Praktikum 8

Pengurutan (Sorting)

Bubble Sort, Shell Sort

POKOK BAHASAN:

- ✓ Konsep pengurutan dengan *bubble sort* dan *shell sort*
- ✓ Struktur data proses pengurutan
- ✓ Implementasi algoritma pengurutan *bubble sort* dan *shell sort* dalam Bahasa C

TUJUAN BELAJAR:

Setelah melakukan praktikum dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami karakteristik algoritma pengurutan *bubble sort dan shell sort.*
- ✓ Mengimplementasikan algoritma *bubble sort dan shell sort* dalam bentuk flowchart.
- ✓ Membuat diagram alir dan mengimplementasikan algoritma pada suatu permasalahan.

DASAR TEORI:

1. METODE GELEMBUNG (*BUBBLE SORT*)

Metode gelembung (*bubble sort*) sering juga disebut dengan metode penukaran (*exchange sort*) adalah metode yang mengurutkan data dengan cara membandingkan masing-masing elemen, kemudian melakukan penukaran bila perlu.

Ilustrasi dari langkah-langkah pengurutan dengan algoritma gelembung (*bubble sort*) dapat dilihat pada tabel berikut :

Iterasi	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
Awal	12	35	9	11	3	17	23	15	31	20
I=1; j=9	12	35	9	11	3	17	23	15	**31**	**20**
j=8	12	35	9	11	3	17	23	**15**	**20**	31
j=7	12	35	9	11	3	17	**23**	**15**	20	31
j=6	12	35	9	11	3	**17**	**15**	23	20	31
j=5	12	35	9	11	**3**	**15**	17	23	20	31
j=4	12	35	9	**11**	**3**	15	17	23	20	31
j=3	12	35	**9**	**3**	11	15	17	23	20	31
j=2	12	**35**	**3**	9	11	15	17	23	20	31
j=1	**12**	**3**	35	9	11	15	17	23	20	31
i=2; j=9	3	12	35	9	11	15	17	23	**20**	**31**
j=8	3	12	35	9	11	15	17	**23**	**20**	31
j=7	3	12	35	9	11	15	**17**	**20**	23	31
j=6	3	12	35	9	11	**15**	**17**	20	23	31
j=5	3	12	35	9	**11**	**15**	17	20	23	31
j=4	3	12	35	**9**	**11**	15	17	20	23	31
j=3	3	12	**35**	**9**	11	15	17	20	23	31
j=2	3	**12**	**9**	35	11	15	17	20	23	31
i=3; j=9	3	9	12	35	11	15	17	20	**23**	**31**
j=8	3	9	12	35	11	15	17	**20**	**23**	31
j=7	3	9	12	35	11	15	**17**	**20**	23	31
j=6	3	9	12	35	11	**15**	**17**	20	23	31
j=5	3	9	12	35	**11**	**15**	17	20	23	31
j=4	3	9	12	**35**	**11**	15	17	20	23	31
j=3	3	9	**12**	**11**	35	15	17	20	23	31
i=4; j=9	3	9	11	12	35	15	17	20	**23**	**31**
j=8	3	9	11	12	35	15	17	**20**	**23**	31
j=7	3	9	11	12	35	15	**17**	**20**	23	31
j=6	3	9	11	12	35	**15**	**17**	20	23	31
j=5	3	9	11	12	**35**	**15**	17	20	23	31
j=4	3	9	11	**12**	**15**	35	17	20	23	31
i=5; j=9	3	9	11	12	15	35	17	20	**23**	**31**
j=8	3	9	11	12	15	35	17	**20**	**23**	31
j=7	3	9	11	12	15	35	**17**	**20**	23	31
j=6	3	9	11	12	15	**35**	**17**	20	23	31
j=5	3	9	11	12	**15**	**17**	35	20	23	31
i=6; j=9	3	9	11	12	15	17	35	20	**23**	**31**
j=8	3	9	11	12	15	17	35	**20**	**23**	31
j=7	3	9	11	12	15	17	**35**	**20**	23	31
j=6	3	9	11	12	15	**17**	**20**	35	23	31
i=7; j=9	3	9	11	12	15	17	20	35	**23**	**31**
j=8	3	9	11	12	15	17	20	**35**	**23**	31
j=7	3	9	11	12	15	17	**20**	**23**	35	31
i=8; j=9	3	9	11	12	15	17	20	23	**35**	**31**
j=8	3	9	11	12	15	17	20	23	31	35
Akhir	3	9	11	12	15	17	20	23	31	35

Proses pengurutan metode gelembung ini menggunakan dua proses *looping. Looping* pertama melakukan pengulangan dari elemen ke 2 sampai dengan elemen ke *Max*-1 (misalnya variable *i*), sedangkan *looping* kedua melakukan pengulangan menurun dari elemen ke *Max* sampai elemen ke *i* (misalnya variable *j*). Pada setiap pengulangan, elemen ke *j*-1 dibandingkan dengan elemen ke *j*. Apabila data ke *j*-1 lebih besar daripada data ke *j*, dilakukan penukaran.

2. METODE SHELL (*SHELL SORT*)

Metode ini disebut juga dengan metode pertambahan menurun (*diminishing increment*) yang dikembangkan oleh Donald L. Shell, sehingga sering disebut dengan Metode Shell Sort. Metode ini mengurutkan data dengan cara membandingkan suatu data dengan data lain yang memiliki jarak tertentu, kemudian dilakukan penukaran bila diperlukan

Ilustrasi langkah-langkah pengurutan dengan algoritma shell (*shell sort*) dapat dilihat pada tabel berikut :

Iterasi	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
Awal	12	35	9	11	3	17	23	15	31	20
Jarak=5	12	35	9	11	3	17	23	15	31	20
i=6	12	**35**	9	11	3	17	**23**	15	31	20
Jarak=2	12	23	9	11	3	17	35	15	31	20
i=2	**12**	23	**9**	11	3	17	35	15	31	20
i=3	9	**23**	12	**11**	3	17	35	15	31	20
i=4	9	11	**12**	23	**3**	17	35	15	31	20
i=5	9	11	3	**23**	12	**17**	35	15	31	20
i=7	9	11	3	17	12	**23**	35	**15**	31	20
i=8	9	11	3	17	12	15	**35**	23	**31**	20
i=9	9	11	3	17	12	15	31	**23**	35	**20**
i=2	**9**	11	**3**	17	12	15	31	20	35	23
i=3	3	**11**	9	**17**	12	15	31	20	35	23
Jarak=1	3	11	9	15	12	17	31	20	35	23
i=2	3	**11**	**9**	15	12	17	31	20	35	23
i=4	3	9	11	**15**	**12**	17	31	20	35	23
i=7	3	9	11	12	15	17	**31**	**20**	35	23
i=9	3	9	11	12	15	17	20	31	**35**	**23**
i=1	**3**	**9**	11	12	15	17	20	31	23	35
i=8	3	9	11	12	15	17	20	**31**	**23**	35
i=9	3	9	11	12	15	17	20	23	**31**	**35**
Akhir	3	9	11	12	15	17	20	23	31	35

Proses pengurutan dengan metode Shell dapat dijelaskan sebagai berikut :

- Pertama-tama adalah menentukan jarak mula-mula dari data yang akan dibandingkan, yaitu *Max* / 2. Data pertama dibandingkan dengan data dengan jarak *Max* / 2. Apabila data pertama lebih besar dari data ke *Max* / 2 tersebut maka kedua data tersebut ditukar. Kemudian data kedua dibandingkan dengan jarak yang sama yaitu *Max* / 2. Demikian seterusnya sampai seluruh data dibandingkan sehingga semua data ke-*j* selalu lebih kecil daripada data ke-(*j* + *Max* / 2).
- Pada proses berikutnya, digunakan jarak (*Max* / 2) / 2 atau *Max* / 4. Data pertama dibandingkan dengan data dengan jarak *Max* / 4. Apabila data pertama lebih besar dari data ke *Max* / 4 tersebut maka kedua data tersebut ditukar. Kemudian data kedua dibandingkan dengan jarak yang sama yaitu *Max* / 4. Demikian seterusnya sampai seluruh data dibandingkan sehingga semua data ke-*j* lebih kecil daripada data ke-(*j* + *Max* / 4).
- Pada proses berikutnya, digunakan jarak (*Max* / 4) / 2 atau *Max* / 8. Demikian seterusnya sampai jarak yang digunakan adalah 1.

TUGAS PENDAHULUAN:

1. Buatlah flowchart untuk operasi pengurutan bilangan bulat dengan algoritma *bubble sort* dan *shell sort*.
2. Buatlah deklarasi data pegawai dan flowchart fungsi untuk mengurutkan data dengan metode *bubble sort* dan *shell sort* sebagai berikut :
   - Data bertipe rekaman bernama Pegawai yang mempunyai 4 data yaitu :
     - `NIP`, bertipe bulat
     - `Nama`, bertipe string
     - `Alamat`, bertipe string
     - `Golongan`, bertipe char
   - Data diurutkan dengan urut naik berdasarkan NIP

PERCOBAAN:

1. Buatlah workspace menggunakan Visual C++.

2. Buatlah file *C source* untuk metode pengurutan *bubble sort* dan *shell sort* pada project SORTING yang telah dibuat pada praktikum 7.
3. Cobalah untuk masing-masing percobaan di bawah dengan menambahkan menu pilihan metode pengurutan pada program utama.

Percobaan 1 : Implementasi pengurutan dengan metode gelembung (*bubble sort*)

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

#define MAX 10

int Data[MAX];

// Prosedur menukar data
void Tukar (int *a, int *b)
{
      int temp;
      temp = *a;
      *a = *b;
      *b = temp;
}

// Prosedur pengurutan metode gelembung
void BubbleSort()
{
      int i, j;
      for(i=1; i<MAX-1; i++)
            for(j=MAX-1; j>=i; j--)
                  if(Data[j-1] > Data[j])
                        Tukar(&Data[j-1], &Data[j]);
}

void main()
{
   int i;
   srand(0);

// Membangkitkan bilangan acak
   printf("DATA SEBELUM TERURUT");
   for(i=0; i<MAX; i++)
   {
         Data[i] = (int) rand()/1000+1;
         printf("\nData ke %d : %d ", i, Data[i]);
   }

   BubbleSort();

   // Data setelah terurut
   printf("\nDATA SETELAH TERURUT");
   for(i=0; i<MAX; i++)
   {
         printf("\nData ke %d : %d ", i, Data[i]);
   }
}
```

Percobaan 2 : Implementasi pengurutan dengan metode shell (*shell sort*)

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

#define MAX 10

int Data[MAX];

// Prosedur menukar data
void Tukar (int *a, int *b)
{
      int temp;
      temp = *a;
      *a = *b;
      *b = temp;
}

// Prosedur pengurutan metode Shell
void ShellSort()
{
      int Jarak, i, j;
      bool Sudah;
      Jarak = MAX;
      while(Jarak > 1)
      {
            Jarak = Jarak / 2;
            Sudah = false;
            while(!Sudah)
            {
                  Sudah = true;
                  for(j=0; j<MAX-Jarak; j++)
                  {
                        i = j + Jarak;
                        if(Data[j] > Data[i])
                        {
                              Tukar(&Data[j], &Data[i]);
                              Sudah = false;
                        }
                  }
            }
      }
}
```

```
void main()
{
   int i;
   srand(0);

// Membangkitkan bilangan acak
   printf("DATA SEBELUM TERURUT");
   for(i=0; i<MAX; i++)
   {
         Data[i] = (int) rand()/1000+1;
         printf("\nData ke %d : %d ", i, Data[i]);
   }

   ShellSort();

   // Data setelah terurut
   printf("\nDATA SETELAH TERURUT");
   for(i=0; i<MAX; i++)
   {
         printf("\nData ke %d : %d ", i, Data[i]);
   }
}
```

LATIHAN:

1. Tambahkan kode program untuk menampilkan perubahan setiap iterasi dari proses pengurutan dengan metode gelembung dan shell.
2. Tambahkan kode program untuk menghitung banyaknya perbandingan dan pergeseran pada algoritma gelembung dan shell.
3. Tambahkan pada project Latihan pada praktikum 7 dan implementasikan pengurutan data Pegawai pada tugas pendahuluan dengan ketentuan :.
   a. Metode pengurutan dapat dipilih.
   b. Pengurutan dapat dipilih secara urut naik atau turun.
   c. Pengurutan dapat dipilih berdasarkan NIP dan NAMA.
   d. Gunakan struktur data array.
4. Berikan kesimpulan dari percobaan dan latihan yang telah Anda lakukan.