



# **JOBSHEET 08**

## **Bubble Sort and Insertion Sort**

| Fakultas | Proram Studi       | Kode MK      | Waktu        |
|----------|--------------------|--------------|--------------|
| Teknik   | Teknik Informatika | INF1.62.2014 | 2 x 50 Menit |

### **TUJUAN PRAKTIKUM**

1. Mahasiswa memahami konsep antrian atau queue dalam pemrograman C.
2. Mahasiswa memahami operasi yang ada dalam pencarian linear dan biner.
3. Mahasiswa mampu mengimplementasikan pencarian struktur data linear atau binary menggunakan pemrograman C dengan IDE.

### **HARDWARE & SOFTWARE**

1. Personal Computer
2. DevC++ IDE

### **TEORI SINGKAT**

#### **1. Pengertian Pengurutan Data**

pengurutan data adalah salah satu proses yang banyak dijumpai dalam aplikasi dengan menggunakan computer. Pengurutan membuat data disusun dalam keadaan urut menurut aturan tertentu; missal dari nilai kecil menuju ke nilai terbesar untuk bilangan atau dari A ke Z untuk string. Sebagai contoh, data yang telah diurutkan dalam sebuah laporan memudahkan kita untuk dibaca dan dipahami oleh si pemakai. Selain itu, data yang urut memudahkan pencarian. Dalam praktik banyak kita jumpai hal-hal yang menggunakan pengurutan data. Berikut contohnya :

- buku telepon berisi informasi yang telah diurutkan tanpa membedakan huruf kecil dan huruf besar.
- Kamus berisi kata-kata yang telah diurutkan tanpa mebedakan huruf kecil dan huruf besar.
- Laporan penjuala disusun berdasarkan produk terlaris hingga paling sedikit terjual.

- File-file dalam direktori ditampilkanurut berdasarkan attribute tertentu
- Indeks buku berisi daftar istilah yang memudahkan pembaca mendapatkan lokasi halaman yang berisi istilah tersebut.
- Glossary dalam buku teks berisi istilah dan definisinya, dengan nama-nama istilah disusun urut.

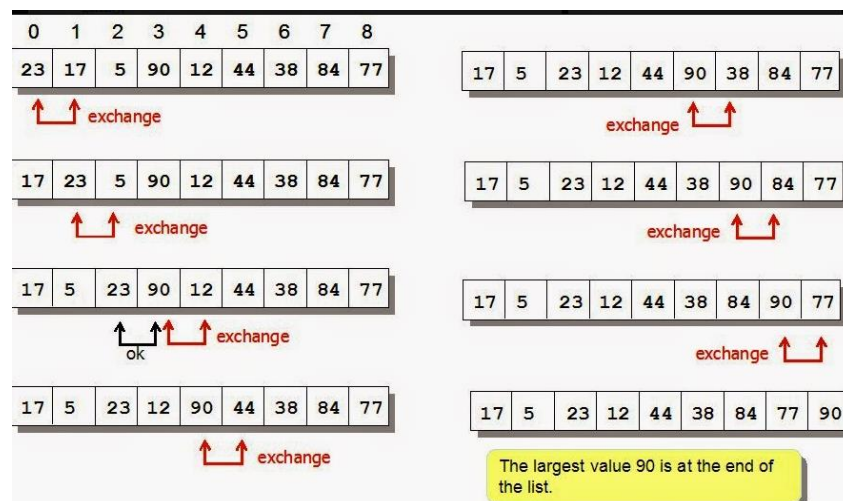
## 2. Teknik Pengurutan Data

Ada beberapa cara yang digunakan untuk mengurutkan data. Beberapa teknik terkenal untuk mengurutkan data, yaitu bubble sort, insertion sort, selection sort, quicksort, mergesort, heapsort, dan binary sort. Tiga yang pertama yaitu bubble sort, insertion sort, dan selection sort dikenal sebagai cara pengurutan dasar.

### a. Bubble Sort

Bubble Sort adalah salah satu algoritma untuk sorting data, atau kata lainnya mengurutkan data dari yang terbesar ke yang terkecil atau sebaliknya (Ascending atau Descending). Bubble sort (metode gelembung) adalah metode/algoritma pengurutan dengan dengan cara melakukan penukaran data dengan tepat disebelahnya secara terus menerus sampai bisa dipastikan dalam satu iterasi tertentu tidak ada lagi perubahan. Jika tidak ada perubahan berarti data sudah terurut. Disebut pengurutan gelembung karena masing-masing kunci akan dengan lambat menggelembung ke posisinya yang tepat. Berikut ilustrasinya

Terdapat bilangan 23, 17, 5, 90, 12, 44, 38, 84, 77 diurutkan dengan bubble sort ascending (dari kecil ke paling besar nilainya)



Pencarian biner adalah sebuah algoritma untuk menemukan posisi elemen dalam sebuah data yang diurutkan. Dalam pendekatan ini, elemen selalu

### b. Insertion Sort

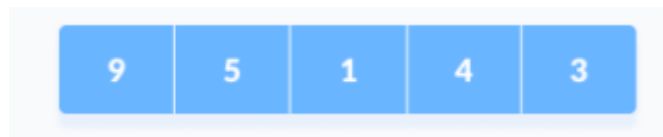
Insertion sort bekerja sama halnya dengan saat kita bermain mengurutkan kartu di tangan. Menggunakan asumsi bahwa kartu pertama telah diurutkan kemudian pilih sebuah kartu acak

yang belum diurutkan. Jika kartu tersebut lebih besar daripada kartu yang di tangan, maka kartu diletakkan di sebelah kanan, sebaliknya jika kartu yang diambil kecil diletakkan sebelah kiri. Cara yang sama dilakukan untuk pengambilan kartu berikutnya.

Pendekatan yang sama digunakan dalam Insertion sort

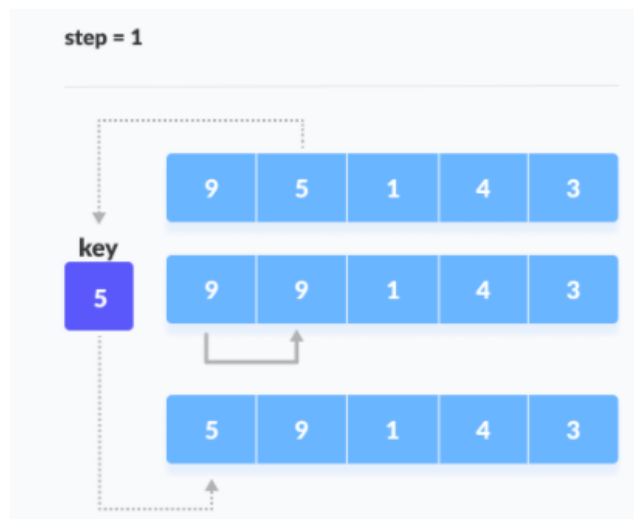
Insertion sort adalah sebuah algoritma yang meletakkan sebuah data/elemen yang belum urut pada tempat yang cocok pada setiap siklus pengurutan.

Contoh Insertion sort bekerja :

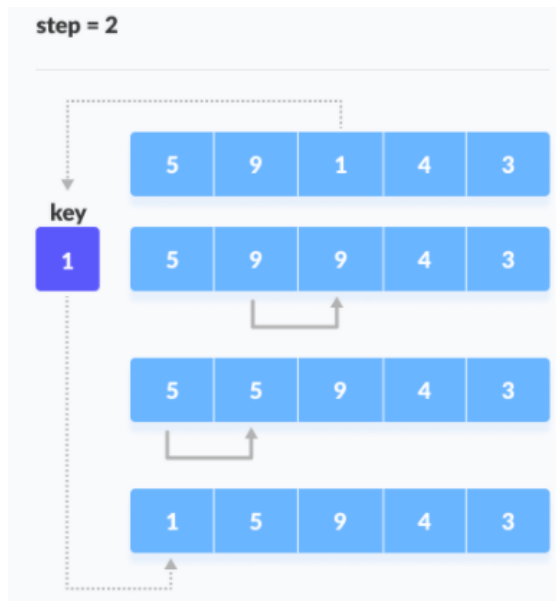


1. Elemen pertama pada array di atas diasumsikan telah diurutkan. Ambil elemen/data kedua dan simpan secara terpisah dalam ruang yang dinamakan **Key**

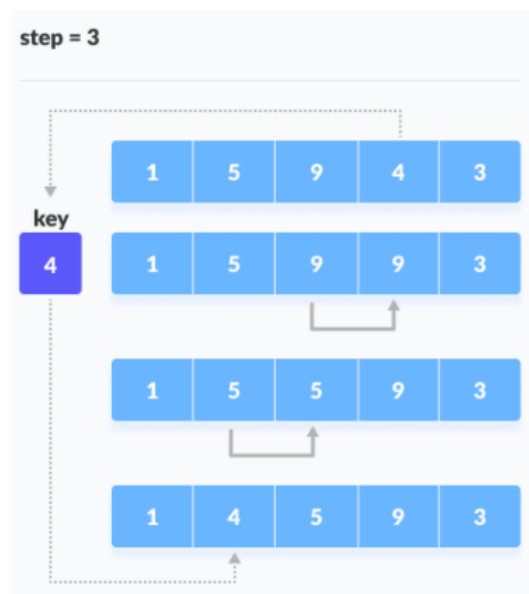
Compare key with the first element. If the first element is greater than key, then key is placed in front of the first element. Bandingkan Elemen pada Key dengan elemen pertama, jika elemen pertama lebih besar daripada key , tempatkan key di depan elemen pertama



2. Tahap pertama dua elemen telah diurutkan, Ambil elemen ketiga dan bandingkan dengan elemen sebelah kirinya. Letakkan elemen paling kecil paling depan. Jika tidak lebih kecil maka tempatkan di ujung array.



3. Langkah ketiga sama halnya dengan di atas



## PERCOBAAN

1. Bubble dengan Array Ascending

```
#include <stdio.h>

void bubbleSort(int array[], int size)
{
    for (int step = 0; step < size - 1; ++step)
    {
        for (int i = 0; i < size - step - 1; ++i)
        {
            // To sort in descending order, change ">" to "<".
            if (array[i] > array[i + 1])
            {
                // Swap array[i] and array[i + 1]
                int temp = array[i];
                array[i] = array[i + 1];
                array[i + 1] = temp;
            }
        }
    }
}
```

```
        int temp = array[i];
        array[i] = array[i + 1];
        array[i + 1] = temp;
    }
}
}
}
void printArray(int array[], int size)
{
    for (int i = 0; i < size; ++i)
    {
        printf("%d ", array[i]);
    }
    printf("\n");
}
int main()
{
    int data[] = {-2, 45, 0, 11, -9};
    int size = sizeof(data) / sizeof(data[0]);
    bubbleSort(data, size);
    printf("Sorted Array in Ascending Order:\n");
    printArray(data, size);
}
```

## 2. Bubble Sort Bentuk Kedua

```
#include<stdio.h>
int main ()
{
    int i, j,temp;
    int a[10] = { 10, 9, 7, 101, 23, 44, 12, 78, 34, 23};
    for(i = 0; i<10; i++)
    {
        for(j = i+1; j<10; j++)
        {
            if(a[j] > a[i])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    printf("Printing Sorted Element List ...\n");
    for(i = 0; i<10; i++)
    {
        printf("%d\n",a[i]);
    }
}
```

## 3. Insertion Sort – Ascending

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

#define MAX 20
```

```
void InsertionSort(int arr[])
{
    int i, j, key;
    for (i=1; i<MAX; i++)
    {
        key = arr[i];
        j = i-1;
        while(j>=0 && arr[j]>key)
        {
            arr[j+1] = arr[j];
            j--;
        }
        arr[j+1] = key;
    }
}

int main()
{
    int data_awal[MAX], data_urut[MAX];
    int i;
    long long k1, k2;
    printf("Sebelum pengurutan : \n");
    for(i=0; i<MAX; i++)
    {
        srand(time(NULL) * (i+1));
        data_awal[i] = rand() % 100 + 1;
        printf("%d ", data_awal[i]);
    }
    printf("\nSetelah pengurutan : \n");
    for(i=0; i<MAX; i++)
        data_urut[i] = data_awal[i];
    time(&k1);
    InsertionSort(data_urut);
    time(&k2);
    for(i=0; i<MAX; i++)
        printf("%d ", data_urut[i]);
    printf("\nWaktu = %ld\n", k2-k1);
}
```

#### 4. Bubble Sort with Iteration

```
#include <stdio.h>
#include <stdbool.h>
#define MAX 10

int list[MAX] = {1,8,4,6,0,3,5,2,7,9};

void display() {
    int i;
    printf("[");

    // navigate through all items
    for(i = 0; i < MAX; i++) {
        printf("%d ",list[i]);
    }

    printf("]\n");
}

void bubbleSort() {
```

```
int temp;
int i,j;
bool swapped = false;

// loop through all numbers
for(i = 0; i < MAX-1; i++) {
    swapped = false;

    // loop through numbers falling ahead
    for(j = 0; j < MAX-1-i; j++) {
        printf("    Items compared: [ %d, %d ] ",
list[j],list[j+1]);

        // check if next number is lesser than current no
        // swap the numbers.
        // (Bubble up the highest number)

        if(list[j] > list[j+1]) {
            temp = list[j];
            list[j] = list[j+1];
            list[j+1] = temp;

            swapped = true;
            printf(" => swapped [%d, %d]\n",list[j],list[j+1]);
        } else {
            printf(" => not swapped\n");
        }
    }

    // if no number was swapped that means
    // array is sorted now, break the loop.
    if(!swapped) {
        break;
    }

    printf("Iteration %d#: ", (i+1));
    display();
}

}

int main() {
    printf("Input Array: ");
    display();
    printf("\n");

    bubbleSort();
    printf("\nOutput Array: ");
    display();
}
```

#### 5. Insertion Sort with Iteration

```
#include <stdio.h>
#include <stdbool.h>

#define MAX 7
```

```
int intArray[MAX] = {4,6,3,2,1,9,7};

void printline(int count) {
    int i;

    for(i = 0;i < count-1;i++) {
        printf("=");
    }

    printf("\n");
}

void display() {
    int i;
    printf("[");

    // navigate through all items
    for(i = 0;i < MAX;i++) {
        printf("%d ",intArray[i]);
    }

    printf("]\n");
}

void insertionSort() {

    int valueToInsert;
    int holePosition;
    int i;

    // loop through all numbers
    for(i = 1; i < MAX; i++) {

        // select a value to be inserted.
        valueToInsert = intArray[i];

        // select the hole position where number is to be inserted
        holePosition = i;

        // check if previous no. is larger than value to be inserted
        while (holePosition > 0 && intArray[holePosition-1] >
valueToInsert) {
            intArray[holePosition] = intArray[holePosition-1];
            holePosition--;
            printf(" item moved : %d\n" , intArray[holePosition]);
        }

        if(holePosition != i) {
            printf(" item inserted : %d, at position : %d\n" ,
valueToInsert,holePosition);
            // insert the number at hole position
            intArray[holePosition] = valueToInsert;
        }

        printf("Iteration %d#:",i);
        display();
    }
}
```



```
int main() {  
    printf("Input Array: ");  
    display();  
    printline(50);  
    insertionSort();  
    printf("Output Array: ");  
    display();  
    printline(50);  
}
```

### **TUGAS**

1. Carilah contoh aplikasi yang mengimplementasikan *bubble sort* dan *insertion sort* serta jelaskan bagaimana aplikasi tersebut bekerja sesuai dengan prinsip kedua metode sorting tersebut!

### **DAFTAR PUSTAKA**

1. Kernighan, Brian W, & Ritchie, Dennis M. 1988. The Ansi C Programming Language Second Edition, Prentice-Hall.
2. Cipta Ramadhani. 2015. Dasar Algoritma & Struktur Data. Yogyakarta: ANDI.