.");
    }
}

public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();

        int pilih, jenisData, urutan, umur;
        double ipk;
        String nama, alamat, dicari;

        do {
            System.out.println("\nMENU MASTER (NUMERIK):");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Tambah Tengah");
            System.out.println("4. Hapus");
            System.out.println("5. Cetak Maju");
            System.out.println("6. Cetak Mundur");
            System.out.println("7. Urutkan Data (SORTING)");
            System.out.println("8. Keluar");
            System.out.print("Pilih: ");
```

```
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    System.out.print("Umur   : "); umur = sc.nextInt();
                    System.out.print("IPK    : "); ipk = sc.nextDouble();
                    dll.tambahDepan(nama, alamat, umur, ipk);
                    break;
                case 2:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    System.out.print("Umur   : "); umur = sc.nextInt();
                    System.out.print("IPK    : "); ipk = sc.nextDouble();
                    dll.tambahBelakang(nama, alamat, umur, ipk);
                    break;
                case 3:
                    System.out.print("Cari Nama : "); dicari = sc.nextLine();
                    System.out.print("Nama Baru : "); nama = sc.nextLine();
                    System.out.print("Alamat    : "); alamat = sc.nextLine();
                    System.out.print("Umur Baru : "); umur = sc.nextInt();
                    System.out.print("IPK Baru  : "); ipk = sc.nextDouble();
                    dll.tambahTengah(dicari, nama, alamat, umur, ipk);
                    break;
                case 4:
                    System.out.print("Nama Hapus: "); nama = sc.nextLine();
                    dll.hapus(nama);
                    break;
                case 5:
                    dll.cetakMaju();
                    break;
                case 6:
                    dll.cetakMundur();
                    break;
                case 7:
                    // Menu Sorting Dinamis
                    System.out.println("\n--- PENGATURAN SORTING ---");
                    System.out.println("Pilih Teknik:");
                    System.out.println("1. Tukar NILAI (Value Swap)");
                    System.out.println("2. Tukar HEAP (Pointer Swap)");
                    System.out.print("Pilihan: ");
                    int teknik = sc.nextInt();

                    System.out.println("Urutkan Berdasarkan:");
                    System.out.println("1. Umur");
                    System.out.println("2. IPK");
                    System.out.print("Pilihan: ");
                    jenisData = sc.nextInt();

                    System.out.println("Jenis Urutan:");
```

```
                    System.out.println("1. Ascending (Kecil -> Besar)");
                    System.out.println("2. Descending (Besar -> Kecil)");
                    System.out.print("Pilihan: ");
                    urutan = sc.nextInt();

                    if (teknik == 1) {
                        dll.sortNumerik_TukarNilai(jenisData, urutan);
                    } else if (teknik == 2) {
                        dll.sortNumerik_TukarHeap(jenisData, urutan);
                    } else {
                        System.out.println("Teknik salah!");
                    }
                    dll.cetakMaju();
                    break;
            }

        } while (pilih != 8);
    }
}
```

```
MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 1
Nama   : Fidelia
Alamat : Kaltim
Umur   : 19
IPK    : 4.0

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 1
Nama   : Melia
Alamat : Kaltim
Umur   : 19
IPK    : 3.9

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 1
Nama   : Mian
Alamat : Sulsel
Umur   : 19
```

```
IPK    : 3.9

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 2
Nama   : Putra
Alamat : Kaltara
Umur   : 19
IPK    : 3.8

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 2
Nama   : Rama
Alamat : Solo
Umur   : 19
IPK    : 3.6

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 3
Cari Nama : Fidelia
Nama Baru : Dewi
```

```
Alamat    : Ambon
Umur Baru : 18
IPK Baru  : 4.0

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 5

CETAK MAJU :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
Mian            Sulsel          19    3.90
Melia           Kaltim          19    3.90
Fidelia         Kaltim          19    4.00
Dewi            Ambon           18    4.00
Putra           Kaltara         19    3.80
Rama            Solo            19    3.60

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 4
Nama Hapus: Fidelia

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
```

```
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 6

CETAK MUNDUR :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
Rama            Solo            19    3.60
Putra           Kaltara         19    3.80
Dewi            Ambon           18    4.00
Melia           Kaltim          19    3.90
Mian            Sulsel          19    3.90

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 7

--- PENGATURAN SORTING ---
Pilih Teknik:
1. Tukar NILAI (Value Swap)
2. Tukar HEAP (Pointer Swap)
Pilihan: 1
Urutkan Berdasarkan:
1. Umur
2. IPK
Pilihan: 1
Jenis Urutan:
1. Ascending (Kecil -> Besar)
2. Descending (Besar -> Kecil)
Pilihan: 1

Proses Sorting Numerik (Tukar Nilai)...
Sorting Selesai.

CETAK MAJU :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
```

```
Dewi            Ambon           18    4.00
Mian            Sulsel          19    3.90
Melia           Kaltim          19    3.90
Putra           Kaltara         19    3.80
Rama            Solo            19    3.60

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 7

--- PENGATURAN SORTING ---
Pilih Teknik:
1. Tukar NILAI (Value Swap)
2. Tukar HEAP (Pointer Swap)
Pilihan: 2
Urutkan Berdasarkan:
1. Umur
2. IPK
Pilihan: 2
Jenis Urutan:
1. Ascending (Kecil -> Besar)
2. Descending (Besar -> Kecil)
Pilihan: 2

Proses Sorting Numerik (Tukar Heap/Pointer)...
Sorting Selesai.

CETAK MAJU :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
Dewi            Ambon           18    4.00
Mian            Sulsel          19    3.90
Melia           Kaltim          19    3.90
Putra           Kaltara         19    3.80
Rama            Solo            19    3.60

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 8
Press any key to continue . . .
```

Pembahasan : Program ini merupakan aplikasi pengelolaan data mahasiswa yang menerapkan struktur data double linked list untuk menyimpan dan memanipulasi informasi berupa nama, alamat, umur, dan IPK. Setiap data direpresentasikan dalam class Node yang memiliki pointer

next dan prev sehingga memungkinkan penelusuran data secara dua arah dari awal maupun dari akhir. Class DoubleLinkedList berfungsi sebagai pengelola utama yang menyediakan operasi penambahan data di depan, belakang, dan tengah berdasarkan nama tertentu, penghapusan data berdasarkan nama, serta penampilan data secara maju dan mundur. Program juga dilengkapi dengan proses pengurutan data numerik menggunakan dua pendekatan, yaitu pengurutan dengan metode tukar nilai yang menukar isi atribut antar node tanpa mengubah posisi node, serta pengurutan dengan metode tukar heap atau pointer yang menukar posisi node dengan mengatur ulang hubungan pointer antar node. Pengurutan dapat dilakukan berdasarkan umur atau IPK dengan pilihan urutan menaik atau menurun sesuai input pengguna. Seluruh fungsi dijalankan melalui menu interaktif pada method main menggunakan Scanner sehingga pengguna dapat mengelola data secara dinamis sekaligus memahami cara kerja linked list, manipulasi pointer, dan algoritma sorting dalam satu program yang terintegrasi.

Program 11.2

```
import java.util.Scanner;

class Node {
    String nama, alamat;
    int umur;       // Tambahan atribut numerik
    double ipk;     // Tambahan atribut numerik
    Node next, prev; // next = kanan, prev = kiri

    public Node(String nama, String alamat, int umur, double ipk) {
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.ipk = ipk;
        next = null;
        prev = null;
    }
}

class DoubleLinkedList {
    Node head, tail; // head = awal, tail = akhir

    public DoubleLinkedList() {
        head = tail = null;
    }

    // --- FUNGSI CRUD (Disesuaikan dengan Umur & IPK) ---

    void tambahDepan(String nama, String alamat, int umur, double ipk) {
        Node baru = new Node(nama, alamat, umur, ipk);
        if (head == null) {
            head = tail = baru;
        } else {
            baru.next = head;
            head.prev = baru;
            head = baru;
```

```
        }
    }

    void tambahBelakang(String nama, String alamat, int umur, double ipk) {
        Node baru = new Node(nama, alamat, umur, ipk);
        if (tail == null) {
            head = tail = baru;
        } else {
            tail.next = baru;
            baru.prev = tail;
            tail = baru;
        }
    }

    void tambahTengah(String dicari, String nama, String alamat, int umur, double
ipk) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(dicari)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data induk tidak ditemukan!");
            return;
        }
        Node baru = new Node(nama, alamat, umur, ipk);
        Node sesudah = bantu.next;
        bantu.next = baru;
        baru.prev = bantu;
        if (sesudah != null) {
            baru.next = sesudah;
            sesudah.prev = baru;
        } else {
            tail = baru;
        }
    }

    void hapus(String nama) {
        Node bantu = head;
        while (bantu != null && !bantu.nama.equals(nama)) {
            bantu = bantu.next;
        }
        if (bantu == null) {
            System.out.println("Data tidak ditemukan!");
            return;
        }
        if (bantu == head && bantu == tail) {
            head = tail = null;
        } else if (bantu == head) {
            head = head.next;
            head.prev = null;
        } else if (bantu == tail) {
            tail = tail.prev;
```

```
            tail.next = null;
        } else {
            bantu.prev.next = bantu.next;
            bantu.next.prev = bantu.prev;
        }
    }

    void cetakMaju() {
        Node bantu = head;
        System.out.println("\nCETAK MAJU :");
        System.out.printf("%-15s %-15s %-5s %-5s\n", "NAMA", "ALAMAT", "UMUR",
"IPK");
        System.out.println("-----------------------------------------------");
        while (bantu != null) {
            System.out.printf("%-15s   %-15s   %-5d   %-5.2f\n",   bantu.nama,
bantu.alamat, bantu.umur, bantu.ipk);
            bantu = bantu.next;
        }
    }

    void cetakMundur() {
        Node bantu = tail;
        System.out.println("\nCETAK MUNDUR :");
        System.out.printf("%-15s %-15s %-5s %-5s\n", "NAMA", "ALAMAT", "UMUR",
"IPK");
        System.out.println("-----------------------------------------------");
        while (bantu != null) {
            System.out.printf("%-15s   %-15s   %-5d   %-5.2f\n",   bantu.nama,
bantu.alamat, bantu.umur, bantu.ipk);
            bantu = bantu.prev;
        }
    }

    public int hitungJumlahSimpul() {
        int jumlah = 0;
        Node bantu = head;
        while (bantu != null) {
            jumlah++;
            bantu = bantu.next;
        }
        return jumlah;
    }

    // ========================================================================
    // 1. SORTING NUMERIK: TUKAR NILAI (VALUE SWAP)
    // ========================================================================
    // Parameter:
    // jenisData: 1=Umur, 2=IPK
    // urutan: 1=Ascending (Kecil->Besar), 2=Descending (Besar->Kecil)
    public void sortNumerik_TukarNilai(int jenisData, int urutan) {
        int N = hitungJumlahSimpul();
        if (N < 2) return;
```

```
        System.out.println("\nProses Sorting Numerik (Tukar Nilai)...");
        Node A, B;

        for (int i = 0; i < N - 1; i++) {
            A = head;
            B = head.next;
            while (B != null) {
                boolean tukar = false;

                // Ambil nilai berdasarkan jenis data
                double valA = (jenisData == 1) ? A.umur : A.ipk;
                double valB = (jenisData == 1) ? B.umur : B.ipk;

                // Cek kondisi berdasarkan urutan (Asc/Desc)
                if (urutan == 1) { // Ascending
                    if (valA > valB) tukar = true;
                } else { // Descending
                    if (valA < valB) tukar = true;
                }

                // Lakukan penukaran data jika kondisi terpenuhi
                if (tukar) {
                    // Tukar Nama
                    String tNama = A.nama; A.nama = B.nama; B.nama = tNama;
                    // Tukar Alamat
                    String tAlamat = A.alamat; A.alamat = B.alamat; B.alamat =
tAlamat;
                    // Tukar Umur
                    int tUmur = A.umur; A.umur = B.umur; B.umur = tUmur;
                    // Tukar IPK
                    double tIpk = A.ipk; A.ipk = B.ipk; B.ipk = tIpk;
                }
                A = A.next;
                B = B.next;
            }
        }
        System.out.println("Sorting Selesai.");
    }

    // ======================================================================
    // 2. SORTING NUMERIK: TUKAR HEAP / POINTER
    // ======================================================================
    public void sortNumerik_TukarHeap(int jenisData, int urutan) {
        int N = hitungJumlahSimpul();
        if (N < 2) return;

        System.out.println("\nProses Sorting Numerik (Tukar Heap/Pointer)...");
        Node bantu;

        for (int i = 1; i <= N; i++) {
            // A. Khusus menguji simpul pertama (HEAD) dgn sebelahnya
```

```
            if (head != null && head.next != null) {
                boolean tukarHead = false;
                double valHead = (jenisData == 1) ? head.umur : head.ipk;
                double valNext = (jenisData == 1) ? head.next.umur : head.next.ipk;

                if (urutan == 1) { // Ascending
                    if (valHead > valNext) tukarHead = true;
                } else { // Descending
                    if (valHead < valNext) tukarHead = true;
                }

                if (tukarHead) {
                    bantu = head.next;
                    head.next = bantu.next;
                    if (bantu.next != null) {
                        bantu.next.prev = head;
                    }
                    bantu.next = head;
                    bantu.prev = null;
                    head.prev = bantu;
                    head = bantu; // Update Head
                }
            }

            // B. Khusus menguji simpul kedua dst (INNER LOOP)
            bantu = head;
            while (bantu.next != tail && bantu.next != null) {
                Node A = bantu.next;
                Node B = bantu.next.next;

                if (B != null) {
                    boolean tukarNode = false;
                    double valA = (jenisData == 1) ? A.umur : A.ipk;
                    double valB = (jenisData == 1) ? B.umur : B.ipk;

                    if (urutan == 1) { // Ascending
                        if (valA > valB) tukarNode = true;
                    } else { // Descending
                        if (valA < valB) tukarNode = true;
                    }

                    if (tukarNode) {
                        // Logika Tukar Pointer (Heap)
                        A.next = B.next;
                        if (B != tail) {
                            if (A.next != null) A.next.prev = A;
                        }
                        B.next = A;
                        A.prev = B;
                        bantu.next = B;
                        B.prev = bantu;
```

```
                        if (B == tail) tail = A; // Update Tail
                    }
                }
                bantu = bantu.next;
            }
        }
        System.out.println("Sorting Selesai.");
    }
}

public class senaraiGanda {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoubleLinkedList dll = new DoubleLinkedList();

        int pilih, jenisData, urutan, umur;
        double ipk;
        String nama, alamat, dicari;

        do {
            System.out.println("\nMENU MASTER (NUMERIK):");
            System.out.println("1. Tambah Depan");
            System.out.println("2. Tambah Belakang");
            System.out.println("3. Tambah Tengah");
            System.out.println("4. Hapus");
            System.out.println("5. Cetak Maju");
            System.out.println("6. Cetak Mundur");
            System.out.println("7. Urutkan Data (SORTING)");
            System.out.println("8. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    System.out.print("Umur   : "); umur = sc.nextInt();
                    System.out.print("IPK    : "); ipk = sc.nextDouble();
                    dll.tambahDepan(nama, alamat, umur, ipk);
                    break;
                case 2:
                    System.out.print("Nama   : "); nama = sc.nextLine();
                    System.out.print("Alamat : "); alamat = sc.nextLine();
                    System.out.print("Umur   : "); umur = sc.nextInt();
                    System.out.print("IPK    : "); ipk = sc.nextDouble();
                    dll.tambahBelakang(nama, alamat, umur, ipk);
                    break;
                case 3:
                    System.out.print("Cari Nama : "); dicari = sc.nextLine();
                    System.out.print("Nama Baru : "); nama = sc.nextLine();
```

```
                    System.out.print("Alamat    : "); alamat = sc.nextLine();
                    System.out.print("Umur Baru : "); umur = sc.nextInt();
                    System.out.print("IPK Baru  : "); ipk = sc.nextDouble();
                    dll.tambahTengah(dicari, nama, alamat, umur, ipk);
                    break;
                case 4:
                    System.out.print("Nama Hapus: "); nama = sc.nextLine();
                    dll.hapus(nama);
                    break;
                case 5:
                    dll.cetakMaju();
                    break;
                case 6:
                    dll.cetakMundur();
                    break;
                case 7:
                    // Menu Sorting Dinamis
                    System.out.println("\n--- PENGATURAN SORTING ---");
                    System.out.println("Pilih Teknik:");
                    System.out.println("1. Tukar NILAI (Value Swap)");
                    System.out.println("2. Tukar HEAP (Pointer Swap)");
                    System.out.print("Pilihan: ");
                    int teknik = sc.nextInt();

                    System.out.println("Urutkan Berdasarkan:");
                    System.out.println("1. Umur");
                    System.out.println("2. IPK");
                    System.out.print("Pilihan: ");
                    jenisData = sc.nextInt();

                    System.out.println("Jenis Urutan:");
                    System.out.println("1. Ascending (Kecil -> Besar)");
                    System.out.println("2. Descending (Besar -> Kecil)");
                    System.out.print("Pilihan: ");
                    urutan = sc.nextInt();

                    if (teknik == 1) {
                        dll.sortNumerik_TukarNilai(jenisData, urutan);
                    } else if (teknik == 2) {
                        dll.sortNumerik_TukarHeap(jenisData, urutan);
                    } else {
                        System.out.println("Teknik salah!");
                    }
                    dll.cetakMaju();
                    break;
            }

        } while (pilih != 8);
    }
}
```

```
MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 1
Nama   : Fidelia
Alamat : Kaltim
Umur   : 19
IPK    : 4.0

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 1
Nama   : Melia
Alamat : Kaltim
Umur   : 19
IPK    : 3.9

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 1
Nama   : Mian
Alamat : Sulsel
Umur   : 19
```

```
IPK    : 3.9

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 2
Nama   : Putra
Alamat : Kaltara
Umur   : 19
IPK    : 3.8

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 2
Nama   : Rama
Alamat : Solo
Umur   : 19
IPK    : 3.6

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 3
Cari Nama : Fidelia
Nama Baru : Dewi
```

```
Alamat    : Ambon
Umur Baru : 18
IPK Baru  : 4.0

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 5

CETAK MAJU :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
Mian            Sulsel          19    3.90
Melia           Kaltim          19    3.90
Fidelia         Kaltim          19    4.00
Dewi            Ambon           18    4.00
Putra           Kaltara         19    3.80
Rama            Solo            19    3.60

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 4
Nama Hapus: Fidelia

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
```

```
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 6

CETAK MUNDUR :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
Rama            Solo            19    3.60
Putra           Kaltara         19    3.80
Dewi            Ambon           18    4.00
Melia           Kaltim          19    3.90
Mian            Sulsel          19    3.90

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 7

--- PENGATURAN SORTING ---
Pilih Teknik:
1. Tukar NILAI (Value Swap)
2. Tukar HEAP (Pointer Swap)
Pilihan: 1
Urutkan Berdasarkan:
1. Umur
2. IPK
Pilihan: 1
Jenis Urutan:
1. Ascending (Kecil -> Besar)
2. Descending (Besar -> Kecil)
Pilihan: 1

Proses Sorting Numerik (Tukar Nilai)...
Sorting Selesai.

CETAK MAJU :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
```

```
Dewi            Ambon           18    4.00
Mian            Sulsel          19    3.90
Melia           Kaltim          19    3.90
Putra           Kaltara         19    3.80
Rama            Solo            19    3.60

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 7

--- PENGATURAN SORTING ---
Pilih Teknik:
1. Tukar NILAI (Value Swap)
2. Tukar HEAP (Pointer Swap)
Pilihan: 2
Urutkan Berdasarkan:
1. Umur
2. IPK
Pilihan: 2
Jenis Urutan:
1. Ascending (Kecil -> Besar)
2. Descending (Besar -> Kecil)
Pilihan: 2

Proses Sorting Numerik (Tukar Heap/Pointer)...
Sorting Selesai.

CETAK MAJU :
NAMA            ALAMAT          UMUR  IPK
-----------------------------------------------
Dewi            Ambon           18    4.00
Mian            Sulsel          19    3.90
Melia           Kaltim          19    3.90
Putra           Kaltara         19    3.80
Rama            Solo            19    3.60

MENU MASTER (NUMERIK):
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Hapus
5. Cetak Maju
6. Cetak Mundur
7. Urutkan Data (SORTING)
8. Keluar
Pilih: 8
Press any key to continue . . .
```

Pembahasan : Program ini merupakan aplikasi pengelolaan data berbasis struktur data double linked list yang digunakan untuk menyimpan dan memproses data mahasiswa berupa nama alamat umur dan IPK. Setiap data disimpan dalam node yang saling terhubung dua arah melalui pointer next dan prev sehingga memungkinkan penelusuran data dari depan maupun dari belakang. Class DoubleLinkedList berfungsi mengatur seluruh operasi seperti menambah data di depan belakang dan tengah berdasarkan nama tertentu menghapus data mencetak data secara

maju dan mundur serta menghitung jumlah simpul yang ada. Program ini juga mengimplementasikan proses sorting data numerik menggunakan dua teknik yaitu tukar nilai yang menukar isi atribut antar node tanpa mengubah posisi node dan tukar heap atau pointer yang menukar posisi node dengan mengatur ulang hubungan pointer antar node. Proses pengurutan dapat dilakukan berdasarkan umur atau IPK dengan pilihan urutan menaik atau menurun sesuai input pengguna. Seluruh fitur dijalankan melalui menu interaktif pada method main sehingga pengguna dapat mengelola data secara dinamis sekaligus memahami konsep linked list manipulasi pointer serta algoritma sorting numerik dalam satu program yang terintegrasi.

D. KESIMPULAN

Berdasarkan hasil praktikum mengurutkan data (sorting) dan pencarian data (searching) pada linked list, dapat disimpulkan bahwa struktur data linked list mampu mendukung proses pengolahan data secara dinamis tanpa bergantung pada indeks seperti array. Proses **sorting** menggunakan algoritma Bubble Sort menunjukkan bahwa data dapat diurutkan baik dengan teknik tukar nilai maupun tukar pointer (node), di mana teknik tukar pointer lebih mencerminkan karakteristik linked list karena memanipulasi hubungan antar simpul. Sementara itu, proses **searching** dengan metode Linear Search memungkinkan pencarian data dilakukan secara berurutan dari simpul awal hingga data ditemukan atau data berakhir. Praktikum ini memperkuat pemahaman bahwa linked list, khususnya double linked list, efektif digunakan untuk operasi penyisipan, penghapusan, pengurutan, dan pencarian data, serta membantu mahasiswa memahami cara kerja pointer dan traversal dalam struktur data berantai.