

## MODUL 14

### HASHING PADA LARIK



#### CAPAIAN PEMBELAJARAN

Mahasiswa dapat melakukan penempatan suatu data ke dalam larik menggunakan teknik hashing



#### KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad



#### DASAR TEORI

##### Hashing

*Hashing* adalah teknik penempatan sebuah record pada larik dengan nomor indeks yang tidak standar menjadi standar [0], [1], [2], [3], .. dst. Hashing menempatkan sebuah record pada larik pada nomor indeks khusus yang merupakan hasil konversi dengan menggunakan **rumus khusus**.

Sebagai contoh, jika ada 8 orang mahasiswa dengan NIM masing-masing :

Tabel 14.1

NIM	NAMA	ALAMAT	UMUR	JEKEL	IPK
125410067	AgungBP	Jakarta	28	L	3.5
155410125	Rulieta	KualaKapuas	17	P	3.0
115410125	Kayra	Yogyakarta	15	P	3.4
115410066	Elnathan	Yogyakarta	13	L	2.4
165410136	Niken	Magelang	35	P	2.5
145410112	Liwin	Palangkaraya	35	P	2.7
155410133	Satrio	Semarang	26	L	3.1
155410143	Dion	Bantul	22	L	2.0

maka record-record tersebut akan ditempatkan pada larik dengan indeks sesuai dengan NIM yang dimilikinya yaitu :

Tabel 14.2

NIM	NAMA	ALAMAT	UMUR	JEKEL	IPK	ARRAY (NIM)
125410067	AgungBP	Jakarta	28	L	3.5	[ 125410067 ]
155410125	Rulieta	KualaKapuas	17	P	3.0	[ 155410125 ]
115410125	Kayra	Yogyakarta	15	P	3.4	[ 115410125 ]
115410066	Elnathan	Yogyakarta	13	L	2.4	[ 115410066 ]
165410136	Niken	Magelang	35	P	2.5	[ 165410136 ]
145410112	Liwin	Palangkaraya	35	P	2.7	[ 145410112 ]

155410133	Satrio	Semarang	26	L	3.1	[ 155410133 ]
155410143	Dion	Bantul	22	L	2.0	[ 155410143 ]

Faktanya untuk mendeklarasikan larik dengan 9 digit seperti di atas tentu memerlukan memori yang sangat besar (9 digit = 1 milyar). Itupun larik dengan kapasitas tersebut tidak akan digunakan seluruhnya karena sesungguhnya jumlah mahasiswa secara keseluruhan tidak banyak misalkan hanya berjumlah 1000 orang. Untuk itu diperlukan sebuah rumus yang mampu mengonversi NIM mahasiswa menjadi nomor larik antara 000 sampai dengan 999. Fungsi tersebut dinamakan ***fungsi hash***.

Ada beberapa metode dalam melakukan hashing yaitu

- (1) **metode pembagian**,
- (2) **metode midsquare**, dan
- (3) **metode penjumlahan digit**.

Pada praktikum ini teknik hashing yang dibahas adalah dengan menggunakan **metode pembagian**.

#### Hashing dengan metode pembagian

Pada metode pembagian, nilai hash diperoleh dengan cara kunci yang akan di cari nilai hashnya (misalkan NIM) di 'modulus' dengan sebuah bilangan prima yang paling dekat dengan kapasitas array (N). Misalnya kapasitas array yang akan kita bangun sebesar 1000, maka bilangan prima terdekat 1000 yang akan digunakan sebagai pembagi NIM adalah 997. Dengan penghitungan indeks menggunakan modulus 997 maka penempatan data akan menjadi sebagai berikut :

Tabel 14.3

NIM	NAMA	ALAMAT	UMUR	JEKEL	IPK	ARRAY (modulus)
125410067	AgungBP	Jakarta	28	L	3.5	[ 428 ]
155410125	Rulieta	KualaKapuas	17	P	3.0	[ 756 ]
115410125	Kayra	Yogyakarta	15	P	3.4	[ 396 ]
115410066	Elnathan	Yogyakarta	13	L	2.4	[ 337 ]
165410136	Niken	Magelang	35	P	2.5	[ 857 ]
145410112	Liwin	Palangkaraya	35	P	2.7	[ 653 ]
155410133	Satrio	Semarang	26	L	3.1	[ 764 ]
155410143	Dion	Bantul	22	L	2.0	[ 774 ]

Program untuk hasing dengan metode pembagian dapat dilihat pada program 14.1 yaitu pada bagian fungsi **hitungNilaiHash()**.

Perhatikan program 14.1 berikut.

```
import java.util.Scanner;
class formatBiodata
{
    //bagian deklarasi struktur record -----
    int    nim;
    String nama;
    String alamat;
    int    umur;
```

```

        char    jekel;
        float   ipk;
    }

    class hashing
    {
        public static int N=0;

        public static int hitungNilaiHash(int nilaiAwal)
        {
            int hasil;
            hasil = nilaiAwal % 997;
            return (hasil);
        }

        public static void ngentriData(formatBiodata biodataMahasiswa[])
        {
            N = 1000;
            int NH;

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

            //bagian menentukan banyaknya data yang akan dientri -----
            System.out.print("Berapa data yang akan dientri ? : ");
            int banyakEntri = masukan.nextInt();

            //bagian entri data baru -----
            formatBiodata biodataMahasiswaBaru;
            for (int i=0; i<=banyakEntri-1; i++)
            {
                //bagian entri data baru ke penyimpanan sementara -----
                biodataMahasiswaBaru = new formatBiodata();
                System.out.print("Silakan masukkan NIM anda : ");
                biodataMahasiswaBaru.nim = masukan.nextInt();
                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.print("Silakan masukkan IPK anda : ");
                biodataMahasiswaBaru.ipk = masukan.nextFloat();

                //bagian memindahkan data baru ke larik sesuai nilai Hashing -----
                NH = hitungNilaiHash(biodataMahasiswaBaru.nim);
                System.out.println ("Biodata " + biodataMahasiswaBaru.nama + " akan
ditempatkan pada larik ke : " + NH);
                biodataMahasiswa[NH] = biodataMahasiswaBaru;
            }
        }

        public static void berhentiSebentar()
        {
            System.out.println ("");
            System.out.println ("Tekan tombol ENTER untuk melanjutkan...");
            Scanner masukan = new Scanner(System.in);
            int bacaTombol;
            do
            {
                bacaTombol=0;
                try
                {
                    bacaTombol = System.in.read();
                }
                catch(java.io.IOException e)
                {
                }
            }
        }
    }

```

```

    }
    while (bacaTombol != 13); //tombol 13 adalah tombol enter
}

public static void tampilkanData(formatBiodata biodataMahasiswa[])
{
    //bagian menampilkan isi struktur Larik -----
    System.out.println("-----");
    System.out.println("NO NAMA          ALAMAT          UMUR    JEKEL    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].jekel + "\t ");
        System.out.println(biodataMahasiswa[i].ipk);
        if (i % 100 == 0)
            berhentiSebentar();
    }
    System.out.println("-----");
}

public static void main(String[] args)
{
    //bagian deklarasi record berbasis LARIK -----
    formatBiodata biodataMahasiswa[] = new formatBiodata[1000];
    for (int i=0; i<=999; i++)
        biodataMahasiswa[i] = new formatBiodata();

    //pemanggilan fungsi-----
    ngentriData(biodataMahasiswa);
    tampilkanData(biodataMahasiswa);
}
}

```

Program 14.1



## PRAKTIK

### 1. Praktek 1

- Tuliskan program 14.1 kemudian eksekusilah.
- Saat eksekusi, entrilah kedelapan data pada tabel 14.1.
- Lakukan pengamatan sebagai berikut: Diletakkan di manakah kedelapan data tersebut di dalam larik? *(Untuk menghemat saat mencetak, screenshotlah hanya bagian-bagian yang menampilkan kedelapan data tersebut, sementara larik-larik yang menunjukkan data kosong/null ditampilkan seperlunya saja).*

### 2. Praktek 2

- Eksekusilah kembali program 14.1.

- b. Entrilah kedelapan data pada tabel 14.1 **ditambah dengan 2 buah berikut ini.**

Tabel 14.3

NIM	NAMA	ALAMAT	UMUR	JEKEL	IPK
155411140	Fifin	Purwokerto	32	P	3.1
155412137	Hermon	Banjarmasin	26	L	3.3

- c. Screenshootlah bagian-bagian yang menampilkan kedelapan data tersebut, dan kedua data yang ditambahkan. Amati, di larik ke berapakah data 'Fifin' dan 'Hermon' ditempatkan? Bagaimana nasib data 'Dion'? Menurut anda mengapa hal ini terjadi? \*\* Jelaskan dalam laporan resmi.

(\*\*terjadi *collision* atau *tabrakan data*)

### 3. Praktek 3

- a. Untuk mengatasi masalah yang terjadi pada pelaksanaan praktikum ke-2, modifikasilah fungsi `ngentriData()` dengan menambahkan instruksi (tercetak tebal) pada program sehingga menjadi seperti berikut.

```
public static void ngentriData(formatBiodata biodataMahasiswa[])
{
    N = 1000;
    int NH;

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

    //bagian menentukan banyaknya data yang akan dientri -----
    System.out.print("Berapa data yang akan dientri ? : ");
    int banyakEntri = masukan.nextInt();

    //bagian membuat record sementara untuk menampung data baru-----
    formatBiodata biodataMahasiswaBaru;

    //bagian entri data baru -----
    for (int i=0; i<=banyakEntri-1; i++)
    {
        //bagian entri data baru ke penyimpanan sementara -----
        biodataMahasiswaBaru = new formatBiodata();
        System.out.print("Silakan masukkan NIM anda : ");
        biodataMahasiswaBaru.nim = masukan.nextInt();
        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.print("Silakan masukkan IPK anda : ");
        biodataMahasiswaBaru.ipk = masukan.nextFloat();
    }
}
```

```

//bagian memindahkan data baru ke larik sesuai nilai Hashing---
NH = hitungNilaiHash(biodataMahasiswaBaru.nim);

//++++++ MENGATASI COLLISION ++++++
while (biodataMahasiswa[NH].nama != null)
{   System.out.println("terjadi tabrakan pada NH=" + NH);
    NH++;
}
//+++++

System.out.println ("Biodata " + biodataMahasiswaBaru.nama + "
akan ditempatkan pada larik ke : " + NH);

    biodataMahasiswa[NH] = biodataMahasiswaBaru;
}
}

```

Program 14.2

- b. Ulangi pelaksanaan praktikum 2 nomor a dan b.
- c. Sekarang amati, di larik ke berapakah data Anton, Mantri dan Satrio ditempatkan? Apakah masih terjadi collision? *(jelaskan dalam laporan resmi)*



## LATIHAN

---

1. Buatlah fungsi pencarian data di mana kunci pencarian yang dientri adalah NIM. Fungsi pencarian yang dimaksud bukanlah fungsi pencarian sekuensial yang mencari mulai dari data ke 0 hingga ke 1000, melainkan pencarian dengan **menggunakan fungsi hash**.
2. ....



## TUGAS

---

1. ....
2. ....



## REFERENSI

---

Disadur dari Buku Ajar Struktur Data Menggunakan Java, Agung Budi Prasetyo, 2017, hal: 124-129, <http://agungbudiprasetyo.atspace.com/buku/index.html>, diakses pada 12:07 PM 8/06/2019.