JOBSHEET 7
ALGORITMA STRUKTUR DATA

Burhnauddin ihsan
244107020189
TI 1E/06

PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

- **PERCOBAAN SEQUENTIAL SEARCH**

1. Kode program

MahasiswaBerprestasi06

```
public class MahasiswaBerprestasi06 {
    Mahasiswa06[] listMhs = new Mahasiswa06[5];
    int idx;

    void tambah(Mahasiswa06 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa06 m : listMhs) {
            m.tampilkanInformasi();
            System.out.println("-----------------------");
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 1; j < listMhs.length - i; j++) {
                if (listMhs[j].ipk > listMhs[j - 1].ipk) {
                    Mahasiswa06 tmp = listMhs[j];
                    listMhs[j] = listMhs[j - 1];
                    listMhs[j - 1] = tmp;
                }
            }
        }
    }

    void selectionSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            int idxMin = i;
            for (int j = i + 1; j < listMhs.length; j++) {
                if (listMhs[j].ipk < listMhs[idxMin].ipk) {
                    idxMin = j;
                }
            }
            Mahasiswa06 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }
```

```
 void insertionSort() {
        for (int i = 1; i < listMhs.length; i++) {
            Mahasiswa06 temp = listMhs[i];
            int j = i;
            while (j > 0 && listMhs[j - 1].ipk < temp.ipk) {
                listMhs[j] = listMhs[j - 1];
                j--;
            }
            listMhs[j] = temp;
        }
    }

    int sequentialSearching(double cari) {
        int posisi = -1;
        for (int j = 0; j < listMhs.length; j++) {
            if (listMhs[j].ipk == cari) {
                posisi = j;
                break;
            }
        }
        return posisi;
    }

    void tampilPosisi(double x, int pos) {
        if (pos != -1) {
            System.out.println("Data mahasiswa dengan IPK: " + x + " ditemukan pada
indeks " + pos);
        } else {
            System.out.println("Data " + x + " tidak ditemukan");
        }
    }

    void tampilDataSearch(double x, int pos) {
        if (pos != -1) {
            System.out.println("NIM  : " + listMhs[pos].nim);
            System.out.println("Nama     : " + listMhs[pos].nama);
            System.out.println("Kelas    : " + listMhs[pos].kelas);
            System.out.println("IPK  : " + x);
        } else {
            System.out.println("Data mahasiswa dengan IPK " + x + " tidak
ditemukan");
        }
    }
}
```

MahasiswaDemo06

```
import java.util.Scanner;

public class MahasiswaDemo06 {
    public static void main(String[] args) {
        MahasiswaBerprestasi06 list = new MahasiswaBerprestasi06();
        Scanner sc = new Scanner(System.in);
        int jumlahMhs = 5;

        for (int i = 0; i < jumlahMhs; i++) {
            System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));

            System.out.print("NIM   : ");
            String nim = sc.nextLine();

            System.out.print("Nama  : ");
            String nama = sc.nextLine();

            System.out.print("Kelas : ");
            String kelas = sc.nextLine();

            System.out.print("IPK   : ");
            double ipk = sc.nextDouble();
            sc.nextLine(); // Membersihkan buffer

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

            Mahasiswa06 mhs = new Mahasiswa06(nim, nama, kelas, ipk);
            list.tambah(mhs);
        }

        System.out.println("Data mahasiswa sebelum sorting:");
        list.tampil();

        // Melakukan Pencarian Data Sequential
        System.out.println("-------------------------------------");
        System.out.println("Pencarian Data");
        System.out.println("-------------------------------------");

        System.out.print("Masukkan IPK mahasiswa yang dicari: ");
        double cari = sc.nextDouble();

        System.out.println("Menggunakan sequential searching");
        double posisi = list.sequentialSearching(cari);
        int pss = (int) posisi;

        list.tampilPosisi(cari, pss);
        list.tampilDataSearch(cari, pss);
    }
}
```

2. Hasil dari kode program

```
Masukkan Data Mahasiswa ke-1
NIM     : 111
Nama    : adi
Kelas   : 2
IPK     : 3,6
---------------------------------
Masukkan Data Mahasiswa ke-2
NIM     : 222
Nama    : tio
Kelas   : 2
IPK     : 3,8
---------------------------------
Masukkan Data Mahasiswa ke-3
NIM     : 333
Nama    : ila
Kelas   : 2
IPK     : 3
---------------------------------
Masukkan Data Mahasiswa ke-4
NIM     : 444
Nama    : lia
Kelas   : 2
IPK     : 3,5
---------------------------------
Masukkan Data Mahasiswa ke-5
NIM     : 555
Nama    : fia
Kelas   : 2
IPK     : 3,3
---------------------------------
```

```
Nama : adi
NIM : 111
Kelas : 2
IPK : 3.6
----------------------
Nama : tio
NIM : 222
Kelas : 2
IPK : 3.8
----------------------
Nama : ila
NIM : 333
Kelas : 2
IPK : 3.0
----------------------
Nama : lia
NIM : 444
Kelas : 2
IPK : 3.5
----------------------
Nama : fia
NIM : 555
Kelas : 2
IPK : 3.3
----------------------
---------------------------------
Pencarian Data
---------------------------------
Masukkan ipk mahasiswa yang dicari:
IPK: 3,5
Menggunakan sequential searching
```

```
Data mahasiswa dengan IPK :3.5 ditemukan pada indeks 3
nim       : 444
nama      : lia
kelas     : 2
ipk       : 3.5
```

PERTANYAAN

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!

   Jawab: Metode tampilPosisi dalam kelas MahasiswaBerprestasi06 digunakan untuk menampilkan posisi (indeks) mahasiswa dalam array berdasarkan pencarian IPK. Jika data ditemukan, metode ini mencetak indeksnya, sedangkan jika tidak ditemukan, akan mencetak pesan bahwa data tidak ada. Sementara itu, metode tampilDataSearch digunakan untuk menampilkan informasi lengkap mahasiswa, seperti NIM, nama, kelas, dan IPK. Jika mahasiswa ditemukan, semua data tersebut ditampilkan, tetapi jika tidak ditemukan, akan muncul pesan bahwa mahasiswa dengan IPK tersebut tidak ada. Perbedaan utama keduanya adalah tampilPosisi hanya menunjukkan indeks, sedangkan tampilDataSearch menampilkan detail lengkap mahasiswa.

2. Jelaskan fungsi break pada kode program dibawah ini!

```
if (listMhs[j].ipk==cari){
    posisi=j;
    break;
}
```

   Fungsi break pada kode program tersebut digunakan untuk menghentikan perulangan for segera setelah menemukan mahasiswa dengan IPK yang sesuai dengan nilai yang dicari (cari).Saat kondisi if (listMhs[j].ipk == cari) terpenuhi, variabel posisi akan menyimpan indeks j, dan break akan keluar dari perulangan, sehingga pencarian tidak perlu dilanjutkan ke elemen-elemen berikutnya. Hal ini mengoptimalkan pencarian dengan menghentikan iterasi lebih awal setelah menemukan hasil yang sesuai, sehingga program berjalan lebih efisien.

- **PERCOBAAN SEARCHING MENGGUNAKAN BINARY SEARCH**

  1. Kode program

     Penambahan kode pada class MahasiswaBerprestasi06

```
int findBinarySearch (double cari, int left, int right){
        int mid;
        if (right >= left) {
            mid = (left + right) /2;
            if (cari == listMhs[mid].ipk) {
                return (mid);
            }
            else if (listMhs[mid].ipk < cari){
                return findBinarySearch(cari, mid+1, right);
            }
            else {
                return findBinarySearch(cari, left, mid-1);
```

```
    }
            }
            return -1;
        }
}
```

Penambahan kode program pada class MahasiswaDemo06

```
 System.out.println("-----------------------------------");
        System.out.println("Menggunakan binary search");
        System.out.println("-----------------------------------");
        double posisi2 = list.findBinarySearch(cari, 0, jumMhs-1);
        int pss2 = (int) posisi2;
        list.tampilPosisi(cari, pss2);
        list.tampilDataSearch(cari, pss2);

    }
}
```

2. Hasil dari kode program

```
Masukkan Data Mahasiswa ke-1
NIM     : 111
Nama    : adi
Kelas   : 2
IPK     : 3,1
---------------------------------
Masukkan Data Mahasiswa ke-2
NIM     : 222
Nama    : ila
Kelas   : 2
IPK     : 3,2
---------------------------------
Masukkan Data Mahasiswa ke-3
NIM     : 333
Nama    : lia
Kelas   : 2
IPK     : 3,3
---------------------------------
Masukkan Data Mahasiswa ke-4
NIM     : 444
Nama    : susi
Kelas   : 2
IPK     : 3,5
---------------------------------
Masukkan Data Mahasiswa ke-5
NIM     : 555
Nama    : anita
Kelas   : 2
IPK     : 3,7
---------------------------------
```

```
Nama : adi
NIM : 111
Kelas : 2
IPK : 3.1
----------------------
Nama : ila
NIM : 222
Kelas : 2
IPK : 3.2
----------------------
Nama : lia
NIM : 333
Kelas : 2
IPK : 3.3
----------------------
Nama : susi
NIM : 444
Kelas : 2
IPK : 3.5
----------------------
Nama : anita
NIM : 555
Kelas : 2
IPK : 3.7
----------------------
```

```
-----------------------------------
Pencarian Data
-----------------------------------
Masukkan ipk mahasiswa yang dicari:
IPK: 3,7
-----------------------------------
Menggunakan binary search
-----------------------------------
Data mahasiswa dengan IPK : 3.7 ditemukan pada indeks 4
nim      : 555
nama     : anita
kelas    : 2
ipk      : 3.7
```

PERTANYAAN

1. Tunjukkan pada kode program yang mana proses divide dijalankan!

```
mid = (left + right) /2;
```

Bagian tersebut merupakan proses divide dijalankan dimana, memecah array menjadi dua (menyederhanakan array menjadi array yang lebih kecil)

2. Tunjukkan pada kode program yang mana proses conquer dijalankan!

```
else if (listMhs[mid].ipk < cari){
    return findBinarySearch(cari, mid+1, right);
}
else {
    return findBinarySearch(cari, left, mid-1);
}
```

Conquer pada program ini terbagi menjadi dua bagian, bagian untuk melakukan pencarian pada bagian kiri (dari left hingga mid-1) jika kata kunci pencarian lebih kecil dari nilai elemen array pada index mid. Dan bagian kanan untuk melakukan pencarian pada bagian kanan (dari mid+1 hingga right) jika kata kunci pencarian lebih besar dari nilai elemen array pada index mid.

3. Jika data IPK yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!

   Jawab: Program pencarian binary search tidak bisa berjalan dan tidak berjalan dengan baik jika data IPK yang dimasukkan tidak urut. Hal tersebut dikarenakan cara kerja binary search membandingkan data dengan elemen array pada index tengah, jika bilangan pencarian lebih besar maka proses binary search akan berjalan ke kanan karena bagian kanan array pada proses binary search dianggap sebagai bilangan yang lebih besar, jika data IPK pada array tidak diurutkan maka bagian kanan array sebagai bilangan yang lebih besar menjadi tidak valid, begitu juga dengan bagian kiri.

4. Jika IPK yang dimasukkan dari IPK terbesar ke terkecil (missal : 3.8, 3.7, 3.5, 3.4, 3.2) dan elemen yang dicari adalah 3.2. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai.

   Jawab: Jika IPK yang dimasukkan dari IPK terbesar ke terkecil (missal : 3.8, 3.7, 3.5, 3.4, 3.2) dan elemen yang dicari adalah 3.2. Maka dari binary search tidak akan ditemukan atau tidak sesuai karena algoritma binary search yang digunakan berfokus pada data yang diinputkan secara ascending. Mengubah kode program binary search agar hasil ditemukan dengan data descending (terbesar ke terkecil). Mengubah method findBinarySearch seperti kode dibawah ini:

```
    int findBinarySearch (double cari, int left, int right){
        int mid;
        if (right >= left) {
            mid = (left + right) /2;
            if (cari == listMhs[mid].ipk) {
                return (mid);
            }
            else if (listMhs[mid].ipk < cari){
                return findBinarySearch(cari, mid-1, right);
            }
            else {
                return findBinarySearch(cari, left, mid+1);
            }
        }
        return -1;
    }
}
```

5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

   Memodifikasi atribut dan menambahkan konstruktor pada class MahasiswaBerprestasi06

```
int jumlahMhs = 5;
Mahasiswa06[] listMhs = new Mahasiswa06[jumlahMhs];
MahasiswaBerprestasi06(){}
MahasiswaBerprestasi06(int jumlahMhs){    Constructor i
    this.jumlahMhs = jumlahMhs;
    listMhs = new Mahasiswa06[jumlahMhs];
}
```

   Menggunakan Scanner untuk menerima input jumlah mahasiswa dan menggunakan konstruktor berparameter pada objek class MahasiswaBerprestasi11 serta menggunakan method insertionSort untuk mengurutkan data mahasiswa yang telah diinput secara descending

   pada class MahasiswaDemo06

```
for (int i = 0; i < jumlahMhs; i++) {
    System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));
    System.out.print(s:"NIM   : ");
    String nim = sc.nextLine();
    System.out.print(s:"Nama  : ");
    String nama = sc.nextLine();
    System.out.print(s:"Kelas : ");
    String kelas = sc.nextLine();
    System.out.print(s:"IPK   : ");
    double ipk = sc.nextDouble();
    sc.nextLine();
    System.out.println(x:"-----------------------------------");
    Mahasiswa06 mhs = new Mahasiswa06(nim, nama, kelas, ipk);
    list.tambah(mhs);
}
```

Hasil dari kode program

```
Masukkan jumlah mahasiswa: 6
Masukkan Data Mahasiswa ke-1
NIM   : 111
Nama  : Adi
Kelas : 2
IPK   : 3.4
---------------------------------
Masukkan Data Mahasiswa ke-2
NIM   : 222
Nama  : Ila
Kelas : 2
IPK   : 3.7
---------------------------------
Masukkan Data Mahasiswa ke-3
NIM   : 333
Nama  : Lia
Kelas : 2
IPK   : 3.2
---------------------------------
Masukkan Data Mahasiswa ke-4
NIM   : 444
Nama  : Susi
Kelas : 2
IPK   : 3.5
---------------------------------
Masukkan Data Mahasiswa ke-5
NIM   : 555
Nama  : Anita
Kelas : 2
IPK   : 3.3
---------------------------------
Masukkan Data Mahasiswa ke-6
NIM   : 666
Nama  : Tio
Kelas : 2
IPK   : 3.9
---------------------------------
```

```
Data mahasiswa:
Nama: Tio
NIM: 666
Kelas: 2
IPK: 3.9
--------------------------------
Nama: Ila
NIM: 222
Kelas: 2
IPK: 3.7
--------------------------------
Nama: Susi
NIM: 444
Kelas: 2
IPK: 3.5
--------------------------------
Nama: Adi
NIM: 111
Kelas: 2
IPK: 3.4
--------------------------------
Nama: Anita
NIM: 555
Kelas: 2
IPK: 3.3
--------------------------------
Nama: Lia
NIM: 333
Kelas: 2
IPK: 3.2
--------------------------------
```

```
-----------------------------------
Pencarian Data
-----------------------------------
Masukkan IPK mahasiswa yang dicari:
IPK : 3.7
-----------------------------------
Menggunakan Binary Search
-----------------------------------
Data mahasiswa dengan IPK : 3.7 ditemukan pada indeks 1
NIM      : 222
Nama     : Ila
Kelas    : 2
IPK      : 3.7
```

LATIHAN

1. Kode program

Dosen06

```
public class Dosen06 {
    String kode, nama;
    Boolean jenisKelamin;
    int usia;
    Dosen06(String kd,String name, Boolean jk,int age){
    kode = kd;
    nama = name;
    jenisKelamin = jk;
    usia = age;
    }
    void tampil(){
    System.out.println("Kode Dosen :"+kode);
    System.out.println("Nama :"+nama);
    System.out.println("Jenis Kelamin :"+(jenisKelamin?"Perempuan":"Laki-
laki"));
    System.out.println("Usia :"+usia);
    }
   }
```

DataDosen06

```
public class DataDosen06 {
    Dosen06[] dataDosen = new Dosen06[10];
    int idx;

    void tambah(Dosen06 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Dosen06 dosen : dataDosen) {
            dosen.tampil();
            System.out.println("--------------------------------");
        }
    }

    void SortingASC() {
        for (int i = 0; i < dataDosen.length - 1; i++) {
            for (int j = 1; j < dataDosen.length - i; j++) {
                if (dataDosen[j].usia < dataDosen[j - 1].usia) {
                    Dosen06 tmp = dataDosen[j];
                    dataDosen[j] = dataDosen[j - 1];
                    dataDosen[j - 1] = tmp;
                }
            }
        }
    }
```

```
    int PencarianDataSequential11(double cari) {
        int count = 0;
        int posisi = -1;
        for (int i = 0; i < dataDosen.length; i++) {
            if (dataDosen[i].usia == cari) {
                posisi = i;
                count++;
            }
        }
        if (count > 1) posisi = -2;
        return posisi;
    }

    int PencarianDataBinary11(int cari, int left, int right) {
        if (left > right) return -1;
        int mid = (left + right) / 2;
        if (cari == dataDosen[mid].usia) {
            if (cari == dataDosen[mid + 1].usia || cari == dataDosen[mid -
1].usia) {
                return -2;
            }
            return mid;
        }
        if (cari < dataDosen[mid].usia) {
            return PencarianDataBinary11(cari, left, mid - 1);
        } else {
            return PencarianDataBinary11(cari, mid + 1, right);
        }
    }

    void tampilDataSearch(int x, int pos) {
        if (pos == -1) {
            System.out.println("Data dosen dengan usia " + x + " tidak
ditemukan");
        } else if (pos == -2) {
            System.out.println("Data Dosen Muncul lebih dari 1 kali");
        } else {
            System.out.println("Data Dosen dengan Usia: " + x + " ditemukan
pada indeks " + pos);
            dataDosen[pos].tampil();
        }
    }
}
```

DosenMain06

```
import java.util.Scanner;

public class DosenMain06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen06 list = new DataDosen06();

        for (int i = 0; i < list.dataDosen.length; i++) {
            System.out.println("Masukkan Data Dosen ke-" + (i + 1));
            System.out.print("Kode Dosen : ");
            String kode = sc.nextLine();
            System.out.print("Nama : ");
            String nama = sc.nextLine();
            System.out.print("Jenis Kelamin (L / P) : ");
            String jenisKelamin = sc.nextLine();
            boolean jk = jenisKelamin.equalsIgnoreCase("L") ? false : true;
            System.out.print("Usia : ");
            int usia = sc.nextInt();
            sc.nextLine();
            System.out.println("---------------------------------");

            Dosen06 dosen = new Dosen06(kode, nama, jk, usia);
            list.tambah(dosen);
        }

        System.out.println("Data dosen: ");
        list.tampil();
        System.out.println("---------------------------------");

        System.out.println("Pencarian Data Dosen dengan Sequential Search");
        System.out.println("---------------------------------");
        System.out.print("Masukkan Usia Dosen yang dicari : ");
        int cari = sc.nextInt();
        System.out.println("---------------------------------");
        int posisi = list.PencarianDataSequential11(cari);
        list.tampilDataSearch(cari, posisi);
        System.out.println("---------------------------------");

        System.out.println("Data dosen terurut ASC: ");
        list.SortingASC();
        list.tampil();
        System.out.println("---------------------------------");

        System.out.println("Pencarian Data Dosen dengan Binary Search");
        System.out.println("---------------------------------");
        System.out.print("Masukkan Usia Dosen yang dicari : ");
        cari = sc.nextInt();
        System.out.println("---------------------------------");
        posisi = list.PencarianDataBinary11(cari, 0, list.dataDosen.length - 1);
        list.tampilDataSearch(cari, posisi);
    }
}
```

2. Hasil dari kode program

```
Masukkan Data Dosen ke-1
Kode Dosen           : 123
Nama                 : burhan
Jenis Kelamin (L / P) : L
Usia                 : 12
--------------------------------
Masukkan Data Dosen ke-2
Kode Dosen           : 124
Nama                 : Udin
Jenis Kelamin (L / P) : L
Usia                 : 13
--------------------------------
Masukkan Data Dosen ke-3
Kode Dosen           : 125
Nama                 : padli
Jenis Kelamin (L / P) : L
Usia                 : 14
--------------------------------
Masukkan Data Dosen ke-4
Kode Dosen           : 126
Nama                 : pudi
Jenis Kelamin (L / P) : L
Usia                 : 15
--------------------------------
Masukkan Data Dosen ke-5
Kode Dosen           : 127
Nama                 : aldi
Jenis Kelamin (L / P) : L
Usia                 : 16
--------------------------------
Masukkan Data Dosen ke-6
Kode Dosen           : 128
Nama                 : ical
Jenis Kelamin (L / P) : L
Usia                 : 17
--------------------------------
Masukkan Data Dosen ke-7
Kode Dosen           : 129
Nama                 : vita
Jenis Kelamin (L / P) : P
Usia                 : 18
--------------------------------
Masukkan Data Dosen ke-8
Kode Dosen           : 130
Nama                 : siti
Jenis Kelamin (L / P) : P
Usia                 : 34
--------------------------------
Masukkan Data Dosen ke-9
Kode Dosen           : 131
Nama                 : finaz
Jenis Kelamin (L / P) : P
Usia                 : 25
--------------------------------
Masukkan Data Dosen ke-10
Kode Dosen           : 132
Nama                 : nunung
Jenis Kelamin (L / P) : P
Usia                 : 28
--------------------------------
```

```
---------------------------------
Pencarian Data Dosen dengan Sequential Search
---------------------------------
Masukkan Usia Dosen yang dicari : 25
---------------------------------
Data Dosen dengan Usia : 25 ditemukan pada indeks 8
Kode Dosen    :131
Nama          :finaz
Jenis Kelamin :Perempuan
Usia          :25
---------------------------------
Data dosen terurut ASC:
Kode Dosen    :123
Nama          :burhan
Jenis Kelamin :Laki-laki
Usia          :12
--------------------------------
Kode Dosen    :124
Nama          :Udin
Jenis Kelamin :Laki-laki
Usia          :13
--------------------------------
Kode Dosen    :125
Nama          :padli
Jenis Kelamin :Laki-laki
Usia          :14
--------------------------------
Kode Dosen    :126
Nama          :pudi
Jenis Kelamin :Laki-laki
Usia          :15
--------------------------------
Kode Dosen    :127
Nama          :aldi
Jenis Kelamin :Laki-laki
Usia          :16
--------------------------------
Kode Dosen    :128
Nama          :ical
Jenis Kelamin :Laki-laki
Usia          :17
--------------------------------
Kode Dosen    :129
Nama          :vita
Jenis Kelamin :Perempuan
Usia          :18
--------------------------------
Kode Dosen    :131
Nama          :finaz
Jenis Kelamin :Perempuan
Usia          :25
--------------------------------
Kode Dosen    :132
Nama          :nunung
Jenis Kelamin :Perempuan
Usia          :28
--------------------------------
Kode Dosen    :130
Nama          :siti
Jenis Kelamin :Perempuan
Usia          :34
--------------------------------
```

```
---------------------------------
Pencarian Data Dosen dengan Binary Search
---------------------------------
Masukkan Usia Dosen yang dicari : 40
---------------------------------
Data dosen dengan usia 40 tidak ditemukan
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```