

BAB 2

STRUKTUR PENYIMPAN BERBASIS RECORD DAN ARRAY OF RECORD

Tujuan

Mahasiswa dapat membuat suatu struktur *record* (rekaman) dan *array of record* (rekaman dalam larik) untuk menyimpan data menggunakan bahasa java.

Pada percobaan terdahulu (bab 1) kita telah belajar bagaimana membuat media penyimpan (variabel) menggunakan tipe data *primitive* baik itu tipe alphabetic, tipe data numeric maupun tipe data *array/ larik*.

Pada bab 2 ini kita akan lebih banyak belajar bagaimana membuat media penyimpan berbasis *record (rekaman)*. Record sering juga disebut Obyek/ Simpul/ List/ Node/ Senarai. Dalam pembuatannya, record didefinisikan sebagai variabel bertipe *data buatan* (harus dideklarasikan menggunakan class).

Sebelum membahas tipe data buatan ada baiknya kita memahami terlebih dahulu mengapa kita memerlukan struktur penyimpan berbasis *record (rekaman)* melalui topik penyimpan yang tidak terstruktur berikut ini.

2.1. Struktur Penyimpan Yang Tidak Terstruktur

Pada percobaan terdahulu (bab 1) kita telah belajar bagaimana membuat beberapa variabel berupa ***nama, alamat, umur, jekel, hobi []*** dan ***ipk*** seperti tersaji pada program 2.1. berikut ini.

```
import java.util.Scanner;
public class inputDataViaKeyboard
{
    public static void main(String[] args)
    {
        String nama;
        String alamat;
        int umur;
        char jekel; //jenis kelamin
        String hobi[] = new String[3];
        float ipk;

        Scanner masukan = new Scanner(System.in);
        int bacaTombol=0;

        System.out.print("Silakan masukkan nama anda : ");
        nama = masukan.next();

        System.out.print("Silakan masukkan alamat anda : ");
        alamat = masukan.next();
    }
}
```

```

System.out.print("Silakan masukkan umur anda : ");
umur = masukan.nextInt();

System.out.print("Silakan masukkan Jenis Kelamin anda : ");
try
{
    bacaTombol = System.in.read();
}
catch(java.io.IOException e)
{
}
jekel = (char)bacaTombol;
System.out.println("Silakan masukkan hobi (maks 3) : ");
System.out.print("hobi ke-0 : "); hobi[0] = masukan.next();
System.out.print("hobi ke-1 : "); hobi[1] = masukan.next();
System.out.print("hobi ke-2 : "); hobi[2] = masukan.next();
System.out.print("Silakan masukkan IPK anda : ");
ipk = masukan.nextFloat();

System.out.println("Nama anda adalah " + nama);
System.out.println("Nama alamat adalah " + alamat);
System.out.println("Umur anda adalah " + umur);
System.out.println("Jenis Kelamin anda adalah " + jekel);
System.out.println("Hobi ke-0 anda adalah " + hobi[0]);
System.out.println("Hobi ke-1 anda adalah " + hobi[1]);
System.out.println("Hobi ke-2 anda adalah " + hobi[2]);
System.out.println("IPK anda adalah " + ipk);
}
}

```

Hasil Eksekusi

```

Silakan masukkan nama anda : AgungBP ↵
Silakan masukkan alamat anda : Jakarta ↵
Silakan masukkan umur anda : 28 ↵
Silakan masukkan Jenis Kelamin anda : L ↵
Silakan masukkan hobi (maks 3) :
hobi ke-0 : musik ↵
hobi ke-1 : mancing ↵
hobi ke-2 : touring ↵
Silakan masukkan IPK anda : 3.5 ↵

Nama anda adalah AgungBP
Nama alamat adalah Jakarta
Umur anda adalah 28
Jenis Kelamin anda L
Hobi ke-0 anda musik
Hobi ke-1 anda mancing
Hobi ke-2 anda touring
IPK anda adalah 3.5

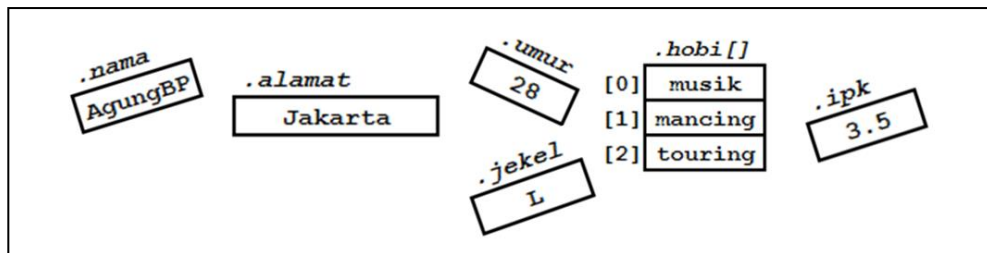
Press any key to continue . . .

```

Program 2.1

Apabila kita perhatikan dengan sungguh-sungguh program 2.1 di atas, meskipun terlihat seperti sebuah kesatuan variabel yang saling berhubungan, namun sebenarnya variabel **nama**, **alamat**, **umur**, **jekel**, **hobi []**, **ipk** bukanlah suatu kesatuan yang utuh. Hal ini disebabkan karena masing-masing variabel tersebut dideklarasikan secara terpisah

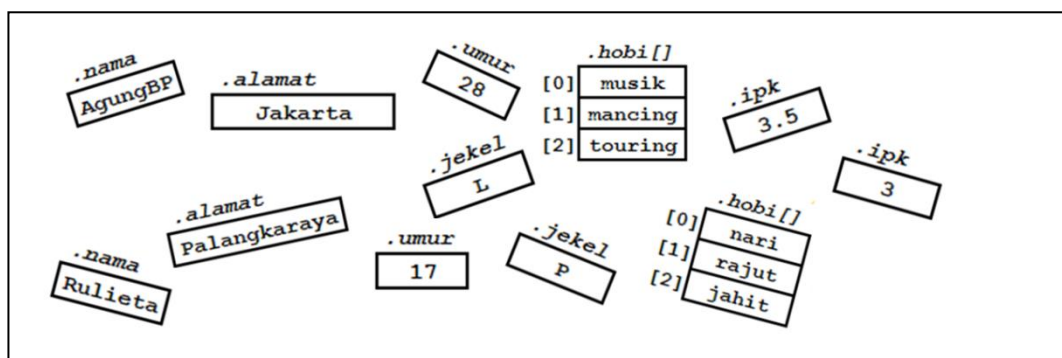
menggunakan tipe data masing-masing sehingga tentu akan membentuk suatu struktur penyimpanan yang terpisah pula sekalipun datanya adalah data milik satu orang ("AgungBP"; "Jakarta", 28, 'L', "musik", "mancing", "touring", 3.5). Akibat dari deklarasi variabel yang terpisah tersebut, maka struktur penyimpanan yang terjadi akan menjadi seperti yang diilustrasikan pada gambar 2.1 di bawah ini.



Gambar 2.1

Pada gambar 2.1 tersebut tidak dapat terlihat bahwa variabel **nama** memiliki hubungan langsung dengan variabel **alamat** dan **umur**. Begitu juga dengan variabel **ipk** dan **jkel**. Jadi, sekalipun variabel-variabel tersebut digunakan untuk menyimpan sebuah kesatuan data milik seorang pribadi ("AgungBP"; "Jakarta", 28, 'L', "musik", "mancing", "touring", 3.5), namun secara struktur variabel-variabel tersebut merupakan variabel-variabel yang terpisah sehingga data-data tersebut belum dapat dikatakan sebuah *kesatuan data*. Nah, dengan demikian model penyimpanan data seperti di atas dapat diartikan sebagai **model penyimpanan data yang tidak terstruktur**.

Apakah anda dapat membayangkan jika model penyimpanan data seperti di atas kita gunakan untuk menyimpan biodata milik lebih dari satu orang? Kira-kira yang terjadi akan seperti gambar 2.2 berikut ini.



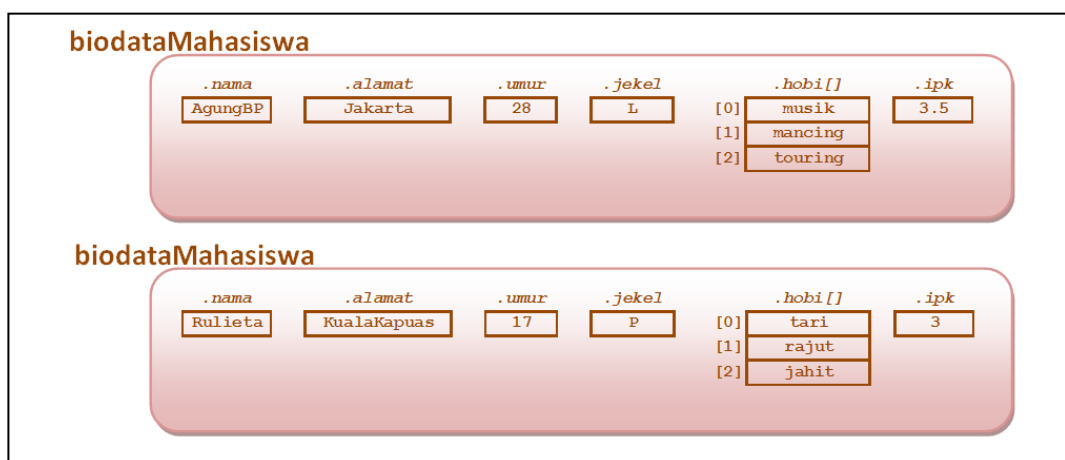
Gambar 2.2

Bila kita melihat gambar 2.2 di atas tentu kita akan kesulitan untuk menentukan keterkaitan data-data di atas. Kita tidak tahu berapa umur AgungBP, apakah 28 atau

17? Siapakah yang memiliki hobi rajut, apakah AgungBP ? atau Rulieta? berapakah IPK dari Rulieta, dan seterusnya. Maka dapat kita simpulkan struktur penyimpanan seperti di atas tidak layak digunakan untuk menyimpan biodata dalam jumlah banyak.

2.2. Struktur Penyimpanan yang Terstruktur (Berbasis Record)

Lalu bagaimana dengan **penyimpanan data yang terstruktur** ? Perhatikan ilustrasi pada gambar 2.3 berikut.



Gambar 2.3

Apabila dibandingkan dengan gambar 2.2, variabel-variabel pada gambar 2.3 tampak jauh lebih rapi dan terstruktur. Hal ini karena baik variabel **nama**, **alamat**, **umur**, **j_kel**, **hobi []** maupun **ipk**, kesemuanya diikat atau dibungkus dalam satu kesatuan oleh satu variabel induk yang bernama **biodataMahasiswa**. Nah, variabel pembungkus inilah yang selanjutnya dikenal dengan istilah **obyek/ heap/ simpul/ node**. Pembahasan lebih jauh tentang obyek/ heap ini dapat anda ikuti pada bab 7.

Untuk dapat membuat struktur sebagaimana gambar 2.3 di atas, anda dapat mengikuti script dalam program 2.2 berikut ini.

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

public class strukturRekamanData
{ public static void main(String[] args)
  { //bagian deklarasi record -----
```

```

formatBiodata    biodataMahasiswa = new formatBiodata();

//bagian entri data melalui keyboard -----
Scanner masukan = new Scanner(System.in);
int bacaTombol=0;

System.out.print("Silakan masukkan nama anda : ");
biodataMahasiswa.nama = masukan.next();

System.out.print("Silakan masukkan alamat anda : ");
biodataMahasiswa.alamat = masukan.next();

System.out.print("Silakan masukkan umur anda : ");
biodataMahasiswa.umur = masukan.nextInt();

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

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

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

System.out.println("Nama anda adalah " + biodataMahasiswa.nama);
System.out.println("Nama alamat adalah "+ biodataMahasiswa.alamat);
System.out.println("Umur anda adalah " + biodataMahasiswa.umur);
System.out.println("Jenis Kelamin anda " + biodataMahasiswa.jekel);
System.out.println("Hobi ke-0 anda " + biodataMahasiswa.hobi[0]);
System.out.println("Hobi ke-1 anda " + biodataMahasiswa.hobi[1]);
System.out.println("Hobi ke-2 anda " + biodataMahasiswa.hobi[2]);
System.out.println("IPK anda adalah " + biodataMahasiswa.ipk);
}
}

```

Hasil Eksekusi

```

Silakan masukkan nama anda : AgungBP ↵
Silakan masukkan alamat anda : Jakarta ↵
Silakan masukkan umur anda : 28 ↵
Silakan masukkan Jenis Kelamin anda : L ↵
Silakan masukkan hobi (maks 3) :
hobi ke-0 : musik ↵
hobi ke-1 : mancing ↵
hobi ke-2 : touring ↵
Silakan masukkan IPK anda : 3.5 ↵

Nama anda adalah Agung
Nama alamat adalah Jakarta
Umur anda adalah 28
Jenis Kelamin anda L
Hobi ke-0 anda musik
Hobi ke-1 anda mancing

```

```
Hobi ke-2 anda touring
IPK anda adalah 3.5

Press any key to continue . . .
```

Program 2.2

Dari program 2.2 di atas, walaupun hasil eksekusinya sama dengan program 2.1, namun secara struktur kedua program tersebut sangat jauh berbeda. Pada program 2.2 anda dapat melihat bahwa ada sebuah variabel bernama **biodataMahasiswa** yang berfungsi untuk menyatukan variabel yang lebih kecil yang berupa **nama, alamat, umur, jekel, hobi []** dan **ipk**.

Jika kita lihat pada bagian deklarasi, variabel **biodataMahasiswa** ini tidaklah dibentuk menggunakan tipe data primitif seperti string, char, int, dan lainnya, tetapi dideklarasikan dalam tipe data bernama **formatBiodata**. Tipe data **formatBiodata** ini tidaklah dikenal oleh Java karena memang bukanlah tipe bawaan dari java. Karena tipe data ini bukanlah bawaan java melainkan merupakan ciptaan programmer maka tipe data ini yang disebut dengan **tipe data buatan**.

Dalam pembuatannya, tipe data buatan dibentuk dengan melibatkan penggunaan **kelas (class)**, sedangkan variabelnya sendiri yaitu **biodataMahasiswa** disebut sebagai **obyek**. (anda akan mempelajari class dan obyek lebih jauh dalam Pemrograman Berorientasi Obyek). Variabel atau obyek yang bernama **biodataMahasiswa** ini akan menjadi pembungkus atau wadah bagi variabel-variabel yang lebih kecil yaitu **nama, alamat, umur, jekel, hobi []** dan **ipk**.

Dari tata cara penyebutannya, memang akhirnya penyebutan variabel **nama, alamat, umur, jekel**, dan juga **ipk** tidaklah menjadi sederhana karena penyebutannya harus senantiasa didahului dengan menyebut nama obyeknya terlebih dahulu baru kemudian disusul dengan tanda titik (.) dan dilanjutkan dengan penyebutan nama variabelnya. Format baku penyebutannya mengikuti aturan sebagai berikut:

[nama obyek] • [nama variabel]

Sebagai contoh :

```
biodataMahasiswa.nama
biodataMahasiswa.alamat
biodataMahasiswa.umur
biodataMahasiswa.jekel
biodataMahasiswa.hobi[0]
biodataMahasiswa.hobi[1]
biodataMahasiswa.hobi[2]
biodataMahasiswa.ipk
```

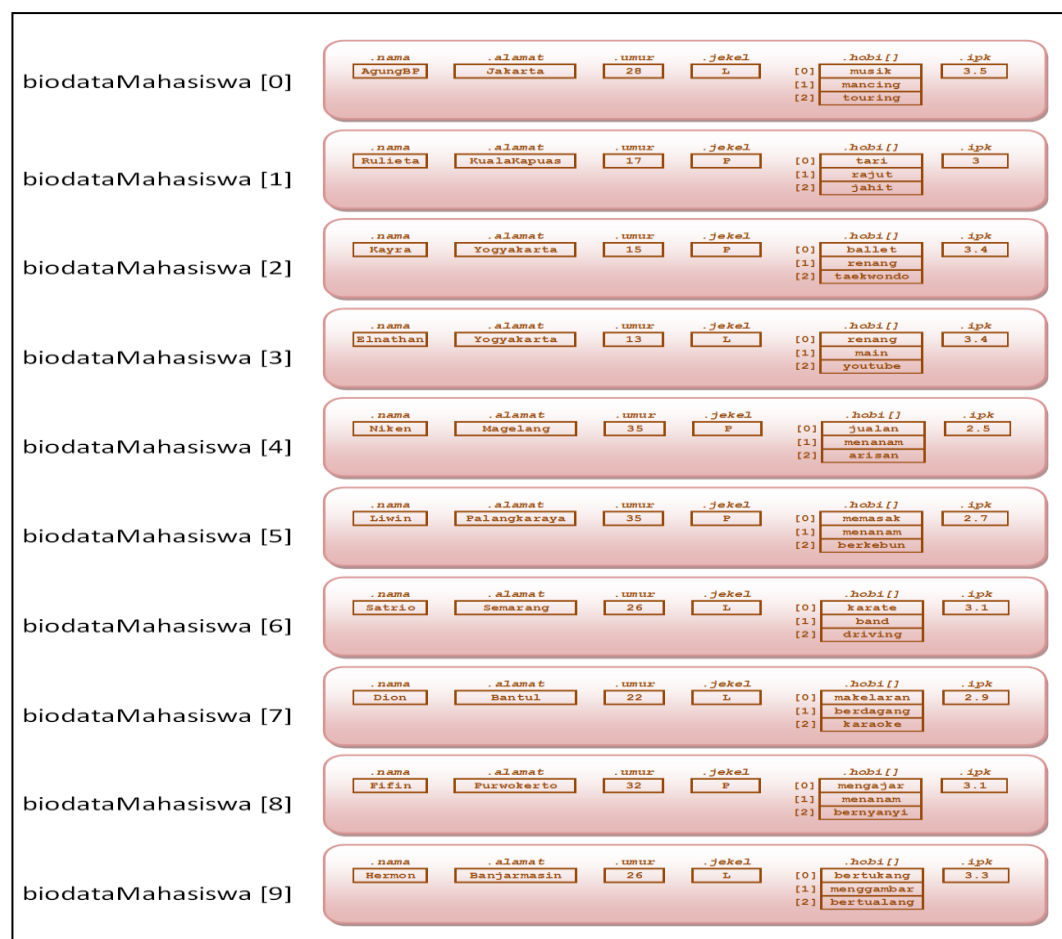
Sekalipun terlihat lebih rumit, dengan menggunakan struktur penyimpan yang telah dipaparkan di atas kita telah dapat memastikan bahwa data-data "AgungBP", "Jakarta", 28, 'L', "musik", "mancing", "touring"

3.5 yang tersimpan dalam variabel *nama*, *alamat*, *umur*, *jekel*, *hobi[0]*, *hobi[1]*, *hobi[2]*, *ipk* adalah biodata milik seorang pribadi yang sama. Dengan demikian model penyimpanan data tersebut sudah dapat diartikan merupakan **model penyimpanan data yang terstruktur**.

Karena fungsinya yang dapat membungkus atau menyatukan beberapa variabel yang terpisah maka konsep penyimpanan data seperti ini juga dapat disebut dengan konsep penyimpanan yang **berbasis pada record (rekaman)**.

2.3. Struktur Penyimpanan Berbasis Array of Record (Rekaman dalam Larik)

Nah, berikut ini kita akan membuat suatu susunan array (larik) yang terdiri dari record-record (rekaman) *biodataMahasiswa* sedemikian sehingga terbentuk sebuah struktur data sebagaimana tergambar dalam ilustrasi gambar 2.4 berikut ini.



Gambar 2.4

Untuk dapat membuat struktur array seperti di atas, anda dapat mengikuti script yang tersaji pada program 2.3 berikut ini.

```

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

class strukturRekamanData
{ public static void main(String[] args)
{
    int N=5;
    //bagian deklarasi record berbasis LARIK -----
    formatBiodata biodataMahasiswa[] = new formatBiodata[5];
    biodataMahasiswa[0] = new formatBiodata();
    biodataMahasiswa[1] = new formatBiodata();
    biodataMahasiswa[2] = new formatBiodata();
    biodataMahasiswa[3] = new formatBiodata();
    biodataMahasiswa[4] = new formatBiodata();

    //bagian entri data ke dalam struktur larik -----
    Scanner masukan = new Scanner(System.in);
    int bacaTombol=0;

    for (int i=0; i<=N-1; i++)
    { System.out.print("Silakan masukkan nama anda : ");
      biodataMahasiswa[i].nama = masukan.next();

      System.out.print("Silakan masukkan alamat anda : ");
      biodataMahasiswa[i].alamat = masukan.next();

      System.out.print("Silakan masukkan umur anda : ");
      biodataMahasiswa[i].umur = masukan.nextInt();

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

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

      System.out.print("Silakan masukkan IPK anda : ");
      biodataMahasiswa[i].ipk = masukan.nextFloat();
      System.out.println("");
    }
    //bagian menampilkan isi struktur Larik -----
    System.out.println("-----");
    System.out.println("NAMA ALAMAT UMUR JEKEL HOBI1 HOBI2 HOBI3 IPK");
    System.out.println("-----");
}
}

```



```

for (int i=0; i<=N-1; i++)
{
    System.out.print (biodataMahasiswa[i].nama + " ");
    System.out.print (biodataMahasiswa[i].alamat + " ");
    System.out.print (biodataMahasiswa[i].umur + " ");
    System.out.print (biodataMahasiswa[i].j_kel + " ");
    System.out.print (biodataMahasiswa[i].hobi[0] + " ");
    System.out.print (biodataMahasiswa[i].hobi[1] + " ");
    System.out.print (biodataMahasiswa[i].hobi[2] + " ");
    System.out.println(biodataMahasiswa[i].ipk);
}
System.out.println("-----");
}
}

```

Hasil Eksekusi :

	NAMA	ALAMAT	UMUR	JEKEL	HOBI[0]	HOBI[1]	HOBI[2]	IPK
0.	AgungBP	Jakarta	28	L	musik	mancing	touring	3.5
1.	Rulieta	KualaKapas	17	P	nari	rajut	jahit	3.0
2.	Karti	Palangkaraya	18	P	renang	fbkan	masak	2.5
3.	Hermon	Banjarmasin	25	L	kasti	bola	mancing	3.4
4.	Anis	Surakarta	30	P	jualan	drama	nonton	0.3
5.	Yanti	Purwokerto	21	P	lukis	musik	selfi	2.0
6.	Handono	Bekasi	29	L	touring	musik	hiking	3.1
7.	Anton	Jakarta	17	L	dangdut	jajan	turing	2.0
8.	Mantri	Magelang	45	L	hiphop	wayang	game	3.1
9.	Satrio	Yogyakarta	30	L	musik	drama	karate	3.3

Press any key to continue . . .

Program 2.3

Sedikit berbeda dengan program 2.2 sebelumnya, karena menggunakan larik (array) maka pada program 2.3 penyebutan nama obyek yang mendahului penyebutan variabel nama, alamat, umur, jenisKelaminnya, dan juga IPK, harus diikuti juga dengan tanda kurung array [].

[nama obyek][indeks] • [nama variabel]

Sebagai contoh :

```

biodataMahasiswa[0].nama
biodataMahasiswa[0].alamat
biodataMahasiswa[0].umur
biodataMahasiswa[0].j_kel
biodataMahasiswa[0].hobi[0]
biodataMahasiswa[0].hobi[1]
biodataMahasiswa[0].hobi[2]
biodataMahasiswa[0].ipk

```

Pada program 2.3 di atas dibangun sebuah struktur penyimpan array yang berorientasi pada record (array of record). Program tersebut mampu menampung data mahasiswa sebanyak N = 5 orang.

2.4. Pembuatan Program Secara Modular (Fungsi)

Pada bagian ini kita akan memperbaiki program 2.3 menjadi lebih rapi. Apabila kita memperhatikan program tersebut dengan seksama tampak semua script ditulis di dalam program utama (main). Hal tersebut bukanlah hal yang baik. Dengan program 2.4 kita akan mencoba mengelompok-kelompokkan beberapa bagian program sesuai dengan kegunaannya.

Apabila kita mencermati program-program yang dibuat sebelum ini terdapat 3 bagian besar. Yang pertama bagian **deklarasi**, kedua bagian **entri data**, dan ketiga bagian **menampilkan data**. Oleh karena itu pada program 2.4 berikut ini script program akan ditulis secara modular berdasarkan ketiga bagian di atas.

```
import java.util.Scanner;
class formatBiodata
{ //bagian deklarasi struktur record -----
    String nama;
    String alamat;
    int umur;
    char jekel;
    String hobi[] = new String[3];
    float ipk;
}
class pertemuan3
{ public static int N=5;

    //-----
    //--- Fungsi untuk mengentri data ke dalam Larik ---
    //-----
    public static void ngentriData(formatBiodata biodataMahasiswa[])
    {
        //bagian entri data ke dalam struktur larik -----
        Scanner masukan = new Scanner(System.in);
        int bacaTombol=0;

        for (int i=0; i<=N-1; i++)
        { System.out.print("Silakan masukkan nama anda : ");
          biodataMahasiswa[i].nama = masukan.next();

          System.out.print("Silakan masukkan alamat anda : ");
          biodataMahasiswa[i].alamat = masukan.next();

          System.out.print("Silakan masukkan umur anda : ");
          biodataMahasiswa[i].umur = masukan.nextInt();

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

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

```

        System.out.print("hobi ke-1 : ");
        biodataMahasiswa[i].hobi[1] = masukan.next();
        System.out.print("hobi ke-2 : ");
        biodataMahasiswa[i].hobi[2] = masukan.next();

        System.out.print("Silakan masukkan IPK anda : ");
        biodataMahasiswa[i].ipk = masukan.nextFloat();
        System.out.println("");
    }
}

//-----
//---          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 HOBI1 HOBI2 HOBI3 IPK");
    System.out.println("-----");
    for (int i=0; i<=N-1; i++)
    {
        System.out.print (biodataMahasiswa[i].nama + " ");
        System.out.print (biodataMahasiswa[i].alamat + " ");
        System.out.print (biodataMahasiswa[i].umur + " ");
        System.out.print (biodataMahasiswa[i].jekel + " ");
        System.out.print (biodataMahasiswa[i].hobi[0] + " ");
        System.out.print (biodataMahasiswa[i].hobi[1] + " ");
        System.out.print (biodataMahasiswa[i].hobi[2] + " ");
        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[4] = new formatBiodata();
    ngentriData(biodataMahasiswa);
    tampilkanData(biodataMahasiswa);
}
}

```

Hasil Eksekusi :

	NAMA	ALAMAT	UMUR	JEKEL	HOBI[0]	HOBI[1]	HOBI[2]	IPK
0.	AgungBP	Jakarta	28	L	musik	mancing	touring	3.5
1.	Rulieta	KualaKapas	17	P	nari	rajut	jahit	3.0
2.	Karti	Palangkaraya	18	P	renang	fbkan	masak	2.5
3.	Hermon	Banjarmasin	25	L	kasti	bola	mancing	3.4
4.	Anis	Surakarta	30	P	jualan	drama	nonton	0.3
5.	Yanti	Purwokerto	21	P	lukis	musik	selfi	2.0
6.	Handono	Bekasi	29	L	touring	musik	hiking	3.1
7.	Anton	Jakarta	17	L	dangdut	jajan	turing	2.0

8. Mantri	Magelang	45	L	hiphop	wayang	game	3.1
9. Satrio	Yogyakarta	30	L	musik	drama	karate	3.3

Press any key to continue . . .							

Program 2.4

2.5. Latihan

Latihan 1 :

Tuliskan dan eksekusilah program 2.1 hingga 2.4. Catat hasilnya.

Latihan 2 :

Modifikasilah program 2.4. agar dapat digunakan untuk mencatat seluruh data pada gambar 2.3 (Nilai N ditentukan secara statis sebanyak 10 orang, dengan data yang telah ditentukan).

Latihan 3 :

Modifikasilah program 2.4 agar dapat digunakan untuk memasukkan data dengan banyak record (N) **dinamis**. Banyak record (N) dientri oleh user **melalui keyboard** dengan maksimum 15 record.

Latihan 4 :

Modifikasilah program 2.4 agar dapat digunakan untuk memasukkan data dengan banyak record (N) dinamis. Banyak record (N) akan bertambah secara otomatis (++) apabila user menghendaki memasukkan data lagi. Maksimum record 20.