



YAYASAN MEMAJUKAN ILMU DAN KEBUDAYAAN
UNIVERSITAS SIBER ASIA

Kampus Menara, Jl. RM. Harsono, Ragunan - Jakarta Selatan. Daerah Khusus Ibukota Jakarta
12550. Telp. (+6221) 27806189. asiacyberuni@acu.ac.id. www.unsia.ac.id

LEMBAR JAWABAN
UJIAN AKHIR SEMESTER
SEMESTER GENAP TAHUN AJARAN 2023/2024

Mata Kuliah : Struktur Data dan Pemrograman
Kelas : IT201
Prodi : Teknik Informatika
Nama Mahasiswa : Nico Ardy Hermawan
NIM : 230401010076
Dosen : Alun Sujjada, S.Kom., M.T

Jawaban Ujian

Berikut ini merupakan Screenshot Kode Program dan juga masing-masing outputnya:

1. Bubble Sort dan Selection Sort.

1) Setup Additional Function dan Struct untuk penulisan Code.

```
struct Profile {  
    string name, address;  
};
```

Struct dengan nama *Profile* berisikan parameter *name* dan *address*

```
// Fungsi untuk mencetak Array  
void printArray(Profile arr[], int size){  
    for (int i = 0; i < size; i++) {  
        cout << "Name: " << arr[i].name << ", Address: " << arr[i].address << endl;  
    }  
    cout << endl;  
}
```

Fungsi untuk mencetak array, digunakan untuk mencetak output agar sesuai dengan ketentuan outputnya.

2) Fungsi Bubble Sort

```
// Fungsi Bubble Sort  
void bubbleSort(Profile arr[], int limit){  
    for (int i = 0; i < limit - 1; i++) {  
        for (int j = 0; j < limit - i - 1; j++) {  
            if (arr[j].name.compare(arr[j + 1].name) > 0) {  
                Profile temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
        }  
    }  
}
```

Berikut merupakan fungsi utama daripada logika untuk penerapan Bubble Sort, menggunakan beberapa perulangan agar dapat menaruh dan melepas data agar sesuai ketika ditampilkan (cout).



YAYASAN MEMAJUKAN ILMU DAN KEBUDAYAAN UNIVERSITAS SIBER ASIA

Kampus Menara, Jl. RM. Harsono, Ragunan - Jakarta Selatan. Daerah Khusus Ibukota Jakarta
12550. Telp. (+6221) 27806189. asiacyberuni@acu.ac.id. www.unsia.ac.id

```
int main(){
    Profile arr[] = {
        {"Fahmi", "Jakarta"},
        {"Romi", "Solo"},
        {"Andri", "Jakarta"},
        {"Fadillah", "Banyuwangi"},
        {"Ruli", "Bandung"},
        {"Rudi", "Bali"},
        {"Dendi", "Purwokerto"},
        {"Zaki", "Madiun"}
    };
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Before Sort: " << endl;
    printArray(arr, n);

    cout << "Bubble Sort: " << endl;
    bubbleSort(arr, n);
    printArray(arr, n);

    return 0;
}
```

Berikut kode *main* untuk menjalankan fungsi Bubble Sort. Pertama kita define nilai array dengan mengambil *struct Profile* pada variabel *arr*. Kemudian nilai *n* merupakan angka yang akan diolah untuk melakukan perulangan pada fungsi *bubbleSort*. Kita menggunakan fungsi *printArray* untuk menampilkan hasilnya

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  ROBOT DOCUMENTATION  ROBOT OUTPUT  COMMENTS

PS C:\Users\WEXSOFT\Desktop\struktur-data> cd "c:\Users\WEXSOFT\Desktop\struktur-data\" ; if ($?) { g++ Sorting.cpp -o Sorting } ; if ($?) { .\Sorting }
Before Sort:
Name: Fahmi, Address: Jakarta
Name: Romi, Address: Solo
Name: Andri, Address: Jakarta
Name: Fadillah, Address: Banyuwangi
Name: Ruli, Address: Bandung
Name: Rudi, Address: Bali
Name: Dendi, Address: Purwokerto
Name: Zaki, Address: Madiun

Bubble Sort:
Name: Andri, Address: Jakarta
Name: Dendi, Address: Purwokerto
Name: Fadillah, Address: Banyuwangi
Name: Fahmi, Address: Jakarta
Name: Romi, Address: Solo
Name: Rudi, Address: Bali
Name: Ruli, Address: Bandung
Name: Zaki, Address: Madiun

PS C:\Users\WEXSOFT\Desktop\struktur-data> █
```

Berikut hasil output dari kodenya.

3) Fungsi Selection Sort

```
// Fungsi Selection Sort
void selectionSort(Profile arr[], int limit){
    for (int i = 0; i < limit - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < limit; j++) {
            if (arr[j].name.compare(arr[minIndex].name) < 0) {
                minIndex = j;
            }
        }
        if (minIndex != i) {
            Profile temp = arr[i];
            arr[i] = arr[minIndex];
            arr[minIndex] = temp;
        }
    }
}
```

Berikut merupakan fungsi dari Selection Sort. Memiliki 2 parameter dengan mengambil nilai dari *array Struct* dan *integer*.



YAYASAN MEMAJUKAN ILMU DAN KEBUDAYAAN UNIVERSITAS SIBER ASIA

Kampus Menara, Jl. RM. Harsono, Ragunan - Jakarta Selatan. Daerah Khusus Ibukota Jakarta
12550. Telp. (+6221) 27806189. asiacyberuni@acu.ac.id. www.unsia.ac.id

```
int main(){
    Profile arr[] = {
        {"Fahmi", "Jakarta"},
        {"Romi", "Solo"},
        {"Andri", "Jakarta"},
        {"Fadillah", "Banyuwangi"},
        {"Ruli", "Bandung"},
        {"Rudi", "Bali"},
        {"Dendi", "Purwokerto"},
        {"Zaki", "Madiun"}
    };
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Before Sort: " << endl;
    printArray(arr, n);

    cout << "After Selection Sort: " << endl;
    selectionSort(arr, n);
    printArray(arr, n);

    return 0;
}
```

Untuk menjalankan kodenya kita masih menggunakan fungsi *main*.

```
PS C:\Users\WEXSOFT\Desktop\struktur-data> cd "c:\Users\WEXSOFT\Desktop\struktur-data\" ; if ($?) { g++ Sorting.cpp -o Sorting } ; if ($?) { .\Sorting }
Before Sort:
Name: Fahmi, Address: Jakarta
Name: Romi, Address: Solo
Name: Andri, Address: Jakarta
Name: Fadillah, Address: Banyuwangi
Name: Ruli, Address: Bandung
Name: Rudi, Address: Bali
Name: Dendi, Address: Purwokerto
Name: Zaki, Address: Madiun

After Selection Sort:
Name: Andri, Address: Jakarta
Name: Dendi, Address: Purwokerto
Name: Fadillah, Address: Banyuwangi
Name: Fahmi, Address: Jakarta
Name: Romi, Address: Solo
Name: Rudi, Address: Bali
Name: Ruli, Address: Bandung
Name: Zaki, Address: Madiun
```

Berikut ini merupakan output dari penggunaan fungsi *selectionSort*.

2. Binary Search.

1) Setup Additional Function dan Struct untuk support penulisan Code.

```
struct Element {
    int value, index;
};

// Fungsi untuk membandingkan element
bool compareElement(Element a, Element b){
    return a.value < b.value;
}
```

Kode di atas merupakan *struct Element* yang berisi parameter *value* dan *index* dalam bentuk *integer*. Kemudian terdapat fungsi *compareElement* yang akan meneruskan nilai *true* atau *false* sebagai hasil dari perbandingan.



YAYASAN MEMAJUKAN ILMU DAN KEBUDAYAAN UNIVERSITAS SIBER ASIA

Kampus Menara, Jl. RM. Harsono, Ragunan - Jakarta Selatan. Daerah Khusus Ibukota Jakarta
12550. Telp. (+6221) 27806189. asiacyberuni@acu.ac.id. www.unsia.ac.id

```
// Fungsi mencari index element
void searchElement(int arr[], int n, int x){
    Element elements[n];
    for (int i = 0; i < n; i++) {
        elements[i].value = arr[i];
        elements[i].index = i + 1;
    }

    sort(elements, elements + n, compareElement);

    int result = binarySearch(elements, 0, n - 1, x);

    // Jika elemen ditemukan
    if (result != -1) {
        cout << "Angka " << x << " ada di indeks ke " << elements[result].index;
        // Cek elemen sebelum dan sesudah untuk menangani duplikasi
        int left = result - 1;
        while (left >= 0 && elements[left].value == x) {
            cout << " dan " << elements[left].index;
            left--;
        }
        int right = result + 1;
        while (right < n && elements[right].value == x) {
            cout << " dan " << elements[right].index;
            right++;
        }
        cout << endl;
    } else {
        cout << "Angka " << x << " tidak ada dalam array" << endl;
    }
}
```

Fungsi diatas berfungsi untuk menentukan index element yang dicari nilainya.

2) Binary Search

```
// Fungsi Binary Search
int binarySearch(Element arr[], int l, int r, int x){
    while (l <= r) {
        int mid = l + (r - l) / 2;
        if (arr[mid].value == x) {
            return mid;
        }
        if (arr[mid].value < x) {
            l = mid + 1;
        } else {
            r = mid - 1;
        }
    }
    return -1;
}
```

Untuk logika daripada fungsi *binarySearch* sendiri adalah seperti di atas.

```
int main(){
    int arr[] = {19, 40, 10, 90, 2, 50, 60, 50, 1};
    int n = sizeof(arr) / sizeof(arr[0]);

    // Test case 1
    int x = 1;
    cout << "Input: " << x << endl;
    searchElement(arr, n, x);
    cout << endl;

    return 0;
}
```

Kita menjalankan case pertama dengan mencari angka 1. Ketika dijalankan kita mendapatkan output seperti berikut.

```
PS C:\Users\NEXSOFT\Desktop\struktur-d
Input: 1
Angka 1 ada di indeks ke 9
```



YAYASAN MEMAJUKAN ILMU DAN KEBUDAYAAN

UNIVERSITAS SIBER ASIA

Kampus Menara, Jl. RM. Harsono, Ragunan - Jakarta Selatan. Daerah Khusus Ibukota Jakarta
12550. Telp. (+6221) 27806189. asiacyberuni@acu.ac.id. www.unsia.ac.id

```
int main(){
    int arr[] = {19, 40, 10, 90, 2, 50, 60, 50, 1};
    int n = sizeof(arr) / sizeof(arr[0]);

    // Test case 2
    int x = 50;
    cout << "Input: " << x << endl;
    searchElement(arr, n, x);
    cout << endl;

    return 0;
}
```

Kita menjalankan case kedua dengan mencari angka 50. Ketika dijalankan kita mendapatkan output seperti berikut.

```
PS C:\Users\WEXSOFT\Desktop\struktur-data> cd "c:\
Input: 50
Angka 50 ada di indeks ke 8 dan 6
```


```
int main(){
    int arr[] = {19, 40, 10, 90, 2, 50, 60, 50, 1};
    int n = sizeof(arr) / sizeof(arr[0]);

    // Test case 3
    int x = 80;
    cout << "Input: " << x << endl;
    searchElement(arr, n, x);
    cout << endl;

    return 0;
}
```

Kita menjalankan case ketiga dengan mencari angka 80 yang tidak ada pada kumpulan data. Ketika dijalankan kita mendapatkan output seperti berikut.

```
PS C:\Users\WEXSOFT\Desktop\struktur-data>
Input: 80
Angka 80 tidak ada dalam array
```

Nilai	Tanda Tangan Dosen Pengampu / Tutor	Tanda Tangan Mahasiswa
	(Alun Sujjada, S.Kom., M.T)	 (Nico Ardy Hermawan)
Diserahkan pada Tanggal:		Tanggal Mengumpulkan:
		4 Agustus 2024