

BAB 3

PENGELOLAAN DATA PADA ARRAY : PENAMBAHAN DAN PENGHAPUSAN DATA

Tujuan

Mahasiswa dapat menambah data baru ke dalam larik dan dapat menghapus data tertentu dari dalam larik.

3.1. Menambah Data Baru Ke Dalam Larik

Menambah data baru ke dalam larik dapat dilakukan pada bagian depan, bagian tengah, dan bagian belakang.

Pada bagian ini akan dicontohkan data seperti tabel berikut ini.

	<i>.nama</i>	<i>.alamat</i>	<i>.umur</i>	<i>.j_kel</i>	<i>.hobi []</i>	<i>.IPK</i>
<i>biodataMahasiswa [0]</i>	AgungBP	Jakarta	28	L	<i>.hobi[0]</i> musik <i>.hobi[1]</i> mancing <i>.hobi[2]</i> touring	3.5
<i>biodataMahasiswa [1]</i>	Rulieta	KualaKapas	17	P	<i>.hobi[0]</i> tari <i>.hobi[1]</i> rajut <i>.hobi[2]</i> jahit	3
<i>biodataMahasiswa [2]</i>	Kayra	Yogyakarta	15	P	<i>.hobi[0]</i> ballet <i>.hobi[1]</i> renang <i>.hobi[2]</i> taekwondo	3.4
<i>biodataMahasiswa [3]</i>	Elnathan	Yogyakarta	13	L	<i>.hobi[0]</i> renang <i>.hobi[1]</i> main <i>.hobi[2]</i> youtube	3.4
<i>biodataMahasiswa [4]</i>	Niken	Magelang	35	P	<i>.hobi[0]</i> jualan <i>.hobi[1]</i> menanam <i>.hobi[2]</i> arisan	2.5
<i>biodataMahasiswa [5]</i>	Liwin	Palangkaraya	35	P	<i>.hobi[0]</i> memasak <i>.hobi[1]</i> menanam <i>.hobi[2]</i> berkebun	2.7
<i>biodataMahasiswa [6]</i>	Satrio	Semarang	26	L	<i>.hobi[0]</i> karate <i>.hobi[1]</i> band <i>.hobi[2]</i> driving	3.1
<i>biodataMahasiswa [7]</i>	Dion	Bantul	22	L	<i>.hobi[0]</i> makelaran <i>.hobi[1]</i> berdagang <i>.hobi[2]</i> karaoke	2.9
<i>biodataMahasiswa [8]</i>	Fifin	Purwokerto	32	P	<i>.hobi[0]</i> mengajar <i>.hobi[1]</i> menanam <i>.hobi[2]</i> bernyanyi	3.1
<i>biodataMahasiswa [9]</i>	Hermon	Banjarmasin	26	L	<i>.hobi[0]</i> bertukang <i>.hobi[1]</i> menggambar <i>.hobi[2]</i> bertualang	3.3

3.1.1. Algoritma Penambahan data di depan

Penambahan data di bagian depan dilakukan dengan menciptakan ruang kosong pada larik paling atas (larik ke-0) dan memasukkan data baru pada ruang kosong tersebut.

Proses ini harus didahului dengan proses penggeseran data secara berturut-turut mulai dari data ke-9 (terakhir) sampai dengan data ke-0 sebanyak satu langkah ke bawah sehingga akan menciptakan ruang kosong pada larik bagian ke-0, sementara data yang bergeser akan menempati larik ke-1 hingga larik ke-10. Setelah proses penggeseran data selesai nilai N (banyaknya data) harus tambah dengan 1.

	.nama	.alamat	.umur	.jkel	.hobi []	.IPK
<i>biodataMahasiswa [0]</i>					.hobi[0] .hobi[1] .hobi[2]	
<i>biodataMahasiswa [1]</i>	AgungBP	Jakarta	28	L	.hobi[0] musik .hobi[1] mancing .hobi[2] touring	3.5
<i>biodataMahasiswa [2]</i>	Rulieta	KualaKapuas	17	P	.hobi[0] tari .hobi[1] rajut .hobi[2] jahit	3
<i>biodataMahasiswa [3]</i>	Kayra	Yogyakarta	15	P	.hobi[0] ballet .hobi[1] renang .hobi[2] taekwondo	3.4
<i>biodataMahasiswa [4]</i>	Elnathan	Yogyakarta	13	L	.hobi[0] renang .hobi[1] main .hobi[2] youtube	3.4
<i>biodataMahasiswa [5]</i>	Niken	Magelang	35	P	.hobi[0] jualan .hobi[1] menanam .hobi[2] arisan	2.5
<i>biodataMahasiswa [6]</i>	Liwin	Palangkaraya	35	P	.hobi[0] memasak .hobi[1] menanam .hobi[2] berkebun	2.7
<i>biodataMahasiswa [7]</i>	Satrio	Semarang	26	L	.hobi[0] karate .hobi[1] band .hobi[2] driving	3.1
<i>biodataMahasiswa [8]</i>	Dion	Bantul	22	L	.hobi[0] makelaran .hobi[1] berdagang .hobi[2] karaoke	2.9
<i>biodataMahasiswa [9]</i>	Fifin	Purwokerto	32	P	.hobi[0] mengajar .hobi[1] menanam .hobi[2] bernyanyi	3.1
<i>biodataMahasiswa [10]</i>	Hermon	Banjarmasin	26	L	.hobi[0] bertukang .hobi[1] menggambar .hobi[2] bertualang	3.3

← ruang kosong pada larik ke-0, yg terbentuk setelah semua data digeser turun ke bawah

kondisi data setelah mengalami pergeseran turun 1 langkah ke bawah

Untuk mengimplementasikan proses penambahan data baru di bagian depan anda dapat memperhatikan program 3.1 berikut ini.

```
//-----
//---      Fungsi untuk Menambah Data Di Depan      ---
//-----
public static void tambahDataDiDepan(formatBiodata biodataMahasiswa[])
{
    //bagian membuat record sementara untuk menampung data baru-----
    formatBiodata biodataMahasiswaBaru = new formatBiodata();

    //bagian entri data baru ke penyimpanan sementara-----
    Scanner masukan = new Scanner(System.in);
    int bacaTombol=0;
    System.out.print("Silakan masukkan nama anda : ");
    biodataMahasiswaBaru.nama = masukan.next();
    System.out.print("Silakan masukkan alamat anda : ");
    biodataMahasiswaBaru.alamat = masukan.next();
    System.out.print("Silakan masukkan umur anda : ");
    biodataMahasiswaBaru.umur = masukan.nextInt();
}
```

```

        System.out.print("Silakan masukkan Jenis Kelamin anda : ");
        try
        {   bacaTombol = System.in.read();
        }
        catch(java.io.IOException e)
        {
        }
        biodataMahasiswaBaru.jekel = (char)bacaTombol;

        System.out.println("Silakan masukkan hobi (maks 3) : ");
        System.out.print("hobi ke-0 : ");
        biodataMahasiswaBaru.hobi[0] = masukan.next();
        System.out.print("hobi ke-1 : ");
        biodataMahasiswaBaru.hobi[1] = masukan.next();
        System.out.print("hobi ke-2 : ");
        biodataMahasiswaBaru.hobi[2] = masukan.next();

        System.out.print("Silakan masukkan IPK anda : ");
        biodataMahasiswaBaru.ipk = masukan.nextFloat();

        //bagian menggeser isi larik mulai dari Belakang s/d 0 selangkah ke bawah
        for (int i=N-1; i>= 0; i--)
        {   biodataMahasiswa[i+1] = biodataMahasiswa[i];
        }

        //bagian memindahkan data baru ke larik ke-0-----
        biodataMahasiswa[0] = biodataMahasiswaBaru;

        //memperbaharui banyaknya data (N), banyaknya data bertambah satu-----
        N++;
    }

```

Program 3.1

3.1.2. Algoritma Penambahan data di tengah

Penambahan data di bagian tengah dilakukan dengan menciptakan ruang kosong pada larik ke-T di mana T adalah posisi target kemudian memasukkan data baru pada ruang kosong tersebut.

Proses ini harus didahului dengan proses penggeseran data secara berturut-turut mulai dari data ke-9 (terakhir) sampai dengan data ke-T sebanyak satu langkah ke bawah sehingga akan menciptakan ruang kosong pada larik ke-T, sementara data yang bergeser akan menempati larik ke-T+1 hingga larik ke-10. Setelah proses penggeseran data selesai nilai N (banyaknya data) harus tambah dengan 1.

	.nama	.alamat	.umur	.jkel	.hobi []	.IPK	
<i>biodataMahasiswa [0]</i>	AgungBP	Jakarta	28	L	.hobi[0] musik .hobi[1] mancing .hobi[2] touring	3.5	kondisi data yang tetap, tidak mengalami pergeseran
<i>biodataMahasiswa [1]</i>	Rulieta	KualaKapuas	17	P	.hobi[0] tari .hobi[1] rajut .hobi[2] jahit	3	
<i>biodataMahasiswa [2]</i>	Kayra	Yogyakarta	15	P	.hobi[0] ballet .hobi[1] renang .hobi[2] taekwondo	3.4	
<i>biodataMahasiswa [3]</i>	Elnathan	Yogyakarta	13	L	.hobi[0] renang .hobi[1] main .hobi[2] youtube	3.4	
<i>biodataMahasiswa [4]</i>					.hobi[0] .hobi[1] .hobi[2]		ruang kosong pada larik ke-T, yg terbentuk setelah sebagian data digeser turun ke bawah
<i>biodataMahasiswa [5]</i>	Niken	Magelang	35	P	.hobi[0] jualan .hobi[1] menanam .hobi[2] arisan	2.5	
<i>biodataMahasiswa [6]</i>	Liwin	Palangkaraya	35	P	.hobi[0] memasak .hobi[1] menanam .hobi[2] berkebun	2.7	kondisi data setelah mengalami pergeseran turun 1 langkah ke bawah
<i>biodataMahasiswa [7]</i>	Satrio	Semarang	26	L	.hobi[0] karate .hobi[1] band .hobi[2] driving	3.1	
<i>biodataMahasiswa [8]</i>	Dion	Bantul	22	L	.hobi[0] makelaran .hobi[1] berdagang .hobi[2] karaoke	2.9	
<i>biodataMahasiswa [9]</i>	Fifin	Purwokerto	32	P	.hobi[0] mengajar .hobi[1] menanam .hobi[2] bernyanyi	3.1	
<i>biodataMahasiswa [10]</i>	Hermon	Banjarmasin	26	L	.hobi[0] bertukang .hobi[1] menggambar .hobi[2] bertualang	3.3	

Untuk mengimplementasikan proses penambahan data baru di bagian tengah anda dapat memperhatikan program 3.2 berikut ini.

```
//-----
//---      Fungsi untuk Menambah Data Di Tengah      ---
//-----
public static void tambahDataDiTengah(formatBiodata biodataMahasiswa[])
{
    //bagian membuat record sementara untuk menampung data baru-----
    formatBiodata biodataMahasiswaBaru = new formatBiodata();

    //bagian entri data baru ke penyimpanan sementara-----
    Scanner masukan = new Scanner(System.in);
    int bacaTombol=0;
    System.out.print("Silakan masukkan nama anda : ");
    biodataMahasiswaBaru.nama = masukan.next();
    System.out.print("Silakan masukkan alamat anda : ");
    biodataMahasiswaBaru.alamat = masukan.next();
    System.out.print("Silakan masukkan umur anda : ");
    biodataMahasiswaBaru.umur = masukan.nextInt();
    System.out.print("Silakan masukkan Jenis Kelamin anda : ");
    try
    {
        bacaTombol = System.in.read();
    }
    catch(java.io.IOException e)
    {
    }
    biodataMahasiswaBaru.jkel = (char)bacaTombol;

    System.out.println("Silakan masukkan hobi (maks 3) : ");
    System.out.print("hobi ke-0 : ");
    biodataMahasiswaBaru.hobi[0] = masukan.next();
    System.out.print("hobi ke-1 : ");
    biodataMahasiswaBaru.hobi[1] = masukan.next();
    System.out.print("hobi ke-2 : ");
```

```

biodataMahasiswaBaru.hobi[2] = masukan.next();

System.out.print("Silakan masukkan IPK anda : ");
biodataMahasiswaBaru.ipk = masukan.nextFloat();

//bagian menentukan posisi target T -----
int T;
System.out.print("Pada posisi ke berapa data akan dimasukkan ? : ");
T = masukan.nextInt();

//bagian menggeser isi larik mulai dari Belakang s/d T selangkah ke belakang
for (int i=N-1; i>= T; i--)
{
    biodataMahasiswa[i+1] = biodataMahasiswa[i];
}

//bagian memindahkan data baru ke larik ke-T-----
biodataMahasiswa[T] = biodataMahasiswaBaru;

//memperbaharui banyaknya data (N), banyaknya data bertambah satu-----
N++;
}

```

Program 3.2

3.1.3. Algoritma Penambahan data di belakang

Penambahan data di belakang dilakukan dengan memasukkan data yang baru pada larik ke-N di mana N adalah posisi paling akhir. Pada proses menambah data di belakang, kita tidak perlu melakukan penggeseran data yang telah ada dalam larik melainkan cukup dengan menaikkan nilai N nya saja (banyaknya data) dengan 1.

	.nama	.alamat	.umur	.jkel	.hobi []	.IPK
<i>biodataMahasiswa [0]</i>	AgungBP	Jakarta	28	L	.hobi[0] musik .hobi[1] mancing .hobi[2] touring	3.5
<i>biodataMahasiswa [1]</i>	Rulieta	KualaKapas	17	P	.hobi[0] tari .hobi[1] rajut .hobi[2] jahit	3
<i>biodataMahasiswa [2]</i>	Kayra	Yogyakarta	15	P	.hobi[0] ballet .hobi[1] renang .hobi[2] taekwondo	3.4
<i>biodataMahasiswa [3]</i>	Elnathan	Yogyakarta	13	L	.hobi[0] renang .hobi[1] main .hobi[2] youtube	3.4
<i>biodataMahasiswa [4]</i>	Niken	Magelang	35	P	.hobi[0] jualan .hobi[1] menanam .hobi[2] arisan	2.5
<i>biodataMahasiswa [5]</i>	Liwin	Palangkaraya	35	P	.hobi[0] memasak .hobi[1] menanam .hobi[2] berkebun	2.7
<i>biodataMahasiswa [6]</i>	Satrio	Semarang	26	L	.hobi[0] karate .hobi[1] band .hobi[2] driving	3.1
<i>biodataMahasiswa [7]</i>	Dion	Bantul	22	L	.hobi[0] makelaran .hobi[1] berdagang .hobi[2] karaoke	2.9
<i>biodataMahasiswa [8]</i>	Fifin	Purwokerto	32	P	.hobi[0] mengajar .hobi[1] menanam .hobi[2] bernyanyi	3.1
<i>biodataMahasiswa [9]</i>	Hermon	Banjarmasin	26	L	.hobi[0] bertukang .hobi[1] menggambar .hobi[2] bertualang	3.3
<i>biodataMahasiswa [10]</i>					.hobi[0] .hobi[1] .hobi[2]	

kondisi data yang tetap, tidak mengalami penggeseran

← ruang kosong di larik ke-N, belum pernah ditempati oleh siapapun

Untuk mengimplementasikan proses penambahan data baru di bagian belakang anda dapat memperhatikan program 3.3 berikut ini.

```
//-----  
//--- Fungsi untuk Menambah Data Di Belakang ---  
//-----  
public static void tambahDataDiBelakang(formatBiodata biodataMahasiswa[])  
{  
    //bagian membuat record sementara untuk menampung data baru-----  
    formatBiodata biodataMahasiswaBaru = new formatBiodata();  
  
    //bagian entri data baru ke penyimpanan sementara-----  
    Scanner masukan = new Scanner(System.in);  
    int bacaTombol=0;  
    System.out.print("Silakan masukkan nama anda : ");  
    biodataMahasiswaBaru.nama = masukan.next();  
    System.out.print("Silakan masukkan alamat anda : ");  
    biodataMahasiswaBaru.alamat = masukan.next();  
    System.out.print("Silakan masukkan umur anda : ");  
    biodataMahasiswaBaru.umur = masukan.nextInt();  
    System.out.print("Silakan masukkan Jenis Kelamin anda : ");  
    try  
    { bacaTombol = System.in.read();  
    }  
    catch(java.io.IOException e)  
    {  
    }  
    biodataMahasiswaBaru.jekel = (char)bacaTombol;  
  
    System.out.println("Silakan masukkan hobi (maks 3) : ");  
    System.out.print("hobi ke-0 : ");  
    biodataMahasiswaBaru.hobi[0] = masukan.next();  
    System.out.print("hobi ke-1 : ");  
    biodataMahasiswaBaru.hobi[1] = masukan.next();  
    System.out.print("hobi ke-2 : ");  
    biodataMahasiswaBaru.hobi[2] = masukan.next();  
  
    System.out.print("Silakan masukkan IPK anda : ");  
    biodataMahasiswaBaru.ipk = masukan.nextFloat();  
  
    //bagian memindahkan data baru ke larik ke-N-----  
    biodataMahasiswa[N] = biodataMahasiswaBaru;  
  
    //memperbaharui banyaknya data (N), banyaknya data bertambah satu----  
    N++;  
}
```

Program 3.3

3.2. Menghapus Data

Menghapus data dari dalam larik dapat dilakukan di bagian depan, bagian tengah, dan bagian belakang.

3.2.1. Algoritma Penghapusan data di depan

Menghapus data di bagian depan dilakukan dengan cara menimpa larik paling depan (larik ke-0) dengan larik ke-1, dilanjutkan dengan menimpa larik ke-1 dengan larik ke-2, ke-2 dengan ke-3, dan seterusnya. Proses ini sama dengan melakukan penggeseran secara berturut-turut data dimulai ke-0 sampai dengan data terakhir sebanyak satu langkah ke depan. Setelah proses penggeseran selesai nilai N (banyaknya data) akan dikurangi 1.

	.nama	.alamat	.umur	.jkel	.hobi []	.IPK	
<i>biodataMahasiswa [0]</i>	Ending	Banjarbaru	15	L	.hobi[0] desain .hobi[1] hiking .hobi[2] driving	2.5	← data yang akan dihapus pada larik ke-0 (akan ditimpa oleh data di bawahnya)
<i>biodataMahasiswa [1]</i>	AgungBP	Jakarta	28	L	.hobi[0] musik .hobi[1] mancing .hobi[2] touring	3.5	} data-data yang harus digeser naik 1 langkah ke atas
<i>biodataMahasiswa [2]</i>	Rulieta	KualaKapas	17	P	.hobi[0] tari .hobi[1] rajut .hobi[2] jahit	3	
<i>biodataMahasiswa [3]</i>	Kayra	Yogyakarta	15	P	.hobi[0] ballet .hobi[1] renang .hobi[2] taekwondo	3.4	
<i>biodataMahasiswa [4]</i>	Elnathan	Yogyakarta	13	L	.hobi[0] renang .hobi[1] main .hobi[2] youtube	3.4	
<i>biodataMahasiswa [5]</i>	Niken	Magelang	35	P	.hobi[0] jualan .hobi[1] menanam .hobi[2] arisan	2.5	
<i>biodataMahasiswa [6]</i>	Liwin	Palangkaraya	35	P	.hobi[0] memasak .hobi[1] menanam .hobi[2] berkebun	2.7	
<i>biodataMahasiswa [7]</i>	Satrio	Semarang	26	L	.hobi[0] karate .hobi[1] band .hobi[2] driving	3.1	
<i>biodataMahasiswa [8]</i>	Dion	Bantul	22	L	.hobi[0] makelaran .hobi[1] berdagang .hobi[2] karaoke	2.9	
<i>biodataMahasiswa [9]</i>	Fifin	Purwokerto	32	P	.hobi[0] mengajar .hobi[1] menanam .hobi[2] bernyanyi	3.1	
<i>biodataMahasiswa [10]</i>	Hermon	Banjarmasin	26	L	.hobi[0] bertukang .hobi[1] menggambar .hobi[2] bertualang	3.3	

Untuk mengimplementasikan proses penghapusan data di depan anda dapat memperhatikan program 3.4 berikut ini.

```
//-----
//--- Fungsi untuk Menghapus Data Di Depan ---
//-----
public static void hapusDataDiDepan(formatBiodata biodataMahasiswa[])
{
    //bagian menggeser isi larik mulai dari 0 - Belakang selangkah ke depan
    for (int i=0; i<= N-2; i++)
    { biodataMahasiswa[i] = biodataMahasiswa[i+1];
    }
    System.out.println("Proses menghapus data ke-0 selesai.");

    //memperbaharui banyaknya data (N), banyaknya data berkurang satu-----
    N--;
}
```

Program 3.4

3.2.2. Algoritma Penghapusan data di tengah

Menghapus data di bagian tengah dilakukan dengan menentukan posisi target data yang ingin dihapus (T). Kemudian data yang akan dihapus pada larik ke-T ditimpa dengan larik ke-T+1, dilanjutkan dengan menimpa larik ke-T+1 dengan larik ke-T+2, larik ke-T+2 dengan ke-T+3, dan seterusnya. Proses ini sama dengan menggeser data mulai dari data ke-T+1 sampai dengan data terakhir sebanyak satu langkah ke atas.

	.nama	.alamat	.umur	.jkel	.hobi []	.IPK	
<i>biodataMahasiswa [0]</i>	AgungBP	Jakarta	28	L	.hobi[0] musik .hobi[1] mancing .hobi[2] touring	3.5	kondisi data yang tetap, tidak mengalami pergeseran
<i>biodataMahasiswa [1]</i>	Rulieta	KualaKapuas	17	P	.hobi[0] tari .hobi[1] rajut .hobi[2] jahit	3	
<i>biodataMahasiswa [2]</i>	Kayra	Yogyakarta	15	P	.hobi[0] ballet .hobi[1] renang .hobi[2] taekwondo	3.4	
<i>biodataMahasiswa [3]</i>	Elnathan	Yogyakarta	13	L	.hobi[0] renang .hobi[1] main .hobi[2] youtube	3.4	
<i>biodataMahasiswa [4]</i>	Ending	Banjarbaru	15	L	.hobi[0] desain .hobi[1] hiking .hobi[2] driving	2.5	data yang akan dihapus pada larik ke-T (akan ditimpa oleh data di bawahnya)
<i>biodataMahasiswa [5]</i>	Niken	Magelang	35	P	.hobi[0] jualan .hobi[1] menanam .hobi[2] arisan	2.5	
<i>biodataMahasiswa [6]</i>	Liwin	Palangkaraya	35	P	.hobi[0] memasak .hobi[1] menanam .hobi[2] berkebun	2.7	data-data yang harus digeser naik 1 langkah ke atas
<i>biodataMahasiswa [7]</i>	Satrio	Semarang	26	L	.hobi[0] karate .hobi[1] band .hobi[2] driving	3.1	
<i>biodataMahasiswa [8]</i>	Dion	Bantul	22	L	.hobi[0] makelaran .hobi[1] berdagang .hobi[2] karaoke	2.9	
<i>biodataMahasiswa [9]</i>	Fifin	Purwokerto	32	P	.hobi[0] mengajar .hobi[1] menanam .hobi[2] bernyanyi	3.1	
<i>biodataMahasiswa [10]</i>	Hermon	Banjarmasin	26	L	.hobi[0] bertukang .hobi[1] menggambar .hobi[2] bertualang	3.3	

Untuk mengimplementasikan proses penghapusan data di tengah anda dapat memperhatikan program 3.5 berikut ini.

```
//-----
//--- Fungsi untuk Menghapus Data Di Tengah ---
//-----
public static void hapusDataDiTengah(formatBiodata biodataMahasiswa[])
{
    //bagian menentukan posisi target T -----
    Scanner masukan = new Scanner(System.in);
    int T;
    System.out.print("Tuliskan posisi data yang akan dibapus : ");
    T = masukan.nextInt();

    //bagian menggeser isi larik mulai dari T - Belakang selangkah ke depan
    for (int i=T; i<= N-2; i++)
    {
        biodataMahasiswa[i] = biodataMahasiswa[i+1];
    }
    System.out.println("Proses menghapus data ke-" + T + " selesai.");

    //memperbaharui banyaknya data (N), banyaknya data berkurang satu-----
    N--;
}
```

Program 3.5

3.2.3. Algoritma Penghapusan data di belakang

Menghapus data di bagian belakang dilakukan tidak perlu ada penggeseran data. Proses menghapus data di bagian belakang hanya dilakukan dengan memotong data terakhir dengan cara mengurangi N dengan 1 (N--).

	.nama	.alamat	.umur	.jkel	.hobi []	.IPK	
<i>biodataMahasiswa [0]</i>	AgungBP	Jakarta	28	L	.hobi[0] musik .hobi[1] mancing .hobi[2] touring	3.5	} kondisi data yang tetap, tidak mengalami pergeseran
<i>biodataMahasiswa [1]</i>	Rulieta	KualaKapuas	17	P	.hobi[0] tari .hobi[1] rajut .hobi[2] jahit	3	
<i>biodataMahasiswa [2]</i>	Kayra	Yogyakarta	15	P	.hobi[0] ballet .hobi[1] renang .hobi[2] taekwondo	3.4	
<i>biodataMahasiswa [3]</i>	Elnathan	Yogyakarta	13	L	.hobi[0] renang .hobi[1] main .hobi[2] youtube	3.4	
<i>biodataMahasiswa [4]</i>	Niken	Magelang	35	P	.hobi[0] jualan .hobi[1] menanam .hobi[2] arisan	2.5	
<i>biodataMahasiswa [5]</i>	Liwin	Palangkaraya	35	P	.hobi[0] memasak .hobi[1] menanam .hobi[2] berkebun	2.7	
<i>biodataMahasiswa [6]</i>	Satrio	Semarang	26	L	.hobi[0] karate .hobi[1] band .hobi[2] driving	3.1	
<i>biodataMahasiswa [7]</i>	Dion	Bantul	22	L	.hobi[0] makelaran .hobi[1] berdagang .hobi[2] karaoke	2.9	
<i>biodataMahasiswa [8]</i>	Fifin	Purwokerto	32	P	.hobi[0] mengajar .hobi[1] menanam .hobi[2] bernyanyi	3.1	
<i>biodataMahasiswa [9]</i>	Hermon	Banjarmasin	26	L	.hobi[0] bertukang .hobi[1] menggambar .hobi[2] bertualang	3.3	← data yang akan dihapus pada larik ke-N (dgn cara dipotong)
<i>biodataMahasiswa [10]</i>	Ending	Banjarbaru	15	L	.hobi[0] desain .hobi[1] hiking .hobi[2] driving	2.5	

Untuk mengimplementasikan proses penghapusan data di belakang anda dapat memperhatikan program 3.6 berikut ini.

```
//-----
//--- Fungsi untuk Menghapus Data Di Belakang ---
//-----
public static void hapusDataDiBelakang(formatBiodata biodataMahasiswa[])
{
    System.out.println("Proses menghapus data paling akhir selesai.");

    //memperbaharui banyaknya data (N), banyaknya data berkurang satu-----
    N--;
}
```

Program 3.6

Pada program 3.7 di bawah ini anda dapat melihat potongan/cuplikan program yang pernah anda tulis pada percobaan-percobaan sebelumnya. Selanjutnya anda dapat menambahkan fungsi-fungsi untuk menambah data dan juga menghapus data.

```
import java.util.Scanner;
class formatBiodata
{ //bagian deklarasi struktur record -----
    String nama;
    String alamat;
    int umur;
    char jkel;
    String hobi[] = new String[3];
    float ipk;
}

class menambahData
{ public static int N=0;
```

```

//-----
//--- Fungsi untuk mengentri data ke dalam Larik ---
//-----
public static void ngentriData(formatBiodata biodataMahasiswa[])
{
    ....;
}

//-----
//--- Fungsi untuk... ---
//-----
public static void .....() //←fungsi tambah/hapus data di depan/tengah/belakang
{
    ....;
}

//-----
//--- Fungsi untuk Menampilkan Data ---
//-----
public static void tampilkanData(formatBiodata biodataMahasiswa[])
{
    //bagian menampilkan isi struktur Larik -----
    System.out.println("-----");
    System.out.println("NAMA ALAMAT UMUR JEKEL HOBI[0] HOBI[1] HOBI[2] IPK");
    System.out.println("-----");
    for (int i=0; i<=N-1; i++)
    {
        System.out.print (i + ".");
        System.out.print (biodataMahasiswa[i].nama + "\t");
        System.out.print (biodataMahasiswa[i].alamat + "\t");
        System.out.print (biodataMahasiswa[i].umur + "\t");
        System.out.print (biodataMahasiswa[i].jekar + "\t");
        System.out.print (biodataMahasiswa[i].hobi[0] + "\t");
        System.out.print (biodataMahasiswa[i].hobi[1] + "\t");
        System.out.print (biodataMahasiswa[i].hobi[2] + "\t");
        System.out.println(biodataMahasiswa[i].ipk);
    }
    System.out.println("-----");
}

//-----
//--- Program Utama ---
//-----
public static void main(String[] args)
{
    //bagian deklarasi record berbasis LARIK -----
    formatBiodata biodataMahasiswa[] = new formatBiodata[10];
    biodataMahasiswa[0] = new formatBiodata();
    biodataMahasiswa[1] = new formatBiodata();
    biodataMahasiswa[2] = new formatBiodata();
    biodataMahasiswa[3] = new formatBiodata();
    ....;
    biodataMahasiswa[8] = new formatBiodata();
    biodataMahasiswa[9] = new formatBiodata();
    ....;
    ....;

    ngentriData(biodataMahasiswa);
    tampilkanData(biodataMahasiswa);
    ....; //← lakukan pemanggilan fungsi tambah dan hapus (depan/tengah/belakang)

    tampilkanData(biodataMahasiswa);
}
}

```

Program 3.7

3.3. Latihan

Latihan 1 :

Modifikasilah program yang telah anda miliki pada percobaan-percobaan sebelumnya, selanjutnya disebut master program (perhatikan program 3.7), dengan menambahkan potongan program 3.1.

Eksekusi dan ujilah program anda dengan menambahkan sebuah data baru di depan. Bagaimanakah hasilnya? Catatlah kondisi larik sebelum dilakukan penambahan data maupun setelah dilakukan penambahan data. Bandingkanlah hasilnya.

Latihan 2 :

Kembangkanlah lagi master program anda dengan menambahkan potongan program 3.2.

Eksekusi dan ujilah program anda dengan menambahkan sebuah data baru di tengah. Bagaimanakah hasilnya? Catatlah kondisi larik sebelum dilakukan penambahan data maupun setelah dilakukan penambahan data. Bandingkanlah hasilnya.

Latihan 3 :

Kembangkanlah lagi master program anda dengan menambahkan potongan program 3.3.

Eksekusi dan ujilah program dengan menambahkan sebuah data baru di belakang. Bagaimanakah hasilnya? Catatlah kondisi larik sebelum dilakukan penambahan data maupun setelah dilakukan penambahan data. Bandingkanlah hasilnya.

Latihan 4 :

Kembangkanlah lagi master program anda dengan menambahkan potongan program 3.4, 3.5 dan 3.6.

Eksekusi dan ujilah program anda untuk menghapus sebuah data di depan/ tengah dan belakang larik.

Bagaimanakah hasilnya? Catatlah kondisi larik sebelum dilakukan penghapusan data maupun setelah dilakukan penghapusan. Bandingkanlah hasilnya.

Latihan : -

Tugas : -

