

SORTING



Nama

Email

Contact



Sorting Method(Part 1)

- Bubble Sort
- Selection Sort
- Straight Insertion Sort
- Binary Insertion Sort
- Shell Sort



Iterative Bubble Sort Algorithm(Only)

1. Deklarasi fungsi untuk bubble sort dengan parameter jumlah data dan isi datanya(berbentuk array)
2. siapkan variabel untuk menyimpan data sementara
3. untuk ($i=0 < \text{jumlah data dengan iterasi ditambah } 1$), lakukan step 4-8
4. untuk ($j=0 < \text{jumlah data} - i - 1$ dengan iterasi ditambah 1), lakukan step 5-8
5. Jika ($\text{isi data kiri} > \text{isi data kanan}$), lakukan step 6-8(**SWAP**)
6. $\text{isi data sementara} = \text{isi data kiri}$
7. $\text{isi data kiri} = \text{isi data kanan}$
8. $\text{isi data kanan} = \text{isi data sementara}$

Source Code(Sort Only)

```
void bubble(int n, int data[]){  
    int temp;  
    for (int i = 0; i < n; ++i)  
    {  
        for (int j = 0; j < n-i-1; ++j)  
        {  
            if (data[j] > data[j+1])  
            {  
                temp = data[j];  
                data[j] = data[j+1];  
                data[j+1] = temp;  
            }  
        }  
    }  
}
```



Recursive Bubble Sort Algorithm(Only)

1. Deklarasi fungsi untuk bubble sort dengan parameter jumlah data dan isi datanya(berbentuk array)
2. **BASE CASE** -> jika (jumlah data == 1), maka keluar dari fungsi
3. **RECURSIVE CASE** -> selain itu lakukan step 4-8
4. untuk ($i=0 < \text{jumlah data} - 1$ dengan iterasi ditambah 1), lakukan step 5-8
5. jika (isi data kiri > isi data kanan), lakukan step 6-8(**SWAP**)
6. isi data sementara = isi data kiri
7. isi data kiri = isi data kanan
8. isi data kanan = isi data sementara

Source Code(Sort Only)

```
void bubbleRecursive(int n, int data[])
{
    int temp;
    if (n == 1){
        return;
    }else{
        for (int i=0; i<n-1; i++){
            if (data[i] > data[i+1]){
                temp = data[i];
                data[i] = data[i+1];
                data[i+1] = temp;
            }
            bubbleRecursive(n-1, data);
        }
    }
}
```

TUGAS(Kelompok)

Buat program(Rekursif atau Iteratif) dan algoritma salah satu algoritma sorting berikut: (Ditentukan oleh Asdos)

- Selection Sort
- Straight Insertion Sort
- Binary Insertion Sort
- Shell Sort

Buat menu untuk memilih metode pengurutan yang akan digunakan



Iterative Selection Sort Algorithm(Only)

1. Deklarasi fungsi untuk selection sort dengan parameter jumlah data dan isi datanya(berbentuk array)
2. Siapkan variabel untuk menyimpan data sementara dan menyimpan index yang memiliki data terkecil
3. untuk ($i=0 < \text{jumlah data}-1$ dengan iterasi ditambah 1), lakukan step 4-11
4. variabel penyimpanan index terkecil = i
5. untuk ($j=i+1 < \text{jumlah data}$ dengan iterasi ditambah 1), lakukan step 6-7
6. jika(data di index $j < \text{data di index terkecil}$)maka, lakukan step 7
7. variabel penyimpan index terkecil = j
8. jika(variabel penyimpan index terkecil $\neq i$), lakukan step 9-11
9. isi data sementara = isi data index i
10. isi data index i = isi data index tekecil
11. isi data index terkecil = isi data sementara

Source Code(Sort Only)

```
void selectionIterative(int n, int data[]){
    int pos_min,temp;
    for (int i=0; i < n-1; i++)
    {
        pos_min = i;
        for (int j=i+1; j < n; j++)
        {
            if (data[j] < data[pos_min]){
                pos_min=j;
            }
        }
    }

    if (pos_min != i){
        temp = data[i];
        data[i] = data[pos_min];
        data[pos_min] = temp;
    }
}
```

Iterative Shell Sort Algorithm(Only)

1. Deklarasi fungsi untuk shellsort dengan parameter jumlah data dan isi datanya(berbentuk array)
2. Siapkan variabel untuk membuat gap, i, j, dan variabel untuk menyimpan data sementara
3. untuk($\text{gap} = \text{jumlah data} / 2 > 0$ dengan iterasi $\text{gap} / 2$), lakukan step 3-8
4. untuk($i = \text{gap} < \text{jumlah data}$ dengan iterasi ditambah 1), lakukan step 4-8
5. simpan data sementara = isi data index i
6. untuk ($j = i$ sampai $j \geq \text{gap}$ dan data di index $j - \text{gap} > \text{isi data sementara}$ dengan iterasi $j - \text{gap}$), lakukan step 7
7. data index j = data index j-gap
8. data index j = data variabel sementara

Source Code(Sort Only)

```
void shellIterative(int n, int data[]){  
    int gap, i, j, temp;  
    for (gap = n/2; gap > 0; gap /= 2)  
    {  
        for (i = gap; i < n; i++)  
        {  
            temp = data[i];  
            for (j = i; j >= gap && data[j - gap] > temp; j -= gap){  
                data[j] = data[j - gap];  
            }  
            data[j] = temp;  
        }  
    }  
}
```

Iterative Straight Insertion Sort Algorithm(Only)

1. Deklarasi fungsi untuk shellsort dengan parameter jumlah data dan isi datanya(berbentuk array)
2. Siapkan variabel untuk menyimpan i,j dan data utama yg akan dikomparasi
3. untuk($i=1$ sampai $i < n$ dengan iterasi ditambah 1), lakukan step 4-9
4. data utama = data di index i
5. $j = i - 1$
6. selama($j \geq 0$ dan data index $j >$ data utama), lakukan step 7-8
7. data index $j+1 =$ data index j
8. $j = j-1$
9. data index $j+1 =$ data utama

Source Code(Sort Only)

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