MODUL 5
PENGELOLAAN DATA PADA ARRAY/ LARIK: PENGURUTAN (SORTING)

CAPAIANPEMBELAJARAN

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

KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad

DASAR TEORI

Sorting (Pengurutan)

Sorting (pengurutan) merupakan konsep/ algoritma yang sering digunakan. Jika berhubungan dengan jumlah data yang besar, data akan mudah dikelola jika sudah terurut. Algoritma pengurutan melibatkan beberapa konsep-konsep dasar pemograman, yaitu:

- Urutan/Sequence. Kaidah pemrograman yang menyatakan bahwa perintah-perintah dalam program komputer akan dieksekusi menurut urutannya dari atas ke bawah
- Seleksi/Selection. Perintah-perintah dalam program komputer akan dieksekusi berdasarkan nilai kebenaran boolean tertentu.
- Perulangan/Loop. Sejumlah perintah dalam program komputer yang akan dieksekusi beberapa kali berdasarkan nilai kebenaran boolean-nya.

Sorting dapat dibuat dengan dua model, yatu Ascending dan Descending. Ascending yaitu melakukan pengurutan dari data yang terkecil ke data yang lebih besar. Contoh : 1, 2, 3, . . . ,10.
Descending yaitu melakukan pengurutan dari data yang terbesar ke data yang lebih kecil. Contoh: 10, 9, 8, . . . , 1.

Ada beberapa metode sorting, misalnya :

- Bubble Sort
- Selection Sort
- Insertion Sort
- Quick Sort
- MergeSort
- dan lainnya

Pada praktikum ini kita akan mempelajari teknik sorting **bubble sort, selection Sort** dan **Insertion Sort**.

a. Algoritma Bubble Sort

Algoritma bubble sort akan membandingkan sebuah elemen dengan elemen di sebelah kanannya. Jika elemen yang dibaca lebih besar dari elemen di sebelah kanannya maka kedua elemen tersebut harus ditukar (untuk kasus Ascending). Proses ”banding-tukar” ini akan dilakukan mulai dari elemen pertama (data paling kiri) hingga elemen terakhir (data paling kanan) sebagai satu buah siklus. Pada akhir siklus yang pertama akan diperoleh hasil berupa kondisi di mana data yang paling besar berada di posisi paling kanan.

Pada siklus berikutnya, proses ”banding-tukar” akan dilakukan lakukan lagi untuk mendapatkan data terbesar kedua untuk diletakkan di posisi kedua dari kanan. Kemudian siklus akan diulang kembali untuk ketiga kalinya, keempat kalinya, kelima kalinya demikian seterusnya hingga diperoleh data terbesar ketiga, keempat, kelima, dan seterusnya. Keberlangsungan siklus baru akan berhenti apabila seluruh data terurutkan.

Agar mudah dipahami akan disimulasikan contoh pengurutan data berikut ini.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
DATA AWAL	Rulieta	Hermon	Satrio	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Dengan metode bubblesort, dapat kita lihat proses pengurutan sebagaimana tersaji pada gambar berikut ini.

Siklus 1 (pada saat nilai *j=0*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**Rulieta**	**Hermon**	Satrio	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion
	[i] > [i+1], tukar									
Langkah 2 : Pada saat *i=1*	Hermon	**Rulieta**	**Satrio**	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion
		[i] < [i+1], tidak tukar								
Langkah 3 : Pada saat *i=2*	Hermon	Rulieta	**Satrio**	**AgungBP**	Niken	Fifin	Liwin	Kayra	Elnathan	Dion
			[i] > [i+1], tukar							
Langkah 4 : Pada saat *i=3*	Hermon	Rulieta	AgungBP	**Satrio**	**Niken**	Fifin	Liwin	Kayra	Elnathan	Dion
				[i] > [i+1], tukar						
Langkah 5 : Pada saat *i=4*	Hermon	Rulieta	AgungBP	Niken	**Satrio**	**Fifin**	Liwin	Kayra	Elnathan	Dion
					[i] > [i+1], tukar					
Langkah 6 : Pada saat *i=5*	Hermon	Rulieta	AgungBP	Niken	Fifin	**Satrio**	**Liwin**	Kayra	Elnathan	Dion
						[i] > [i+1], tukar				
Langkah 7 : Pada saat *i=6*	Hermon	Rulieta	AgungBP	Niken	Fifin	Liwin	**Satrio**	**Kayra**	Elnathan	Dion
							[i] > [i+1], tukar			
Langkah 8 : Pada saat *i=7*	Hermon	Rulieta	AgungBP	Niken	Fifin	Liwin	Kayra	**Satrio**	**Elnathan**	Dion
								[i] > [i+1], tukar		
Langkah 9 : Pada saat *i=8*	Hermon	Rulieta	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	**Satrio**	**Dion**
									[i] > [i+1], tukar	
Hasil :	Hermon	Rulieta	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion	**Satrio**

Siklus 2 (pada saat nilai *j=1*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**Hermon**	**Rulieta**	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion	*Satrio*
	[i] < [i+1], tidak tukar									
Langkah 2 : Pada saat *i=1*	Hermon	**Rulieta**	**AgungBP**	Niken	Fifin	Liwin	Kayra	Elnathan	Dion	*Satrio*
		[i] > [i+1], tukar								
Langkah 3 : Pada saat *i=2*	Hermon	AgungBP	**Rulieta**	**Niken**	Fifin	Liwin	Kayra	Elnathan	Dion	*Satrio*
			[i] > [i+1], tukar							
Langkah 4 : Pada saat *i=3*	Hermon	AgungBP	Niken	**Rulieta**	**Fifin**	Liwin	Kayra	Elnathan	Dion	*Satrio*
				[i] > [i+1], tukar						
Langkah 5 : Pada saat *i=4*	Hermon	AgungBP	Niken	Fifin	**Rulieta**	**Liwin**	Kayra	Elnathan	Dion	*Satrio*
					[i] > [i+1], tukar					
Langkah 6 : Pada saat *i=5*	Hermon	AgungBP	Niken	Fifin	Liwin	**Rulieta**	**Kayra**	Elnathan	Dion	*Satrio*
						[i] > [i+1], tukar				
Langkah 7 : Pada saat *i=6*	Hermon	AgungBP	Niken	Fifin	Liwin	Kayra	**Rulieta**	**Elnathan**	Dion	*Satrio*
							[i] > [i+1], tukar			
Langkah 8 : Pada saat *i=7*	Hermon	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	**Rulieta**	**Dion**	*Satrio*
								[i] > [i+1], tukar		
Hasil :	Hermon	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion	**Rulieta**	*Satrio*

Siklus 3 (pada saat nilai *j=2*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**Hermon**	**AgungBP**	Niken	Fifin	Liwin	Kayra	Elnathan	Dion	*Rulieta*	*Satrio*
		*[i] > [i+1], **tukar***								
Langkah 2 : Pada saat *i=1*	AgungBP	**Hermon**	**Niken**	Fifin	Liwin	Kayra	Elnathan	Dion	*Rulieta*	*Satrio*
			*[i] < [i+1], tidak **tukar***							
Langkah 3 : Pada saat *i=2*	AgungBP	Hermon	**Niken**	**Fifin**	Liwin	Kayra	Elnathan	Dion	*Rulieta*	*Satrio*
				*[i] > [i+1], **tukar***						
Langkah 4 : Pada saat *i=3*	AgungBP	Hermon	Fifin	**Niken**	**Liwin**	Kayra	Elnathan	Dion	*Rulieta*	*Satrio*
					*[i] > [i+1], **tukar***					
Langkah 5 : Pada saat *i=4*	AgungBP	Hermon	Fifin	Liwin	**Niken**	**Kayra**	Elnathan	Dion	*Rulieta*	*Satrio*
						*[i] > [i+1], **tukar***				
Langkah 6 : Pada saat *i=5*	AgungBP	Hermon	Fifin	Liwin	Kayra	**Niken**	**Elnathan**	Dion	*Rulieta*	*Satrio*
							*[i] > [i+1], **tukar***			
Langkah 7 : Pada saat *i=6*	AgungBP	Hermon	Fifin	Liwin	Kayra	Elnathan	**Niken**	**Dion**	*Rulieta*	*Satrio*
								*[i] > [i+1], **tukar***		
Hasil :	AgungBP	Hermon	Fifin	Liwin	Kayra	Elnathan	Dion	**Niken**	*Rulieta*	*Satrio*

Siklus 4 (pada saat nilai *j=3*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**AgungBP**	**Hermon**	Fifin	Liwin	Kayra	Elnathan	Dion	*Niken*	*Rulieta*	*Satrio*
		*[i] < [i+1], tidak **tukar***								
Langkah 2 : Pada saat *i=1*	AgungBP	**Hermon**	**Fifin**	Liwin	Kayra	Elnathan	Dion	*Niken*	*Rulieta*	*Satrio*
			*[i] > [i+1], **tukar***							
Langkah 3 : Pada saat *i=2*	AgungBP	Fifin	**Hermon**	**Liwin**	Kayra	Elnathan	Dion	*Niken*	*Rulieta*	*Satrio*
				*[i] < [i+1], tidak **tukar***						
Langkah 4 : Pada saat *i=3*	AgungBP	Fifin	Hermon	**Liwin**	**Kayra**	Elnathan	Dion	*Niken*	*Rulieta*	*Satrio*
					*[i] > [i+1], **tukar***					
Langkah 5 : Pada saat *i=4*	AgungBP	Fifin	Hermon	Kayra	**Liwin**	**Elnathan**	Dion	*Niken*	*Rulieta*	*Satrio*
						*[i] > [i+1], **tukar***				
Langkah 6 : Pada saat *i=5*	AgungBP	Fifin	Hermon	Kayra	Elnathan	**Liwin**	**Dion**	*Niken*	*Rulieta*	*Satrio*
							*[i] > [i+1], **tukar***			
Hasil :	AgungBP	Fifin	Hermon	Kayra	Elnathan	Dion	**Liwin**	*Niken*	*Rulieta*	*Satrio*

Siklus 5 (pada saat nilai *j=3*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**AgungBP**	**Fifin**	Hermon	Kayra	Elnathan	Dion	*Liwin*	*Niken*	*Rulieta*	*Satrio*
		*[i] < [i+1], tidak **tukar***								
Langkah 2 : Pada saat *i=1*	AgungBP	**Fifin**	**Hermon**	Kayra	Elnathan	Dion	*Liwin*	*Niken*	*Rulieta*	*Satrio*
			*[i] < [i+1], tidak **tukar***							
Langkah 3 : Pada saat *i=2*	AgungBP	Fifin	**Hermon**	**Kayra**	Elnathan	Dion	*Liwin*	*Niken*	*Rulieta*	*Satrio*
				*[i] < [i+1], tidak **tukar***						
Langkah 4 : Pada saat *i=3*	AgungBP	Fifin	Hermon	**Kayra**	**Elnathan**	Dion	*Liwin*	*Niken*	*Rulieta*	*Satrio*
					*[i] > [i+1], **tukar***					
Langkah 5 : Pada saat *i=4*	AgungBP	Fifin	Hermon	Elnathan	**Kayra**	**Dion**	*Liwin*	*Niken*	*Rulieta*	*Satrio*
						*[i] > [i+1], **tukar***				
Hasil :	AgungBP	Fifin	Hermon	Elnathan	Dion	**Kayra**	*Liwin*	*Niken*	*Rulieta*	*Satrio*

Siklus 6 (pada saat nilai *j=5*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**AgungBP**	**Fifin**	Hermon	Elnathan	Dion	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
		[i] < [i+1], tidak ***tukar***								
Langkah 2 : Pada saat *i=1*	AgungBP	**Fifin**	**Hermon**	Elnathan	Dion	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
			[i] < [i+1], tidak ***tukar***							
Langkah 3 : Pada saat *i=2*	AgungBP	Fifin	**Hermon**	**Elnathan**	Dion	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
				[i] > [i+1], ***tukar***						
Langkah 4 : Pada saat *i=3*	AgungBP	Fifin	Elnathan	**Hermon**	**Dion**	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
					[i] > [i+1], ***tukar***					
Hasil :	AgungBP	Fifin	Elnathan	Dion	**Hermon**	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*

Siklus 7 (pada saat nilai *j=6*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**AgungBP**	**Fifin**	Elnathan	Dion	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
		[i] < [i+1], tidak ***tukar***								
Langkah 2 : Pada saat *i=1*	AgungBP	**Fifin**	**Elnathan**	Dion	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
			[i] > [i+1], ***tukar***							
Langkah 3 : Pada saat *i=2*	AgungBP	Elnathan	**Fifin**	**Dion**	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
				[i] > [i+1], ***tukar***						
Hasil :	AgungBP	Elnathan	Dion	**Fifin**	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*

Siklus 8 (pada saat nilai *j=7*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**AgungBP**	**Elnathan**	Dion	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
		[i] < [i+1], tidak ***tukar***								
Langkah 2 : Pada saat *i=1*	AgungBP	**Elnathan**	**Dion**	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
			[i] > [i+1], ***tukar***							
Hasil :	AgungBP	Dion	**Elnathan**	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*

Siklus 9 (pada saat nilai *j=8*)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Langkah 1 : Pada saat *i=0*	**AgungBP**	**Dion**	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*
		[i] < [i+1], tidak ***tukar***								
Hasil :	AgungBP	**Dion**	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*

Hasil Akhir

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	*Rulieta*	*Satrio*

Untuk mengimplementasikan teknik bubblesort, anda dapat memperhatikan program 5.1 berikut ini.

```
//-------------------------------------------------------
//---  Fungsi untuk Mengurutkan Data (BubbleSort)   ---
//-------------------------------------------------------
public static void mengurutkanDataBubble(formatBiodata
biodataMahasiswa[])
{
    formatBiodata biodataSementara = new formatBiodata();
    int indeksTerakhir = N-1;
    for (int j=0;j<=indeksTerakhir - 1; j++)
    {
```

```
        for (int i=0; i<=indeksTerakhir -1 -j;  i++)
        {
            // perintah dibawah ini identik dengan if (nama[i]>nama[i+1])
            if (biodataMahasiswa[i].nama.compareTo
                                (biodataMahasiswa[i+1].nama) > 0)
            {   biodataSementara = biodataMahasiswa[i];
                biodataMahasiswa[i]   = biodataMahasiswa[i+1];
                biodataMahasiswa[i+1] = biodataSementara;
            }
        }
    }
}
```

b. Algoritma Selection Sort

Algoritma selection sort akan membagi larik menjadi dua bagian. Bagian kiri akan menjadi data yang sudah urut sedangkan bagian kanan adalah bagian yang masih acak.

Proses pengurutan dilakukan dengan cara mengambil sebuah elemen kunci, kemudian membandingkan dengan elemen terkecil dari larik bagian kanan. Jika elemen kunci lebih besar dari elemen terkecil dari larik bagian kanan tersebut, tukarkan kedua elemen tersebut.

Proses dilanjutkan dengan mengambil sebuah elemen kunci yang baru, kemudian membandingkan lagi elemen kunci tersebut dengan elemen terkecil dari larik bagian kanan yang masih tersisa. Jika elemen kunci lebih besar dari elemen terkecil dari larik bagian kanan tersebut, lakukan kembali pertukarkan kedua elemen seperti langkah sebelumnya. Proses seperti ini harus diulang terus menerus hingga larik bagian kiri penuh dan larik bagian kanan habis.

Agar mudah dipahami akan disimulasikan contoh pengurutan data berikut ini.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
DATA AWAL	Rulieta	Hermon	Satrio	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Dengan metode selection sort, dapat kita lihat proses pengurutan sebagaimana tersaji pada gambar berikut ini.

LANGKAH 1

Rulieta	Hermon	Satrio	**AgungBP**	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

[0] <---bandingkan---> (terkecil)

Rulieta > AgungBP, tukarkan.

AgungBP	Hermon	Satrio	**Rulieta**	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

LANGKAH 2

AgungBP	**Hermon**	Satrio	Rulieta	Niken	Fifin	Liwin	Kayra	Elnathan	**Dion**

[1] <----------bandingkan----------> (terkecil)

Hermon > Dion, tukarkan.

AgungBP	**Dion**	Satrio	Rulieta	Niken	Fifin	Liwin	Kayra	Elnathan	**Hermon**

LANGKAH 3

AgungBP	*Dion*	**Satrio**	Rulieta	Niken	Fifin	Liwin	Kayra	**Elnathan**	Hermon

[2] <----------bandingkan----------> (terkecil)

Satrio > Elnathan, tukarkan.

AgungBP	*Dion*	**Elnathan**	Rulieta	Niken	Fifin	Liwin	Kayra	**Satrio**	Hermon

LANGKAH 4

AgungBP	*Dion*	*Elnathan*	**Rulieta**	Niken	**Fifin**	Liwin	Kayra	Satrio	Hermon

[3] <--------> (terkecil)

Rulieta > Fifin, tukarkan.

AgungBP	*Dion*	*Elnathan*	**Fifin**	Niken	**Rulieta**	Liwin	Kayra	Satrio	Hermon

LANGKAH 5

AgungBP	*Dion*	*Elnathan*	*Fifin*	**Niken**	Rulieta	Liwin	Kayra	Satrio	**Hermon**

[4] <----------bandingkan----------> (terkecil)

Niken > Hermon, tukarkan.

AgungBP	*Dion*	*Elnathan*	*Fifin*	**Hermon**	Rulieta	Liwin	Kayra	Satrio	**Niken**

LANGKAH 6

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	**Rulieta**	Liwin	**Kayra**	Satrio	Niken

[4] <--------> (terkecil)

Rulieta > Kayra, tukarkan.

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	**Kayra**	Liwin	**Rulieta**	Satrio	Niken

LANGKAH 7

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	**Liwin**	Rulieta	Satrio	**Niken**

[5] <----------> (terkecil)

Liwin < Niken, TIDAK ditukar.

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	**Liwin**	Rulieta	Satrio	**Niken**

LANGKAH 8

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	*Liwin*	**Rulieta**	Satrio	**Niken**

[6] <---------> (terkecil)

Rulieta > Niken, tukarkan.

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	*Liwin*	**Niken**	Satrio	**Rulieta**

LANGKAH 9

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	**Satrio**	**Rulieta**

[6] (terkecil)

Satrio > Rulieta, tukari

AgungBP	*Dion*	*Elnathan*	*Fifin*	*Hermon*	*Kayra*	*Liwin*	*Niken*	**Rulieta**	**Satrio**

HASIL AKHIR

Untuk mengimplementasikan teknik selection sort, anda dapat memperhatikan program 5.2 berikut ini.

```
//-------------------------------------------------------
//---  Fungsi untuk Mengurutkan Data (Selection)    ---
//-------------------------------------------------------
public static void mengurutkanDataSelection(formatBiodata
biodataMahasiswa[])
{
   formatBiodata biodataSementara = new formatBiodata();
   String teksTerkecil = "";
   int lokasi=0;

   //bagian mengurutkan dengan teknik selection
   for (int i=0;   i<=N-2;  i++)
   {
      //data pertama yang dibaca dianggap data terkecil
      teksTerkecil = "zzzzzzz";

      //menentukan bilangan terkecil mulai larik ke i+1 sampai N-1
      for (int S=i+1;  S<=N-1;  S++)
      {
         if (biodataMahasiswa[S].nama.compareTo(teksTerkecil)<0)
         {  //jika data[S] adlh bilangan terkecil, simpan di
teksTerkecil
            teksTerkecil = biodataMahasiswa[S].nama;
            //mencatat posisi dimana data terkecil ada
            lokasi = S;
         }
      }
      //membandingkan data[lokasi] yang adalah data terkecil,
      // versus data[i] yang adalah 'diagonal ke-i'
      if (biodataMahasiswa[i].nama.compareTo
                              (biodataMahasiswa[lokasi].nama)>0)
      {
         //tukar posisi
         {  biodataSementara = biodataMahasiswa[i];
            biodataMahasiswa[i] = biodataMahasiswa[lokasi];
            biodataMahasiswa[lokasi] = biodataSementara;
         }
      }
   }
}
```

c. Algoritma Insertion Sort

Algoritma Insertion sort akan membagi larik menjadi dua bagian. Bagian kiri akan menjadi data yang sudah urut sedangkan bagian kanan adalah bagian yang masih acak.

Proses pengurutan dilakukan dengan cara mengambil sebuah elemen kunci, kemudian meletakkan di tempat yang sesuai pada bagian kiri larik. Proses pencarian tempat yang sesuai dilakukan dengan membandingkan elemen kunci tersebut dengan setiap elemen yang ada di bagian kiri larik, dimulai dari kanan. Apabila ditemukan tempat yang sesuai maka elemen kunci tersebut akan disisipkan di lokasi yang dimaksud. Proses dilanjutkan dengan mengambil sebuah elemen kunci yang baru yang

ada di sebelah kanan elemen kunci yang sebelumnya, kemudian dilakukan kembali proses di atas, hingga semua data pada bagian kanan larik habis.

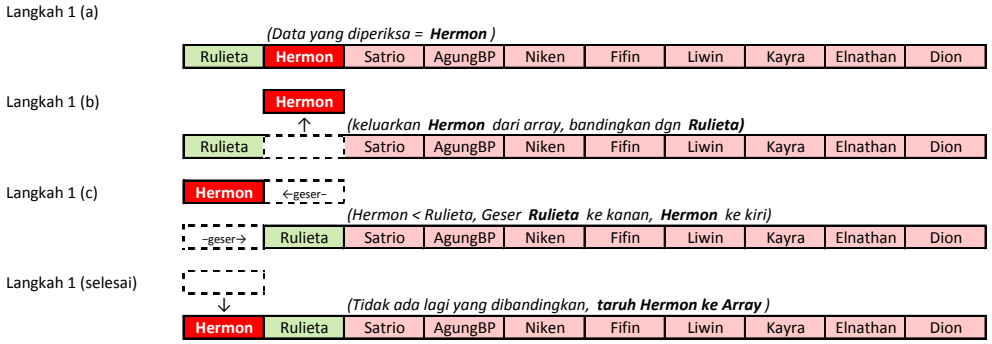
Agar mudah dipahami akan disimulasikan contoh pengurutan data berikut ini.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
DATA AWAL	Rulieta	Hermon	Satrio	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Dengan metode insertion sort, dapat kita lihat proses pengurutan sebagaimana tersaji pada gambar berikut ini.

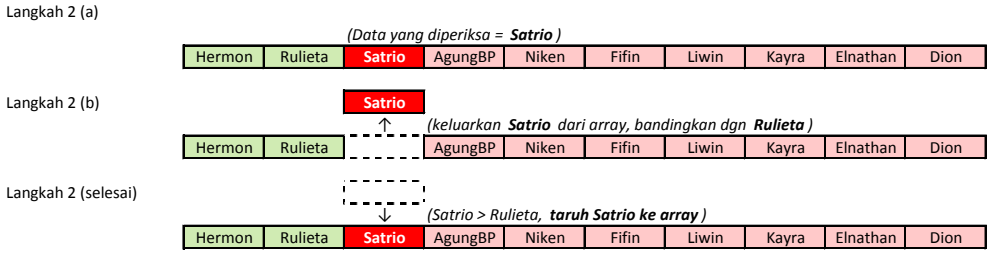
LANGKAH 1

(memeriksa orang ke-2)



LANGKAH 2

(memeriksa orang ke-3)



LANGKAH 3

(memeriksa orang ke-4)

Langkah 3 (a)

*(Data yang diperiksa = **AgungBP**)*

Hermon	Rulieta	Satrio	AgungBP	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 3 (b)

AgungBP

↑ *(keluarkan **AgungBP** dari array, bandingkan dgn **Satrio**)*

Hermon	Rulieta	Satrio		Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 3 (c)

AgungBP ←geser–

*(AgungBP < Satrio, Geser **Satrio** ke kanan, **AgungBP** ke kiri)*

Hermon	Rulieta	–geser→	Satrio	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 3 (d)

AgungBP ←geser–

*(AgungBP < Rulieta, Geser **Rulieta** ke kanan, **AgungBP** ke kiri)*

Hermon	–geser→	Rulieta	Satrio	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 3 (e)

AgungBP ←geser–

*(AgungBP < Hermon, Geser **Hermon** ke kanan, **AgungBP** ke kiri)*

–geser→	Hermon	Rulieta	Satrio	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 3 (selesai)

↓ *(Tidak ada lagi yang dibandingkan, taruh AgungBP ke Array)*

AgungBP	Hermon	Rulieta	Satrio	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

LANGKAH 4

(memeriksa orang ke-5)

Langkah 4 (a)

*(Data yang diperiksa = **Niken**)*

AgungBP	Hermon	Rulieta	Satrio	Niken	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 4 (b)

Niken

↑ *(keluarkan **Niken** dari array, bandingkan dgn **Satrio**)*

AgungBP	Hermon	Rulieta	Satrio		Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 4 (c)

Niken ←geser–

*(Niken < Satrio, Geser **Satrio** ke kanan, **Niken** ke kiri)*

AgungBP	Hermon	Rulieta	–geser→	Satrio	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 4 (d)

Niken ←geser–

*(Niken < Rulieta Geser **Rulieta** ke kanan, **Niken** ke kiri)*

AgungBP	Hermon	–geser→	Rulieta	Satrio	Fifin	Liwin	Kayra	Elnathan	Dion

Langkah 4 (selesai)

↓ *(Niken > Hermon, **taruh Niken ke array**)*

AgungBP	Hermon	Niken	Rulieta	Satrio	Fifin	Liwin	Kayra	Elnathan	Dion

LANGKAH 5

(memeriksa orang ke-6)

Langkah 5 (a)

*(Data yang diperiksa = **Fifin**)*

AgungBP	Hermon	Niken	Rulieta	Satrio	**Fifin**	Liwin	Kayra	Elnathan	Dion

Langkah 5 (b)

Fifin

*(keluarkan **Fifin** dari array, bandingkan dgn **Satrio**)* ↑

AgungBP	Hermon	Niken	Rulieta	Satrio		Liwin	Kayra	Elnathan	Dion

Langkah 5 (c)

Fifin ←geser–

*(Fifin < Satrio, Geser **Satrio** ke kanan, **Fifin** ke kiri)*

AgungBP	Hermon	Niken	Rulieta	–geser→	Satrio	Liwin	Kayra	Elnathan	Dion

Langkah 5 (d)

Fifin ←geser–

*(Fifin < Rulieta, Geser **Rulieta** ke kanan, **Fifin** ke kiri)*

AgungBP	Hermon	Niken	–geser→	Rulieta	Satrio	Liwin	Kayra	Elnathan	Dion

Langkah 5 (e)

Fifin ←geser–

*(Fifin < Niken, Geser **Niken** ke kanan, **Fifin** ke kiri)*

AgungBP	Hermon	–geser→	Niken	Rulieta	Satrio	Liwin	Kayra	Elnathan	Dion

Langkah 5 (f)

Fifin ←geser–

*(Fifin < Hermon, Geser **Hermon** ke kanan, **Fifin** ke kiri)*

AgungBP	–geser→	Hermon	Niken	Rulieta	Satrio	Liwin	Kayra	Elnathan	Dion

Langkah 5 (selesai)

↓ *(Fifin > AgungBP, **taruh Fifin ke array**)*

AgungBP	**Fifin**	Hermon	Niken	Rulieta	Satrio	Liwin	Kayra	Elnathan	Dion

LANGKAH 6

(memeriksa orang ke-7)

Langkah 6 (a)

*(Data yang diperiksa = **Liwin**)*

AgungBP	Fifin	Hermon	Niken	Rulieta	Satrio	**Liwin**	Kayra	Elnathan	Dion

Langkah 6 (b)

Liwin

*(keluarkan **Liwin** dari array, bandingkan dgn **Satrio**)* ↑

AgungBP	Fifin	Hermon	Niken	Rulieta	Satrio		Kayra	Elnathan	Dion

Langkah 6 (c)

Liwin ←geser–

*(Liwin < Satrio, Geser **Satrio** ke kanan, **Liwin** ke kiri)*

AgungBP	Fifin	Hermon	Niken	Rulieta	–geser→	Satrio	Kayra	Elnathan	Dion

Langkah 6 (d)

Liwin ←geser–

*(Liwin < Rulieta, Geser **Rulieta** ke kanan, **Liwin** ke kiri)*

AgungBP	Fifin	Hermon	Niken	–geser→	Rulieta	Satrio	Kayra	Elnathan	Dion

Langkah 6 (e)

Liwin ←geser–

*(Liwin < Niken, Geser **Niken** ke kanan, **Liwin** ke kiri)*

AgungBP	Fifin	Hermon	–geser→	Niken	Rulieta	Satrio	Kayra	Elnathan	Dion

Langkah 6 (selesai)

↓ *(Liwin < Hermon **taruh Liwin ke array**)*

AgungBP	Fifin	Hermon	**Liwin**	Niken	Rulieta	Satrio	Kayra	Elnathan	Dion

LANGKAH 7

(memeriksa orang ke-8)

Langkah 7 (a)

*(Data yang diperiksa = **Kayra**)*

AgungBP	Fifin	Hermon	Liwin	Niken	Rulieta	Satrio	**Kayra**	Elnathan	Dion

Langkah 7 (b)

Kayra

*(keluarkan **Kayra** dari array, bandingkan dgn **Satrio**)* ↑

AgungBP	Fifin	Hermon	Liwin	Niken	Rulieta	Satrio		Elnathan	Dion

Langkah 7 (c)

Kayra ←geser–

*(Kayra < Satrio, Geser **Satrio** ke kanan, **Kayra** ke kiri)*

AgungBP	Fifin	Hermon	Liwin	Niken	Rulieta	–geser→	Satrio	Elnathan	Dion

Langkah 7 (d)

Kayra ←geser–

*(Kayra < Rulieta, Geser **Rulieta** ke kanan, **kayra** ke kiri)*

AgungBP	Fifin	Hermon	Liwin	Niken	–geser→	Rulieta	Satrio	Elnathan	Dion

Langkah 7 (e)

Kayra ←geser–

*(Kayra < Niken, Geser **Niken** ke kanan, **Kayra** ke kiri)*

AgungBP	Fifin	Hermon	Liwin	–geser→	Niken	Rulieta	Satrio	Elnathan	Dion

Langkah 7 (f)

Kayra ←geser–

*(Kayra < Liwin, Geser **Liwin** ke kanan, **Kayra** ke kiri)*

AgungBP	Fifin	Hermon	–geser→	Liwin	Niken	Rulieta	Satrio	Elnathan	Dion

Langkah 7 (selesai)

↓ *(Kayra > Hermon **taruh Kayra ke array**)*

AgungBP	Fifin	Hermon	**Kayra**	Liwin	Niken	Rulieta	Satrio	Elnathan	Dion

LANGKAH 8

(memeriksa orang ke-9)

Langkah 8 (a)

*(Data yang diperiksa = **Elnathan**)*

AgungBP	Fifin	Hermon	Kayra	Liwin	Niken	Rulieta	Satrio	**Elnathan**	Dion

Langkah 8 (b)

Elnathan

*(keluarkan **Elnathan** dari array, bandingkan dgn **Satrio**)* ↑

AgungBP	Fifin	Hermon	Kayra	Liwin	Niken	Rulieta	Satrio		Dion

Langkah 8 (c)

Elnathan ←geser–

*(Elnathan < Satrio, Geser **Satrio** ke kanan, **Elnathan** ke kiri)*

AgungBP	Fifin	Hermon	Kayra	Liwin	Niken	Rulieta	–geser→	Satrio	Dion

Langkah 8 (d)

Elnathan ←geser–

*(Elnathan < Rulieta, Geser **Rulieta** ke kanan, **Elnathan** ke kiri)*

AgungBP	Fifin	Hermon	Kayra	Liwin	Niken	–geser→	Rulieta	Satrio	Dion

Langkah 8 (e)

Elnathan ←geser–

*(Elnathan < Niken, Geser **Niken** ke kanan, **Elnathan** ke kiri)*

AgungBP	Fifin	Hermon	Kayra	Liwin	–geser→	Niken	Rulieta	Satrio	Dion

Langkah 8 (f)

Elnathan ←geser–

*(Elnathan < Liwin, Geser **Liwin** ke kanan, **Elnathan** ke kiri)*

AgungBP	Fifin	Hermon	Kayra	–geser→	Liwin	Niken	Rulieta	Satrio	Dion

Langkah 8 (g)

Elnathan ←geser–

*(Elnathan < Kayra, Geser **Kayra** ke kanan, **Elnathan** ke kiri)*

AgungBP	Fifin	Hermon	–geser→	Kayra	Liwin	Niken	Rulieta	Satrio	Dion

Langkah 8 (h)

Elnathan ←geser–

*(Elnathan < Hermon, Geser **Hermon** ke kanan, **Elnathan** ke kiri)*

AgungBP	Fifin	–geser→	Hermon	Kayra	Liwin	Niken	Rulieta	Satrio	Dion

Langkah 8 (i)

Elnathan ←geser–

*(Elnathan < Fifin, Geser **Fifin** ke kanan, **Elnathan** ke kiri)*

AgungBP	–geser→	Fifin	Hermon	Kayra	Liwin	Niken	Rulieta	Satrio	Dion

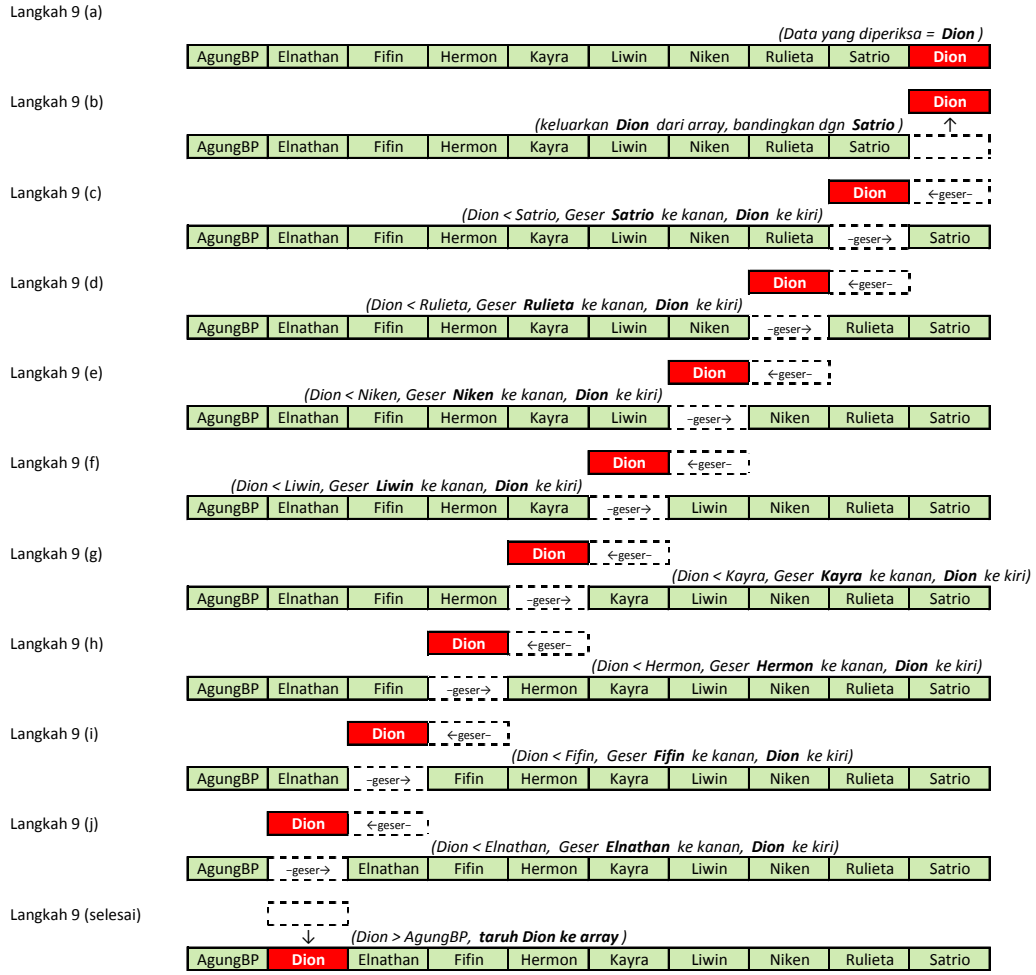
Langkah 8 (selesai)

↓ *(Elnathan > AgungBP, **taruh Elnathan ke array**)*

AgungBP	**Elnathan**	Fifin	Hermon	Kayra	Liwin	Niken	Rulieta	Satrio	Dion

LANGKAH 9

(memeriksa orang ke-10)



HASIL AKHIR

HASIL AKHIR	AgungBP	Dion	Elnathan	Fifin	Hermon	Kayra	Liwin	Niken	Rulieta	Satrio

Perhatikan program 5.3 berikut.

```
//-------------------------------------------------------
//---  Fungsi untuk Mengurutkan Data (Insertion)     ---
//-------------------------------------------------------
public static void mengurutkanDataInsertion(formatBiodata biodataMahasiswa[])
{
    formatBiodata biodataSementara = new formatBiodata();

    //untuk menentukan awal dari data sisi kanan (sisi yg masih berantakan)
    int awal;
    //untuk mencari posisi yg tepat pada sisi kiri (sisi yg sudah berurutan)
    int cari;
    awal = 1;
    while (awal <=N-1)
    {
        biodataSementara = biodataMahasiswa[awal];
        cari = awal-1;
```

```
        //cari akan bergerak dari kanan (awal-1) ke kiri
        while ( cari >= 0)
        {
            //( biodataMahasiswa[cari].nama > biodataSementara.nama )
            if (biodataMahasiswa[cari].nama.compareTo(biodataSementara.nama)>0)
            {
                biodataMahasiswa[cari+1] = biodataMahasiswa[cari];
                biodataMahasiswa[cari] = biodataSementara;
                cari--; //cari digeser kekiri 1 langkah
            }
            else
            {   biodataMahasiswa[cari+1] = biodataSementara;
                // perintah ini untuk keluar dari loop while
                cari=-1;
            }
        }
        awal++;
    }
}
```

Program 5.3

PRAKTIK

1. Praktek 1

Tambahkan program 5.1 ke dalam master program yang telah anda buat pada praktikum yang lalu. Eksekusi master program anda untuk melakukan pengurutan data secara ***Bubblesort***.

Ujilah program dengan mengentri data secara acak (nama mahasiswa dientri tidak dalam keadaan urut) kemudian lakukan pengurutan terhadap data tersebut, kemudian tampilkan data kembali. Bagaimana hasilnya? Catat dan simpulkan dalam laporan anda.

2. Praktek 2

Tambahkan program 5.2 ke dalam master program anda. Eksekusi master program anda untuk melakukan pengurutan data secara ***SelecionSort***

Ujilah program dengan mengentri data secara acak (nama mahasiswa dientri tidak dalam keadaan urut) kemudian lakukan pengurutan terhadap data tersebut, kemudian tampilkan data kembali. Bagaimana hasilnya? Catat dan simpulkan dalam laporan anda

3. Praktek 3

Tambahkan program 5.3 ke dalam master program yang telah anda buat pada praktikum yang lalu. Eksekusi master program anda untuk melakukan pengurutan data secara ***InsertionSort***.

Ujilah program dengan mengentri data secara acak (nama mahasiswa dientri tidak dalam keadaan urut) kemudian lakukan pengurutan terhadap data tersebut, kemudian tampilkan data kembali. Bagaimana hasilnya? Catat dan simpulkan dalam laporan anda

4. Praktek 4

Sekarang pikirkan, kemudian modifikasilah ketiga program sorting di atas supaya nama-nama dalam larik menjadi urut turun (descending).

LATIHAN

1. Modifikasilah program 5.1 dan program 5.2 agar dapat mengurutkan data di dalam larik ***berdasarkan IPK***
2. .....

TUGAS

1. Buatlah sebuah program Sorting dengan metode insertion dalam java menggunakan dengan menu :
   - insert,
   - View,
   - Sort,
   - Exit

REFERENSI

Disadur dari Buku Ajar Struktur Data Menggunakan Java, Agung Budi Prasetyo, 2017, hal: 43-57, http://agungbudiprasetyo.atspace.com/buku/index.html, diakses pada 12:07 PM 8/06/2019.