

No.	Catatan	Tanggal	Paraf
1.	Perbaiki format	8 November 2022	
2.	Perbaiki tujuan		
3.	Perbaiki kesimpulan		

NPM.

I. JUDUL PERCOBAAN

SORTING

II. TUJUAN PERCOBAAN

Adapun tujuan dari percobaan ini adalah sebagai berikut :

1. Memahami tujuan dari sorting
2. Mampu menggunakan sorting

III. TEORI DASAR

3.1 Pengertian Algoritma

Algoritma adalah suatu prosedur untuk menyelesaikan permasalahan yang tersusun secara logis dan terstruktur. Suatu algoritma pemrograman membawa keuntungan serta fungsi penting dalam aktivitas pembuatan program. Algoritma pada umumnya digunakan untuk membantu setiap orang yang ingin mengonversikan sebuah permasalahan ke bahasa pemrograman.

Dari fungsinya ini, algoritma adalah hasil dari pemikiran yang terkonsep sehingga perintah dalam algoritma dapat dilaksanakan oleh komputer lewat bahasa pemrograman.

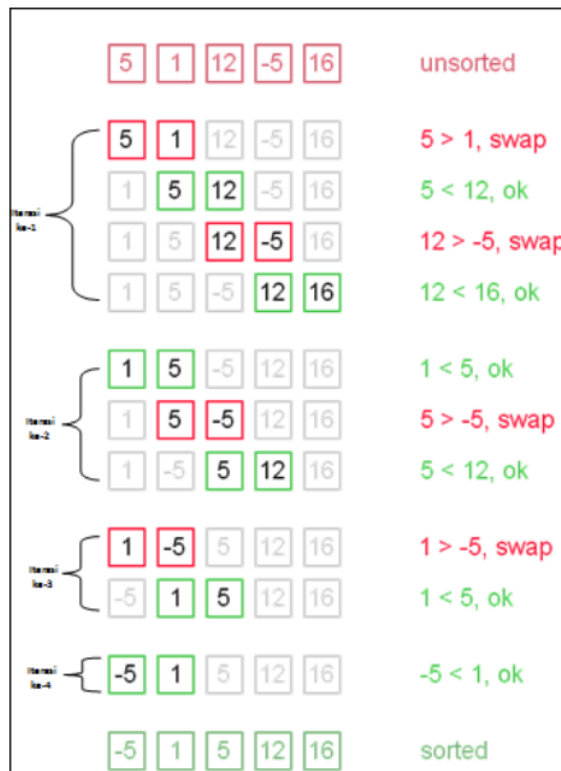
3.2 Pengertian Sorting

Pengurutan data dalam struktur data sangat penting terutama untuk data yang bertipe data numerik ataupun karakter. Pengurutan dapat dilakukan secara ascending dan descending. Ascending artinya data diurutkan dari yang nilainya paling kecil menuju nilai yang paling besar, sedangkan descending artinya data diurutkan dari yang nilainya paling besar menuju nilai yang paling kecil. Pengurutan (Sorting) adalah proses pengurutan data yang sebelumnya disusun secara acak sehingga tersusun secara teratur menurut aturan tertentu. Untuk melakukan pengurutan, terdapat berbagai macam metode yang bisa digunakan.

Beberapa metode yang sering digunakan yaitu bubble sort, exchange sort, selection sort, dan insertion sort.

3.3 Bubble sort

Bubble Sort adalah metode pengurutan algoritma dengan cara melakukan penukaran data secara terus menerus sampai bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan/penukaran. Algoritma ini menggunakan perbandingan dalam operasi antar elemennya.



Gambar 1 Bubble sort

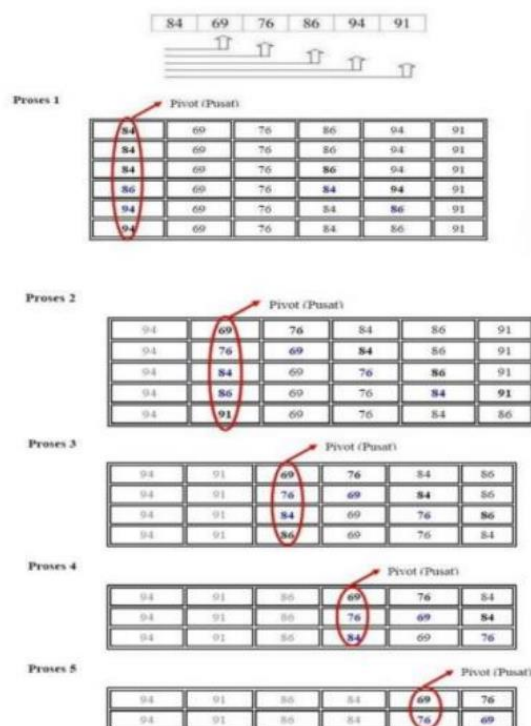
Berikut ini adalah gambaran dari algoritma bubble sort:

- Bandingkan nilai data ke-1 dan data ke-2
- Jika data ke-1 lebih besar dari data ke-2 maka tukar posisinya
- Kemudian data yang lebih besar tadi dibandingkan dengan data ke-3
- Lakukan langkah nomor 2 hingga selesai.

Bubble sort paling baik digunakan untuk mengenalkan konsep dari algoritma sorting pada pendidikan komputer karena idenya yang cukup sederhana, yaitu

mengurutkan data dengan cara membandingkan elemen sekarang dengan elemen berikutnya. Metode sorting ini sangat tidak cocok digunakan untuk hal lainnya karena algoritma ini paling lambat dan termasuk sangat tidak efisien untuk dilakukan dibandingkan dengan algoritma yang lain apalagi pengurutan dilakukan terhadap elemen yang banyak jumlahnya.

3.4 Exchange Sort



Gambar 3.4 Exchane Sort

Metode pengurutan excahange sort mirip dengan metode pengurutan Buble Sort. Perbedaannya ada pada cara membandingkan antar elemennya memiliki tentu memiliki perbedaan. Exchange sort membandingkan suatu elemen dengan elemen-elemen lainnya dalam array tersebut, dan melakukan pertukaran elemen jika perlu. Jadi ada elemen yang selalu menjadi elemen pusat (pivot).

Sedangkan Bubble sort akan membandingkan elemen pertama/terakhir dengan elemen sebelumnya/sesudahnya, kemudian elemen tersebut itu akan menjadi pusat (pivot) untuk dibandingkan dengan elemen sebelumnya/sesudahnya lagi, begitu seterusnya. Worst case pada exchange sort adalah data yang diurutkan berada pada

posisi decreasing atau menurun, dan best case pada exchange sort adalah ketika data yang sudah dalam posisi berurutan

3.5 Selection Sort

Selection Sort merupakan kombinasi antara sorting dan searching. Untuk setiap proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil atau terbesar akan dipertukarkan ke posisi yang tepat di dalam array.

Kelebihan dan kekurangan Selection Sort:

1. Kompleksitas selection sort relatif lebih kecil
2. Mudah menggabungkannya kembali, tetapi sulit membagi masalah
3. Membutuhkan metode tambahan

Untuk putaran pertama, akan dicari data dengan nilai terkecil dan data ini akan ditempatkan di indeks terkecil (data[1]), pada putaran kedua akan dicari data kedua terkecil, dan akan ditempatkan di indeks kedua (data[2]).

Selama proses, perbandingan dan pengubahan hanya dilakukan pada indeks perbandingan saja, pertukaran data secara fisik terjadi pada akhir proses. Teknik pengurutan dengan cara pemilihan elemen atau proses kerja dengan memilih elemen data terkecil untuk kemudian dibandingkan & ditukarkan dengan elemen pada data awal, dan seterusnya sampai dengan seluruh elemen sehingga akan menghasilkan pola data yang telah diurutkan

3.6 Insertion Sort

Prinsip dasar Insertion adalah secara berulang-ulang menyisipkan/ memasukkan setiap elemen ke dalam posisinya/tempatnya yang benar. Mirip dengan cara orang mengurutkan kartu, selembat demi selembat kartu diambil dan disisipkan (insert) ke tempat yang seharusnya. Pengurutan dimulai dari data ke-2 sampai dengan data terakhir, jika ditemukan data yang lebih kecil, maka akan ditempatkan (diinsert) diposisi yang seharusnya. Pada penyisipan elemen, maka elemen-elemen lain akan bergeser ke belakang

IV. PROSEDUR PERCOBAAN

Adapun source code percobaan ini adalah sebagai berikut :

4.1. Percobaan II-1 Exchange Sort Ascending

```
C++ exchangeSortAscending.cpp
1  #include <iostream>
2  using namespace std;
3
4  void tukar (int *a, int *b);
5
6  int main()
7  {
8      int n,i,j;
9      int arr[1005];
10     cin >> n;
11
12     for(i=0; i<n; i++)
13     {
14         cin >> arr[i];
15     }
16
17     for(i=0; i<n; i++)
18     {
19         for(j=n-1; j>=i; j--)
20         {
21             if(arr[i]>arr[j])
22             {
23                 tukar(&arr[i], &arr[j]);
24             }
25         }
26     }
27
28     for(i=0; i<n;i++)
29     {
30         cout << arr[i] << " ";
31     }
32
33     return 0;
34 }
35
36 void tukar(int *a, int *b)
37 {
38     int t=*a;
39     *a=*b;
40     *b=t;
41 }
```

Gambar 4.1 Percobaan II-1 Exchange Sort Ascending

4.2 Percobaan II-2 Exchange Sort Descending

```

C++ exchangeSortDescending.cpp
1  #include <iostream>
2  using namespace std;
3
4  void tukar (int *a, int *b);
5
6  int main()
7  {
8      int n,i,j;
9      int arr[1005];
10     cin >> n;
11
12     for(i=0; i<n; i++)
13     {
14         cin >> arr[i];
15     }
16
17     for(i=0; i<n-1; i++)
18     {
19         for(j=i+1; j<n; j++)
20         {
21             if(arr[i]<arr[j])
22             {
23                 tukar(&arr[i], &arr[j]);
24             }
25         }
26     }
27
28     for(i=0; i<n;i++)
29     {
30         cout << arr[i] << " ";
31     }
32
33     return 0;
34 }
35
36 void tukar(int *a, int *b)
37 {
38     int t=*a;
39     *a=*b;
40     *b=t;
41 }

```

Gambar 4.2 Percobaan II-2 Exchange Sort Descending

4.3 Percobaan II-3 Selection Sort

```

C++ selectionSort.cpp
1  #include <iostream>
2  using namespace std;
3  void tukar(int *a, int *b);
4  int main()
5  {
6      int n,i,j, pos;
7      int arr[1005];
8
9      cin >> n;
10
11     for(i=0; i<n;i++)
12     {
13         cin >> arr[i];
14     }
15
16     for(i=0; i<n-1;i++)
17     {
18         pos = i;
19         for(j=i+1; j<n; j++)
20         {
21             if(arr[j] > arr[pos])
22             {
23                 pos = j;
24             }
25         }
26         if(pos != i)
27         {
28             tukar(&arr[pos], &arr[i]);
29         }
30     }
31     for(i=0; i<n;i++)
32     {
33         cout << arr[i] << " ";
34     }
35
36     return 0;
37 }
38
39 void tukar(int *a, int *b)
40 {
41     int t=*a;
42     *a=*b;
43     *b=t;
44 }

```

Gambar 4.3 Percobaan II-3 Selection Sort

4.4 Percobaan II-4 Insertion Sort

```
C++ insertionSort.cpp
1  #include <iostream>
2  using namespace std;
3
4  void tukar(int *a, int *b);
5
6  int main()
7  {
8      int n,i,j, temp;
9      int arr[1005];
10     cin >> n;
11
12     for(i=0; i<n; i++)
13     {
14         cin >> arr[i];
15     }
16
17     for(i=1; i<n; i++)
18     {
19         temp = arr[i];
20         j = i-1;
21
22         while(arr[j]>temp && j>=0)
23         {
24             arr[j+1] = arr[j];
25             j--;
26         }
27         arr[j+1] = temp;
28     }
29
30     for(i=0; i<n;i++)
31     {
32         cout << arr[i] << " ";
33     }
34
35     return 0;
36 }
37 void tukar(int *a, int *b)
38 {
39     int t=*a;
40     *a=*b;
41     *b=t;
42 }
```

Gambar 4.4 Percobaan II-4 Insertion Sort

V. PEMBAHASAN

Adapun pembahasan untuk percobaan ini adalah sebagai berikut :

5.1 Percobaan II-1 Exchange Sort Ascending

5.1.a Source Code

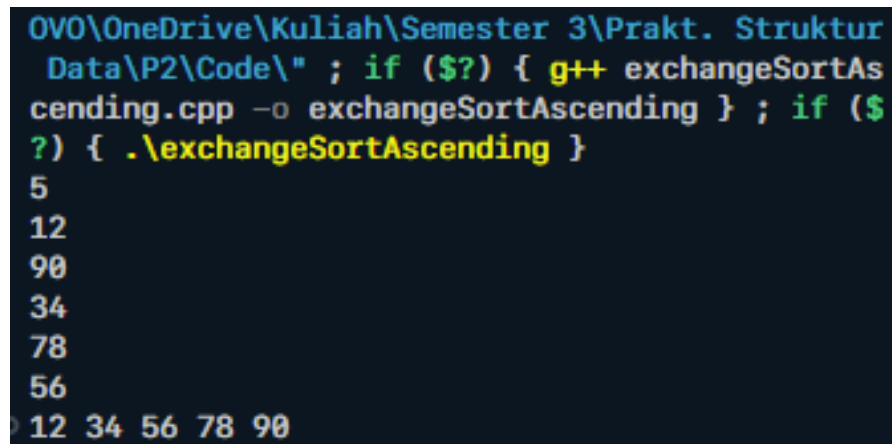
```
C++ exchangeSortAscending.cpp
1  #include <iostream>
2  using namespace std;
3
4  void tukar (int *a, int *b);
5
6  int main()
7  {
8      int n,i,j;
9      int arr[1005];
10     cin >> n;
11
12     for(i=0; i<n; i++)
13     {
14         cin >> arr[i];
15     }
16
17     for(i=0; i<n; i++)
18     {
19         for(j=n-1; j>=i; j--)
20         {
21             if(arr[i]>arr[j])
22             {
23                 tukar(&arr[i], &arr[j]);
24             }
25         }
26     }
27
28     for(i=0; i<n;i++)
29     {
30         cout << arr[i] << " ";
31     }
32
33     return 0;
34 }
35
36 void tukar(int *a, int *b)
37 {
38     int t=*a;
39     *a=*b;
40     *b=t;
41 }
```

Gambar 5.1.a Percobaan II-1 Exchange Sort Ascending

Berdasarkan gambar 5.1.a yang merupakan source code dari percobaan 1 ArrayIntAddress, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library

iostream ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler. Kemudian pada baris ke 4 terdapat deklarasi prototype void function dengan nama tukar yang berisi variabel integer pointer a dan variabel integer pointer b. Kemudian pada baris ke 6 terdapat deklarasi fungsi utama. Pada baris ke 8 terdapat deklarasi variabel integer dengan nama n, i dan j. Pada baris ke 9 terdapat deklarasi array dengan nama arr, tipe data integer, dan Panjang array 1005. Pada baris ke 10 terdapat perintah untuk menginputkan nilai ke dalam variabel n. Pada baris ke 12 terdapat perulangan for dengan banyak perulangan sebanyak nilai dari variabel n. Di dalam perulangan, yaitu pada baris ke 14, terdapat perintah untuk memasukkan nilai ke dalam array arr dari indeks ke 0 hingga n. Kemudian, setelah keluar dari perulangan, pada baris ke 17 terdapat perulangan for dengan banyak perulangan sebanyak nilai dari variabel n. Di dalam perulangan, terdapat perulangan kembali (nested loop) dengan banyak perulangan sebanyak nilai n dikurang 1. Di dalam nested loop tersebut, tepatnya pada baris ke 21, terdapat percabangan if dengan syarat jika nilai dari arr indeks ke i lebih besar dari nilai arr indeks ke j, maka jalankan fungsi tukar dengan argumen pertama adalah address dari arr[i] dan argumen kedua adalah address dari arr[j]. Keluar dari nested loop dan perulangan loop, tepatnya pada baris ke 28, terdapat perulangan for dengan banyak perulangan sebanyak nilai dari n. Di dalamnya, yaitu pada baris ke 30, terdapat perintah untuk menampilkan nilai dari array arr per indeks dimulai dari indeks ke 0 hingga indeks ke n. Terakhir dari fungsi utama terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal. Kemudian di bawah dari fungsi utama, tepatnya pada baris ke 36, terdapat pendefinisian fungsi tukar dengan parameter variabel integer pointer a dan variabel integer pointer b. Di dalamnya, yaitu pada baris ke 38, terdapat inisialisasi dari variabel t dengan nilai yaitu nilai dari variabel t dikali dengan nilai dari variabel a. Kemudian pada baris ke 39, terdapat inisialisasi dari variabel pointer a dengan nilai yaitu nilai dari variabel pointer a dikali dengan nilai dari b. Kemudian pada baris ke 40 terdapat inisialisasi dari variabel pointer b dengan nilai dari variabel t.

5.1.b Output



```
OVO\OneDrive\Kuliah\Semester 3\Prakt. Struktur
Data\P2\Code\" ; if ($?) { g++ exchangeSortAs
cending.cpp -o exchangeSortAscending } ; if ($
?) { .\exchangeSortAscending }
5
12
90
34
78
56
12 34 56 78 90
```

Gambar 5.1.b Output II-1 Exchange Sort Ascending

Berdasarkan gambar 5.1.b yang merupakan output dari percobaan II-1 Exchange Sort Ascending, dapat dilihat bahwa output pada terminal meminta user untuk memasukkan suatu nilai. Nilai ini nantinya akan menjadi penentu banyak data yang akan disorting oleh sistem, sesuai dengan perintah pada baris ke 10. Kemudian berikutnya program meminta user untuk memasukkan nilai-nilai yang akan disorting. Ini sesuai dengan perintah ke 14 yang berada di dalam perulangan. Pada percobaan ini, saya memasukkan nilai 12, 90, 34, 78, dan 56. Setelah itu, program akan menampilkan data yang sudah tersorting.

5.2 Percobaan II-2 Exchange Sort Descending

5.2.a Source Code

```
C++ insertionSort.cpp
1  #include <iostream>
2  using namespace std;
3
4  void tukar(int *a, int *b);
5
6  int main()
7  {
8      int n,i,j, temp;
9      int arr[1005];
10     cin >> n;
11
12     for(i=0; i<n; i++)
13     {
14         cin >> arr[i];
15     }
16
17     for(i=1; i<n; i++)
18     {
19         temp = arr[i];
20         j = i-1;
21
22         while([arr[j]>temp && j>=0])
23         {
24             arr[j+1] = arr[j];
25             j--;
26         }
27         arr[j+1] = temp;
28     }
29
30     for(i=0; i<n;i++)
31     {
32         cout << arr[i] << " ";
33     }
34
35     return 0;
36 }
37 void tukar(int *a, int *b)
38 {
39     int t=*a;
40     *a=*b;
41     *b=t;
42 }
43
```

Gambar 5.2.a Source Code Percobaan II-2 Exchange Sort Descending

Berdasarkan gambar 5.1.2.a yang merupakan source code dari percobaan II-2 Exchange Sort Descending, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library iostream ke dalam program yang berfungsi untuk

menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler. Kemudian pada baris ke 4, terdapat deklarasi void function dengan nama tukar dan parameter variabel integer pointer dengan nama a dan variabel integer pointer dengan nama b. Kemudian pada baris ke 6 terdapat pendefinisian dari fungsi main(). Di dalamnya, yaitu pada baris ke 8, terdapat deklarasi tiga variabel integer dengan nama n, i, dan j. Kemudian pada baris ke 9, terdapat deklarasi array arr dengan Panjang array yaitu 1005. Kemudian pada baris ke 10, terdapat perintah untuk user agar memasukkan nilai ke variabel n. Kemudian pada baris ke 12, terdapat perulangan for dengan banyak perulangan sebanyak nilai dari variabel n. Di dalamnya, terdapat perintah agar user memasukkan nilai ke array arr mulai dari indeks ke 0 hingga indeks ke n. Pada baris ke 17 terdapat perulangan for dengan banyak perulangan sebanyak nilai n-1, dan di dalamnya terdapat perulangan for (nested loop) dengan j sebagai counter, dan nilai $j=i+1$, akan berulang sebanyak nilai dari n. Di dalamnya, tepatnya pada baris ke 21, terdapat percabangan dengan kondisi jika array arr dengan indeks ke i kurang dari nilai array arr dengan indeks ke j, maka akan memanggil fungsi tukar dengan argumen alamat dari array arr indeks ke i dan alamat dari array arr dengan indeks ke j. Kemudian setelah keluar dari kedua perulangan dan percabangan tersebut, terdapat perulangan dengan banyak perulangan sebanyak nilai dari n pada baris ke 28. Perulangan ini akan mengulangi menampilkan nilai dari array arr dengan indeks ke i. Kemudian pada baris ke 33 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal. Kemudian pada baris ke 36 terdapat fungsi void tukar dengan parameter variabel integer pointer dengan nama a dan variabel integer pointer dengan nama b.

5.2.b Output

```
OVO\OneDrive\Kuliah\Semester 3\Prakt. Struktur
Data\P2\Code\" ; if ($?) { g++ exchangeSortDe
scending.cpp -o exchangeSortDescending } ; if
($?) { .\exchangeSortDescending }
5
12
90
34
78
56
90 78 56 34 12
```

Gambar 5.2.b Output Percobaan II-2 Exchange Sort Descending

Berdasarkan gambar 5.1.2.b yang merupakan output dari percobaan II-2 Exchange Sort Descending, dapat dilihat bahwa pertama, program meminta user untuk menginputkan sebuah nilai. Hal tersebut sesuai dengan perintah pada baris 10. Nilai tersebut nantinya akan menjadi jumlah dari data yang akan diurutkan. Kemudian program akan meminta user untuk memasukkan data yang akan diurutkan satu per satu. Hal itu sesuai dengan perintah pada baris ke 14 dengan perulangan ke 12. Kemudian program akan menampilkan data yang sudah diurutkan. Hal ini berdasarkan perintah pada baris ke 32 yang berada di dalam perulangan ke 30.

5.3 Percobaan II-3 Selection Sort

5.3.a Source Code

```
C++ selectionSort.cpp
1  #include <iostream>
2  using namespace std;
3  void tukar(int *a, int *b);
4  int main()
5  {
6      int n,i,j, pos;
7      int arr[1005];
8
9      cin >> n;
10
11     for(i=0; i<n;i++)
12     {
13         cin >> arr[i];
14     }
15
16     for(i=0; i<n-1;i++)
17     {
18         pos = i;
19         for(j=i+1; j<n; j++)
20         {
21             if(arr[j] > arr[pos])
22             {
23                 pos = j;
24             }
25         }
26         if(pos != i)
27         {
28             tukar(&arr[pos], &arr[i]);
29         }
30     }
31     for(i=0; i<n;i++)
32     {
33         cout << arr[i] << " ";
34     }
35
36     return 0;
37 }
38
39 void tukar(int *a, int *b)
40 {
41     int t=*a;
42     *a=*b;
43     *b=t;
44 }
```

Gambar 5.3.a Source Code Percobaan II-3 Selection Sort

Berdasarkan gambar 5.1.3.a yang merupakan source code dari percobaan 3 Selection Sort, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library

iostream ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler. Kemudian pada baris ke 3 terdapat deklarasi void prototype function dengan nama tukar dengan parameter integer pointer a dan integer pointer b. Kemudian pada baris ke 4 terdapat deklarasi fungsi main dengan tipe data integer. Pada baris ke 6 terdapat deklarasi 4 variabel bertipe data integer dengan nama n, i, j, dan pos. Pada baris ke 7 terdapat deklarasi array integer dengan nama arr dan Panjang array yaitu 1005. Pada baris ke 9 terdapat perintah bagi user untuk memasukkan nilai ke variabel n. Pada baris ke 11 terdapat perulangan sebanyak n kali dan berisi perintah untuk bagi user untuk memasukkan nilai ke array i dimulai dari indeks 0. Kemudian setelah keluar dari perulangan, pada baris ke 16, terdapat perulangan kembali dengan banyak perulangan sebanyak n-1. Di dalamnya terdapat inisialisasi nilai pos = i pada baris ke 18 dan perulangan dengan kondisi j=i+1 sebagai counter dan mengulang sebanyak j<n pada baris berikutnya. Di dalam nested loop tersebut, terdapat percabangan dengan kondisi jika arr[j] > arr[pos], maka lakukan inisialisasi nilai j kepada pos, tepatnya pada baris ke 23. Setelah keluar dari perulangan tersebut, pada baris ke 26, terdapat percabangan if dengan kondisi jika pos != i maka akan menjalankan fungsi tukar dengan argumen alamat dari array arr[pos] dan alamat dari array arr[i]. Setelah keluar dari perulangan tersebut, terdapat perulangan kembali, tepatnya pada baris ke 31, dengan banyak perulangan sebanyak nilai n. Perulangan tersebut akan mengulangi menampilkan nilai dari array indeks ke 0 hingga indeks ke n. Kemudian pada baris ke 36 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal.

Berikutnya pada baris ke 39 terdapat pendefinisian fungsi tukar dengan parameter variabel integer pointer a dan variabel integer pointer b. Di dalamnya, yaitu pada baris ke 41, terdapat inisialisasi dari variabel t dengan nilai yaitu nilai dari variabel t dikali dengan nilai dari variabel a. Kemudian pada baris ke 42, terdapat inisialisasi dari variabel pointer a dengan nilai yaitu nilai dari variabel pointer a dikali dengan nilai dari b. Kemudian pada baris ke 43 terdapat inisialisasi dari variabel pointer b dengan nilai dari variabel t.

5.3.b Output

```
OVO\OneDrive\Kuliah\Semester 3\Prakt. Struktur
Data\P2\Code\" ; if ($?) { g++ selectionSort.
cpp -o selectionSort } ; if ($?) { .\selection
Sort }
5
12
90
34
78
56
90 78 56 34 12
```

Gambar 5.3.b Output Percobaan II-3 Selection Sort

Berdasarkan gambar 5.1.3.b yang merupakan output dari percobaan selection sort terdapat, dapat dilihat bahwa pertama program meminta user untuk memasukkan banyak data yang akan diinputkan. Kemudian program meminta user untuk menginputkan data yang akan diurutkan satu per satu. Pada percobaan ini saya memasukkan 5 data yaitu 12, 90, 34, 78, dan 56. Pada

5.4 Percobaan II-4 Insertion Sort

5.4.a Source Code

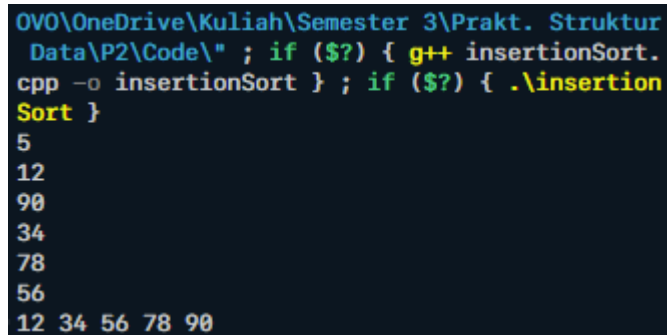
```
C++ insertionSort.cpp
1  #include <iostream>
2  using namespace std;
3
4  void tukar(int *a, int *b);
5
6  int main()
7  {
8      int n,i,j, temp;
9      int arr[1005];
10     cin >> n;
11
12     for(i=0; i<n; i++)
13     {
14         cin >> arr[i];
15     }
16
17     for(i=1; i<n; i++)
18     {
19         temp = arr[i];
20         j = i-1;
21
22         while(arr[j]>temp && j>=0)
23         {
24             arr[j+1] = arr[j];
25             j--;
26         }
27         arr[j+1] = temp;
28     }
29
30     for(i=0; i<n;i++)
31     {
32         cout << arr[i] << " ";
33     }
34
35     return 0;
36 }
37 void tukar(int *a, int *b)
38 {
39     int t=*a;
40     *a=*b;
41     *b=t;
42 }
```

Gambar 5.4.a Source Code Percobaan II-4 Insertion Sort

Berdasarkan gambar 5.1.4.a yang merupakan source code dari percobaan II-4 Insertion sort, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library iostream ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar

untuk compiler. Kemudian pada baris ke 4 terdapat deklarasi void prototype function dengan nama tukar dengan parameter integer pointer a dan integer pointer b. Kemudian pada baris ke 6 terdapat deklarasi fungsi main dengan tipe data integer. Pada baris ke 8 terdapat deklarasi empat variabel integer dengan nama n, i, j, dan temp. Pada baris ke 9 terdapat deklarasi array integer dengan nama arr dan Panjang array yaitu 1005. Pada baris ke 10, terdapat perintah cin yang berfungsi agar user memasukkan nilai ke variabel n. Pada baris ke 12, terdapat perulangan for dengan banyak perulangan sebanyak nilai dari variabel n. Di dalam perulangan tersebut, terdapat perintah bagi user untuk memasukkan nilai ke array arr mulai dari indeks ke 0 hingga ke n. Setelah keluar dari perulangan tersebut, pada baris ke 17, terdapat perulangan for dengan banyak perulangan dimulai dari 1 hingga ke n. Di dalam perulangan tersebut, terdapat inisialisasi nilai temp dengan nilai array yang sedang dibaca dan nilai $j = i - 1$. Kemudian pada baris ke 22, masih di dalam perulangan tersebut, terdapat perulangan while dengan kondisi jika array $arr[j]$ lebih besar dari temp dan j lebih besar atau sama dengan 0, maka lakukan inisialisasi $arr[j+1] = arr[j]$ dan decrement j. Kemudian setelah keluar dari perulangan while, pada baris ke 27 terdapat inisialisasi nilai array $arr[j+1] = temp$. Kemudian keluar dari perulangan loop, pada baris ke 30, terdapat perulangan dengan banyak perulangan sebanyak n dan berfungsi untuk menampilkan nilai dari array arr mulai dari indeks ke 0 hingga indeks ke n. Kemudian pada baris ke 33 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal. Berikutnya pada baris ke 37 terdapat pendefinisian fungsi tukar dengan parameter variabel integer pointer a dan variabel integer pointer b. Di dalamnya, yaitu pada baris ke 39, terdapat inisialisasi dari variabel t dengan nilai yaitu nilai dari variabel t dikali dengan nilai dari variabel a. Kemudian pada baris ke 40, terdapat inisialisasi dari variabel pointer a dengan nilai yaitu nilai dari variabel pointer a dikali dengan nilai dari b. Kemudian pada baris ke 41 terdapat inisialisasi dari variabel pointer b dengan nilai dari variabel t

5.4.b Output



```
OVO\OneDrive\Kuliah\Semester 3\Prakt. Struktur
Data\P2\Code\" ; if ($?) { g++ insertionSort.
cpp -o insertionSort } ; if ($?) { .\insertion
Sort }
5
12
90
34
78
56
12 34 56 78 90
```

Gambar 5.4.b Output Percobaan II-4 Insertion Sort

Berdasarkan gambar 5.4.b yang merupakan output dari percobaan II-4 Insertion Sort, pertama program akan meminta user untuk memasukkan nilai. Berdasarkan source code yang ada, hal ini sesuai dengan perintah pada baris ke 10. Nilai yang dimasukkan user nantinya akan menjadi nilai dari variable *n* dan berfungsi untuk menentukan banyak data yang akan diurutkan. Kemudian berikutnya user diminta untuk memasukkan data yang akan diurutkan satu per satu. Hal ini sesuai dengan perintah pada baris ke 14 yang berada di dalam perulangan pada baris ke 12. Setelah semua data yang akan diurutkan telah selesai dimasukkan, kemudian program akan menampilkan data yang telah terurut. Hal ini sesuai dengan perintah pada baris ke 32 yang berada di dalam loop yang ada pada baris ke 30,

VI. KESIMPULAN

Adapun kesimpulan dari percobaan ini adalah sebagai berikut :

1. Berdasarkan percobaan 5.1 Exchange Sort Ascending dan 5.2 Exchange Sort Descending, dapat disimpulkan bahwa pada exchange sort ascending, terdapat satu data pivot yang dibandingkan dengan data lainnya. Jika data pivot tersebut lebih besar daripada data lainnya yang sedang dibandingkan, maka data akan ditukar posisinya. Hal ini sesuai dengan perintah pada baris ke 21 : `if(arr[i] > arr[j])`. Sementara itu pada exchange sort descending, jika data pivot tersebut lebih kecil daripada data lainnya yang sedang dibandingkan, maka data akan ditukar posisinya. Hal ini sesuai dengan perintah pada baris ke 21 : `if(arr[i] < arr[j])`
2. Berdasarkan percobaan 5.1 Exchange Sort Ascending, 5.2 Exchange Sort Descending, dan 5.3 Selection Sort, dapat disimpulkan bahwa pada program tersebut menggunakan void function tukar. Function ini digunakan untuk menukar nilai dari 2 parameter yang dimasukkan
3. Berdasarkan percobaan 5.4 Insertion Sort, dapat disimpulkan bahwa fungsi void tukar pada source code tidak digunakan di dalam proses yang akan dieksekusi oleh program ketika dijalankan karena program menggunakan temporary variabel sendiri dan tidak menggunakan variabel t pada fungsi tukar.
4. Berdasarkan percobaan 5.1 Exchange Sort Ascending, 5.2 Exchange Sort Descending, dan 5.3 Selection Sort, dapat disimpulkan bahwa pada program tersebut, pertukaran terjadi dengan memanfaatkan alamat address dan bukan menukar nilainya secara langsung.
5. Berdasarkan percobaan 5.1 Exchange Sort Ascending, 5.2 Exchange Sort Descending, dan 5.3 Selection Sort, dapat disimpulkan bahwa pada program tersebut, pertukaran terjadi dengan memanfaatkan variabel t untuk menyimpan data sementara. Pertama program memasukkan nilai variabel a ke dalam variabel t, lalu nilai variabel b dipindah ke dalam variabel a dan kemudian variabel b akan diisi nilai variabel a.

DAFTAR PUSTAKA

Erliansyah Nasution dan Indra Yatini B.2005. Algoritma dan Struktur Data. Graha Ilmu. Yogyakarta.

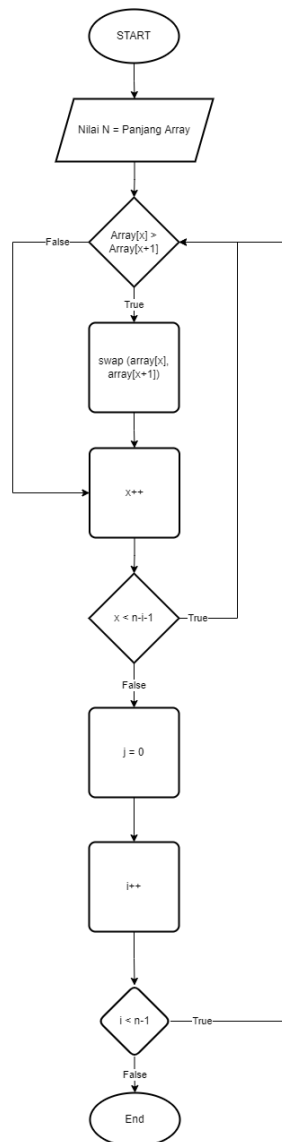
Sjukani, Moh. 2010. (Algoritma Dan Struktur Data I) Dengan C, C++ dan Java. Jakarta: Mitra Wacana Media

Muhammad, M.A. (018. Struktur Data. Modul Struktur Data Unila PSTI. 2 (2) : 1 - 2.

TUGAS AKHIR

1. Flowchart

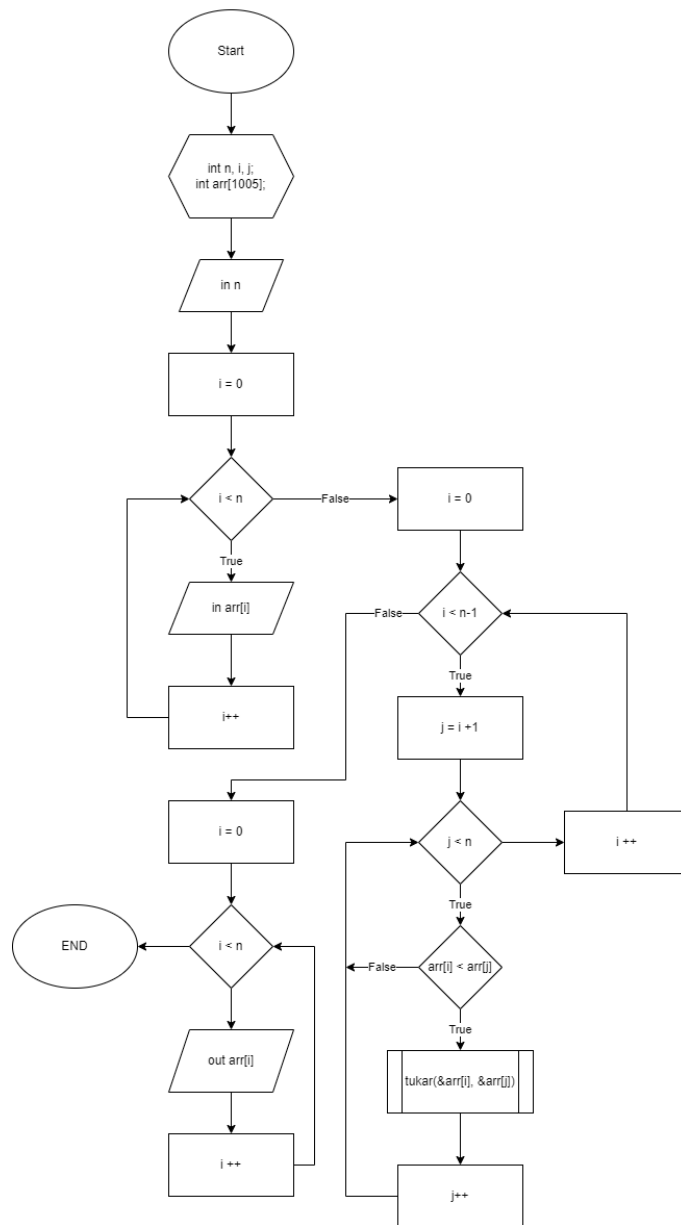
Adapun flowchart dari bubble sort adalah sebagai berikut :



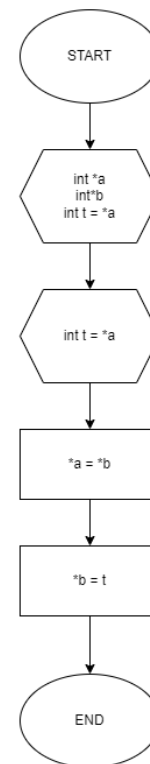
Gambar 1.1 flowchart bubble sort

Adapun flowchart dari exchange sort adalah sebagai berikut :

int main()

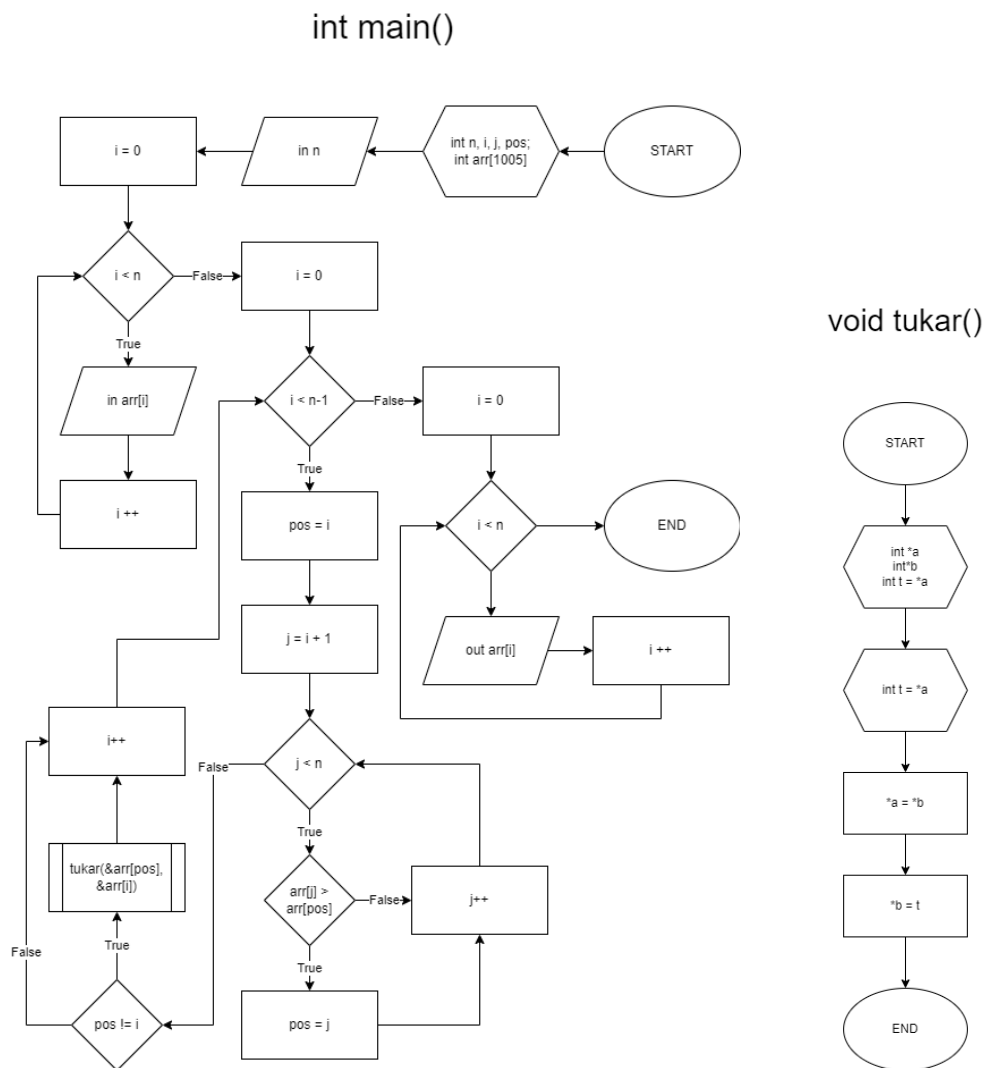


void tukar()



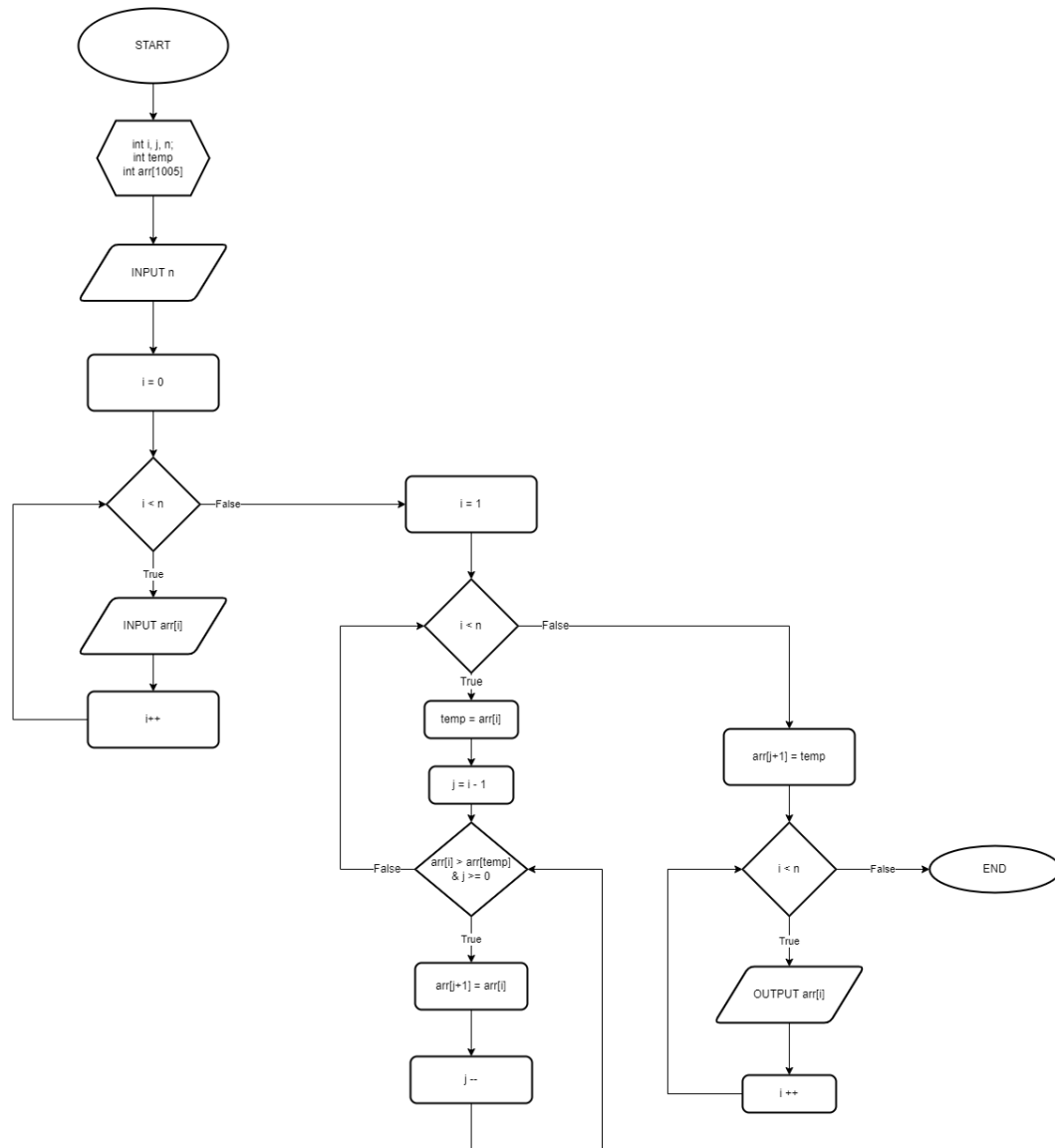
Gambar 1.2 flowchart exchange sort

Adapun flowchart dari selection sort adalah sebagai berikut :



Gambar 1.3 flowchart selection sort

Adapun flowchart dari insertion sort adalah sebagai berikut :



Gambar 1.4 flowchart insertion sort

2. Source code

Adapun source code dari program adalah sebagai berikut :

```
C++ bak.cpp
1  #include <iostream>
2  #include <stdlib.h>
3  using namespace std;
4
5  // function prototype
6  void swap(int *a, int *b);
7  void printArray(int arrayInput[], int arrayLength);
8  void bubbleSort(int arrayInput[], int arrayLength);
9  void exchangeSort(int arrayInput[], int arrayLength);
10 void insertionSort(int arrayInput[], int arrayLength);
11 void selectionSort(int arrayInput[], int arrayLength);
12 void duplicateArray(int arrayInput[], int temporaryArray[], int length);
13 void inputArrayElement(int arrayInput[], int arrayLength);
14 void menu();
15
16 // main function
17 int main() {
18
19     menu(); // eksekusi fungsi menu
20
21     return 0;
22 }
23
24 void menu(){
25     string choice;
26     int arrayInput[1000];
27     int arrayLength;
28     int randNum;
29     bool ongoing = true;
30     bool needChoiceInput = false;
31
32     cout << endl << "Program Penerapan Metode Sorting." << endl;
33
34     // membuat array
35     cout << "Ukuran array: ";
36     cin >> arrayLength; // input ukuran array
37     arrayInput[arrayLength];
38
39     inputArrayElement(arrayInput, arrayLength);
40
41     cout << endl << "Kondisi awal array:" << endl;
42     printArray(arrayInput, arrayLength); // cetak array
43
44     while(ongoing){
45         needChoiceInput = true;
46         cout << endl << "===== " << endl;
47         cout << "Program Penerapan Metode Sorting" << endl;
48         cout << "===== " << endl;
49         cout << "1. Bubble Sort" << endl;
50         cout << "2. Exchange Sort" << endl;
```

Gambar 2.1 source code

```

51     cout << "3. Selection Sort" << endl;
52     cout << "4. Insertion Sort" << endl;
53     cout << "5. Input ulang Array" << endl;
54     cout << "6. Test dengan nilai random" << endl;
55     cout << "7. Test dengan array kosong" << endl;
56     cout << "8. AKHIRI PROGRAM" << endl;
57     // memilih metode sorting
58     while (needChoiceInput){
59         cout << "Silahkan pilih antara [1-8]: ";
60         cin >> choice;
61         needChoiceInput = false;
62         switch (choice[0]){
63             case '1':
64                 bubbleSort(arrayInput, arrayLength);
65                 break;
66             case '2':
67                 exchangeSort(arrayInput, arrayLength);
68                 break;
69             case '3':
70                 selectionSort(arrayInput, arrayLength);
71                 break;
72             case '4':
73                 insertionSort(arrayInput, arrayLength);
74                 break;
75             case '5': // apabila user ingin membuat array baru
76                 cout << "Ukuran array: ";
77                 cin >> arrayLength; // input ukuran array
78                 arrayInput[arrayLength];
79                 inputArrayElement(arrayInput, arrayLength);
80                 break;
81             case '6':
82                 cout << endl << "#06. Array dengan Nilai Random" << endl;
83                 cout << "Ukuran array: ";
84                 cin >> arrayLength;
85                 for(int i = 0; i < arrayLength; i++){
86                     randNum = rand() % 100 + 1;
87                     arrayInput[i] = randNum;
88                 }
89                 cout << "Kondisi array sekarang:" << endl;
90                 printArray(arrayInput, arrayLength);
91                 break;
92             case '7':
93                 cout << endl << "#07. Array Kosong" << endl;
94                 cout << "Ukuran array: ";
95                 cin >> arrayLength;
96                 cout << "Array kosong berhasil dibuat." << endl;
97                 break;
98             case '8':
99                 cout << endl << "Program Selesai." << endl;
100                ongoing = false;

```

Gambar 2.2 source code

```

101         break;
102     default:
103         // jika input tidak sesuai akan diminta kembali
104         cout << "Input tidak tepat." << endl << endl;
105         needChoiceInput = true;
106         break;
107     }
108 }
109 }
110 // akhir fungsi menu
111
112 // fungsi untuk input array
113 void inputArrayElement(int arrayInput[], int arrayLength){
114     for(int i = 0; i < arrayLength; i++) {
115         cout << "elemen-" << i << ": ";
116         cin >> arrayInput[i]; // input elemen array
117     }
118 }
119
120 // fungsi untuk mentukar elemen array
121 void swap(int *a, int *b){
122     int t=*a;
123     *a=*b;
124     *b=t;
125 }
126
127 // fungsi untuk mencetak seluruh elemen array
128 void printArray(int arrayInput[], int arrayLength){
129     cout << " ";
130     for(int i = 0; i < arrayLength; i++) {
131         cout << arrayInput[i] << " ";
132     } cout << endl;
133 }
134
135 // fungsi untuk duplikasi array input ke array sementara
136 void duplicateArray(int arrayInput[], int temporaryArray[], int length){
137     for (int i = 0; i < length; i++){
138         temporaryArray[i] = arrayInput[i];
139     }
140 }
141
142 // fungsi bubble sort
143 void bubbleSort(int arrayInput[], int arrayLength){
144     int temporaryArray[arrayLength];
145     int stepCounter = 0, swapCounter = 0;
146     duplicateArray(arrayInput, temporaryArray, arrayLength);
147
148     cout << endl << "#01 Metode Bubble Sort" << endl;
149     cout << "Kondisi awal array:" << endl;
150     printArray(arrayInput, arrayLength);

```

Gambar 2.3 source code

```

151     cout << "Proses Sorting" << endl;
152
153     for (int i = 0; i < arrayLength-1; i++) {
154         for (int j = 0; j < arrayLength-i-1; j++) {
155             // perbandingan elemen
156             if (temporaryArray[j] > temporaryArray[j+1]) {
157                 swap(&temporaryArray[j], &temporaryArray[j+1]);
158                 swapCounter++; // perhitungan jumlah pertukaran
159                 printArray(temporaryArray, arrayLength);
160             }
161             stepCounter++; // perhitungan jumlah langkah
162         }
163     }
164     // mencetak array hasil sorting, beserta jumlah langkah dan pertukaran
165     cout << "Hasil Sorting Bubble Sort: " << endl;
166     printArray(temporaryArray, arrayLength);
167     cout << "Total langkah Bubble Sort: " << stepCounter << endl;
168     cout << "Total pertukaran Bubble Sort: " << swapCounter << endl;
169 }
170
171 // fungsi exchange sort
172 void exchangeSort(int arrayInput[], int arrayLength){
173     int temporaryArray[arrayLength];
174     int stepCounter = 0, swapCounter = 0;
175     duplicateArray(arrayInput, temporaryArray, arrayLength);
176
177     cout << endl << "#02 Metode Exchange Sort" << endl;
178     cout << "Kondisi awal array:" << endl;
179     printArray(arrayInput, arrayLength);
180     cout << "Proses Sorting" << endl;
181     for (int i = 0; i < arrayLength; i++) {
182         for (int j = i + 1; j < arrayLength; j++) {
183             if (temporaryArray[i] < temporaryArray[j]) {
184                 swap(&temporaryArray[i], &temporaryArray[j]);
185                 printArray(temporaryArray, arrayLength);
186                 swapCounter++;
187             }
188             stepCounter++;
189         }
190     }
191     cout << endl << "Hasil Sorting Exchange Sort: " << endl;
192     printArray(temporaryArray, arrayLength);
193     cout << endl << "Total langkah Exchange Sort: " << stepCounter << endl;
194     cout << "Total pertukaran Exchange Sort: " << swapCounter << endl;
195 }
196
197 // fungsi selection sort
198 void selectionSort(int arrayInput[], int arrayLength) {
199     int temporaryArray[arrayLength];
200     int pos, stepCounter = 0, swapCounter = 0;

```

Gambar 2.4 source code

```

201     duplicateArray(arrayInput, temporaryArray, arrayLength);
202
203     cout << endl << "#03 Metode Selection Sort" << endl;
204     cout << "Kondisi awal array:" << endl;
205     printArray(arrayInput, arrayLength);
206     cout << "Proses Sorting" << endl;
207     for (int i = 0; i < arrayLength - 1; i++) {
208         pos = i;
209         for (int j = i + 1; j < arrayLength; j++) {
210             if (temporaryArray[j] < temporaryArray[pos]) {
211                 pos = j;
212             }
213             stepCounter++;
214         }
215         if (pos != i) {
216             swap(&temporaryArray[pos], &temporaryArray[i]);
217             swapCounter++;
218             printArray(temporaryArray, arrayLength);
219         }
220     }
221
222     cout << "Hasil Sorting Selection Sort: " << endl;
223     printArray(temporaryArray, arrayLength);
224     cout << endl << "Total langkah Selection Sort: " << stepCounter << endl;
225     cout << "Total pertukaran Selection Sort: " << swapCounter << endl;
226 }
227
228 // fungsi insertion sort
229 void insertionSort(int arrayInput[], int arrayLength){
230     int temporaryArray[arrayLength];
231     int j, temp, stepCounter = 0, swapCounter = 0;
232     duplicateArray(arrayInput, temporaryArray, arrayLength);
233
234     cout << endl << "#04 Metode Insertion Sort" << endl;
235     cout << "Kondisi awal array:" << endl;
236     printArray(arrayInput, arrayLength);
237     cout << "Proses Sorting" << endl;
238     for (int i = 1; i < arrayLength; i++) {
239         temp = temporaryArray[i];
240         j = i - 1;
241         while (temporaryArray[j] > temp && j >= 0) {
242             temporaryArray[j+1] = temporaryArray[j];
243             j--;
244             printArray(temporaryArray, arrayLength);
245             stepCounter++;
246             swapCounter++;
247         }
248         temporaryArray[j+1] = temp;
249     }
250
251     cout << "Hasil Sorting Insertion Sort: " << endl;
252     printArray(temporaryArray, arrayLength);
253     cout << endl << "Total langkah Insertion Sort: " << stepCounter << endl;
254     cout << "Total pertukaran Insertion Sort: " << swapCounter << endl;
255 }

```

Gambar 2.5 source code

Berdasarkan gambar 2.1, 2.2, 2.3, 2.4, dan 2.5, dapat dilihat bahwa pada baris pertama terdapat penggunaan library iostream agar dapat terjadi input dan output pada program. Kemudian pada baris kedua terdapat penggunaan library stdlib.h.

Pada baris ketiga, terdapat using namespace std agar program menggunakan penamaan standar. Kemudian pada baris ke 6 terdapat void function dengan nama swap dan parameter integer pointer a dan integer pointer b. Pada baris ke 7 terdapat prototype void function bernama printArray dengan parameter int arrayInput[] dan int arrayLength. Pada baris ke 8 terdapat prototype void function bernama bubbleSort dengan parameter int arrayInput[] dan int arrayLength. Pada baris ke 9 terdapat prototype void function bernama exchangeSort dengan parameter int arrayInput[] dan int arrayLength. Pada baris ke 10 terdapat prototype void function bernama insertionSort dengan parameter int arrayInput[] dan int arrayLength. Pada baris ke 11 terdapat prototype void function bernama selectionSort dengan parameter int arrayInput[] dan int arrayLength. Pada baris ke 12 terdapat prototype void function bernama duplicateArray dengan parameter int arrayInput[], int temporaryArray[], dan length. Pada baris ke 13 terdapat prototype void function bernama inputArrayElement dengan parameter int arrayInput[] dan int arrayLength. Pada baris ke 14 terdapat prototype void function bernama menu. Pada baris ke 17, terdapat deklarasi fungsi utama int main(). Di dalamnya, yaitu pada baris ke 19, terdapat perintah untuk menjalankan fungsi menu. Pada baris ke 21 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal. Pada baris ke 24 terdapat pendefinisian dari void function menu. Pada baris ke 25 terdapat variabel string dengan nama choice. Pada baris ke 26 terdapat array dengan nama arrayInput, bertipe data integer, dan Panjang array 1000. Pada baris ke 27 terdapat variabel integer arrayLength. Pada baris ke 28 terdapat variabel integer dengan nama randNum. Pada baris ke 29 terdapat variabel boolean dengan nama ongoing dan nilai true. Pada baris ke 30 terdapat variabel boolean needChoiceInput dengan nilai false. Pada baris ke 32 terdapat perintah untuk menampilkan pesan "Program Penerapan Metode Sorting." Pada baris ke 35 terdapat perintah untuk menampilkan pesan "Ukuran array: ". Pada baris ke 36 terdapat perintah bagi user untuk memasukkan nilai ke variabel arrayLength. Pada baris ke 37 terdapat perintah untuk memasukkan nilai dari arrayLength ke variabel arrayInput. Pada baris ke 39 terdapat perintah untuk menjalankan fungsi inputArrayElement dengan argument arrayInput dan arrayLength. Pada baris ke 41 terdapat perintah untuk menampilkan pesan "Kondisi awal array: ". Pada baris ke

42 terdapat perintah untuk menjalankan fungsi `printArray` dengan argument `arrayInput` dan `arrayLength`. Pada baris ke 44 terdapat looping `while` dengan kondisi `ongoing` hingga baris ke 109. Pada baris ke 45, di dalam looping `while`, terdapat inisialisasi nilai `needChoiceInput` dengan nilai `true`. Pada baris ke 46 terdapat perintah untuk menampilkan pesan `"=====`". Pada baris ke 47 terdapat perintah untuk menampilkan pesan "Program Penerapan Metode Sorting". Pada baris ke 48 terdapat perintah untuk menampilkan pesan `"=====`". Pada baris ke 49 terdapat perintah untuk menampilkan pesan "1. Bubble Sort". Pada baris ke 50 terdapat perintah untuk menampilkan pesan "2. Exchange Sort". Pada baris ke 51 terdapat perintah untuk menampilkan pesan "3. Selection Sort". Pada baris ke 52 terdapat perintah untuk menampilkan pesan "4. Insertion Sort". Pada baris ke 53 terdapat perintah untuk menampilkan pesan "5. Input ulang Array". Pada baris ke 54 terdapat perintah untuk menampilkan pesan "6. Test dengan nilai random". Pada baris ke 55 terdapat perintah untuk menampilkan pesan "7. Test dengan array kosong". Pada baris ke 56 terdapat perintah untuk menampilkan pesan "8. AKHIRI PROGRAM". Pada baris ke 58 terdapat perulangan `while` dengan kondisi `needChoiceInput` hingga baris ke 108. Pada baris 59 di dalam perulangan tersebut, terdapat perintah untuk menampilkan pesan "Silahkan pilih anantara [1-8]: ". Pada baris ke 60 terdapat perintah untuk user memasukkan nilai ke variabel `choice`. Pada baris ke 61, terdapat inisialisasi nilai `needChoiceInput` menjadi `false`. Pada baris ke 62 terdapat percabangan `if` dengan kondisi `choice[0]`. Pada baris ke 63 terdapat case 1, dimana isinya terdapat pada baris ke 64 yaitu menjalankan fungsi `bubbleSort` dengan argument `arrayInput` dan `arrayLength` serta perintah untuk `break` dari `switch` pada baris ke 65. Pada baris ke 66 terdapat case 2, dimana isinya terdapat pada baris ke 67 yaitu menjalankan fungsi `exchangeSort` dengan argumen `arrayInput` dan `arrayLength` serta perintah untuk `break` dari `switch` pada baris ke 68. Pada baris ke 69 terdapat case 3, dimana isinya terdapat pada baris ke 70 yaitu menjalankan fungsi `selectionSort` dengan argument `arrayInput` dan `arrayLength` serta perintah untuk `break` dari `switch` pada baris ke 71. Pada baris ke 72 terdapat case 4, dimana isinya terdapat pada baris ke 73 yaitu menjalankan fungsi `insertionSort` dengan argument

arrayInput dan arrayLength serta perintah untuk break dari switch pada baris ke 74. Pada baris ke 75 terdapat case 5, dimana isinya terdapat pada baris ke 76 yaitu perintah untuk menampilkan pesan "Ukuran array: ", pada baris ke 77 yaitu perintah untuk user agar memasukkan nilai ke arrayLength, pada baris ke 78 yaitu inisialisasi Panjang array arrayInput dengan arrayLength, pada baris ke 79 yaitu perintah untuk menjalankan fungsi inputArrayElement dengan argument arrayInput dan arrayLength, dan break dari switch pada baris ke 80. Pada baris ke 81 terdapat case 6 yang berisi perintah untuk menampilkan pesan "#06. Array dengan Nilai Random" pada baris ke 82, perintah untuk menampilkan pesan "Ukuran array: " pada baris ke 83, perintah agar user memasukkan nilai ke variabel arrayLength pada baris ke 84, perulangan for dengan kondisi $i=0$, $i < \text{arrayLength}$, $i++$ pada baris ke 85, perintah untuk menampilkan pesan "Kondisi array sekarang" pada baris ke 89, perintah untuk menjalankan fungsi printArray pada baris ke 90, dan break dari switch pada baris ke 91. Pada perulangan di baris ke 85, terdapat perintah untuk menginisialisasi nilai dari variabel randNum dengan nilai dari fungsi $\text{rand} \% 100 + 1$. Kemudian pada baris ke 87 terdapat inisialisasi nilai dari arrayInput indeks ke i dengan nilai dari randNum. Pada baris ke 92 terdapat case ke 7 yang berisi perintah untuk menampilkan pesan "#07. Array Kosong" pada baris ke 93, perintah untuk menampilkan pesan "Ukuran array: " pada baris ke 94, perintah untuk agar user memasukkan nilai ke variabel arrayLength pada baris ke 95, perintah untuk menampilkan pesan "Array kosong berhasil dibuat" pada baris ke 96, dan break dari case pada baris ke 97. Pada baris ke 98 terdapat case 8 yang berisi perintah untuk menampilkan pesan "Program selesai" pada baris ke 99, inisialisasi variabel ongoing dengan nilai false pada baris ke 100, dan break dari switch pada baris ke 101. Pada baris ke 102, terdapat default dari switch yang berisi perintah untuk menampilkan pesan "Input tidak tepat" pada baris ke 104, inisialisasi nilai needChoiceInput menjadi true pada baris ke 105 dan break dari switch pada baris ke 106. Pada baris ke 113 terdapat deklarasi void function inputArrayElement dengan parameter int arrayInput[] dan int arrayLength. Pada baris ke 114 terdapat perulangan for dengan kondisi $i = 0$ dan jika $i < \text{arrayLength}$ maka tampilkan pesan "elemen-", nilai dari i , dan ": " pada baris ke 115 dan perintah agar user memasukkan nilai ke array input dimulai dari indeks ke 0 hingga ke i pada baris ke

116. Pada baris ke 121 terdapat deklarasi void function swap dengan parameter variabel integer pointer a dan variabel integer pointer b. Di dalamnya, yaitu pada baris ke 122 terdapat inisialisasi variabel t dengan nilai dari pointer a. Pada baris ke 123 terdapat inisialisasi nilai dari pointer a dengan nilai dari pointer b. Pada baris ke 124 terdapat inisialisasi nilai dari pointer b dengan nilai dari variabel t. Pada baris ke 128 terdapat deklarasi fungsi printArray dengan parameter int arrayInput[] dan int arrayLength. Di dalamnya, terdapat perintah untuk menampilkan “ “ pada baris ke 129, perulangan for pada baris ke 130 dengan kondisi int i = 0, dan jika i < arrayLength maka lakukan tampilkan nilai dari arrayInput[i] pada baris ke 131, dan perintah untuk menampilkan baris baru pada baris ke 132. Pada baris ke 136 terdapat deklarasi void function duplicateArray dengan parameter arrayInput[i], temporaryArray[], dan int length. Di dalamnya, terdapat perulangan for pada baris ke 137 dengan kondisi int i = 0 dan jika i < length maka lakukan inisialisasi nilai temporaryArray[i] dengan nilai dari arrayInput[i] pada baris ke 138. Pada baris ke 143 terdapat deklarasi void function bubbleSort dengan parameter int arrayInput[] dan int arrayLength. Di dalamnya, terdapat inisialisasi array temporaryArray dengan Panjang array sama dengan nilai dari arrayLength pada baris ke 144, inisialisasi dua variabel integer dengan nama stepCounter dengan nilai 0 dan swapCounter dengan nilai 0 pada baris ke 145, perintah untuk menjalankan fungsi duplicateArray dengan argument arrayInput, temporaryArray, dan arrayLength pada baris ke 146. Kemudian di dalamnya juga terdapat perintah untuk menampilkan pesan "#01 Metode Bubble Sort" pada baris ke 148. Perintah untuk menampilkan pesan “Kondisi awal array: “ pada baris ke 149, perintah untuk menjalankan fungsi printArray dengan argument arrayInput, dan arrayLength pada baris ke 150, perintah untuk menampilkan pesan “Proses sorting” pada baris ke 151, dan perulangan for pada baris ke 153 dengan kondisi int i = 0 dan jika i < arrayLength. Di dalam perulangan tersebut terdapat perulangan for kembali (nested loop), tepatnya pada baris ke 154 dengan kondisi j = 0 dan jika j < arrayLength, maka lakukan percabangan if pada baris ke 156 dengan kondisi jika temporaryArray[j] > temporaryArray[j+1] dan increment stepCounter pada baris ke 161. Di dalam percabangan tersebut terdapat perintah untuk menjalankan fungsi swap dengan argument &temporaryArray[j], &temporaryArray[j+1] pada baris ke

157, increment nilai swapCounter pada baris ke 158, dan perintah untuk menjalankan fungsi printArray dengan argument temporaryArray dan arrayLength pada baris ke 166. Pada baris ke 165 terdapat perintah untuk menampilkan pesan “Hasil Sorting Bubble Sort: “. Pada baris ke 166 terdapat perintah untuk menjalankan fungsi printArray dengan argument temporaryArray dan arrayLength. Pada baris ke 167 terdapat perintah untuk menampilkan pesan “Total Langkah bubble sort: “ dan nilai dari stepCounter. Pada baris ke 168 terdapat perintah untuk menampilkan pesan “Total pertukaran bubble sort: “ dan nilai dari swapCounter. Pada baris ke 172 terdapat deklarasi fungsi exchangeSort dengan parameter int arrayInput[] dan int arrayLength. Di dalamnya, terdapat deklarasi array temporaryArray dengan Panjang array sama dengan nilai dari arrayLength pada baris ke 173, inisialisasi variabel integer stepCounter dengan nilai 0 dan variabel integer swapCounter dengan nilai 0 pada baris ke 174, perintah untuk menjalankan fungsi duplicateArray dengan argument arrayInput, temporaryArray, dan arrayLength pada baris ke 175. Selain itu terdapat perintah untuk menampilkan pesan “ #02 Metode Exchange Sort” pada baris ke 177, perintah untuk menampilkan pesan “Kondisi awal array: “ pada baris ke 178, perintah untuk menjalankan fungsi printArray dengan argument arrayInput, dan arrayLength pada baris ke 179, perintah untuk menampilkan pesan “Proses sorting” pada baris ke 180, dan perulangan for pada baris ke 181 dengan kondisi int i = 0 dan jika i < arrayLength. Di dalam perulangan tersebut terdapat perulangan for kembali (nested loop), tepatnya pada baris ke 182 dengan kondisi j = i dan jika j < arrayLength, maka lakukan percabangan if pada baris ke 183 dengan kondisi jika temporaryArray[i] < temporaryArray[j] dan increment stepCounter pada baris ke 188. Di dalam percabangan tersebut terdapat perintah untuk menjalankan fungsi swap dengan argument &temporaryArray[i], &temporaryArray[j] pada baris ke 184, perintah untuk menjalankan fungsi printArray dengan argument temporaryArray dan arrayLength, pada baris ke 185 dan increment nilai swapCounter pada baris ke 186. Pada baris ke 191 terdapat perintah untuk menampilkan pesan “Hasil Sorting Exchange Sort: “. Pada baris ke 192 terdapat perintah untuk menjalankan fungsi printArray dengan argument temporaryArray dan arrayLength. Pada baris ke 193 terdapat perintah untuk menampilkan pesan “Total Langkah Exchange Sort: “ dan nilai dari stepCounter.

Pada baris ke 194 terdapat perintah untuk menampilkan pesan “Total pertukaran Exchange Sort: “ dan nilai dari swapCounter. Pada baris ke 198 terdapat deklarasi fungsi selectionSort dengan parameter int arrayInput[] dan int arrayLength. Di dalamnya, terdapat deklarasi array temporaryArray dengan Panjang array sama dengan nilai dari arrayLength pada baris ke 199, inialisasi variabel integer pos, variabel integer stepCounter dengan nilai 0 dan variabel integer swapCounter dengan nilai 0 pada baris ke 200, perintah untuk menjalankan fungsi duplicateArray dengan argument arrayInput, temporaryArray, dan arrayLength pada baris ke 201. Selain itu terdapat perintah untuk menampilkan pesan “#03 Metode Selection Sort” pada baris ke 203, perintah untuk menampilkan pesan “Kondisi awal array: “ pada baris ke 204, perintah untuk menjalankan fungsi printArray dengan argument arrayInput, dan arrayLength pada baris ke 205, perintah untuk menampilkan pesan “Proses sorting” pada baris ke 206, dan perulangan for pada baris ke 207 dengan kondisi int i = 0 dan jika i < arrayLength. Di dalam perulangan tersebut terdapat inialisasi nilai pos = i pada baris ke 208 dan perulangan for kembali (nested loop), tepatnya pada baris ke 209 dengan kondisi j = i + 1 dan jika j < arrayLength, maka lakukan percabangan if pada baris ke 210 dengan kondisi jika temporaryArray[j] > temporaryArray[pos] dan increment stepCounter pada baris ke 213. Di dalam percabangan tersebut terdapat inialisasi nilai pos = j; Pada baris ke 215 terdapat percabangan if dengan kondisi pos != i. Di dalam percabangan tersebut terdapat perintah untuk menjalankan fungsi swap dengan argument &temporaryArray[pos], &temporaryArray[i] pada baris ke 216, increment nilai swapCounter pada baris ke 217, dan perintah untuk menjalankan fungsi printArray dengan argument temporaryArray dan arrayLength, pada baris ke 218. Pada baris ke 222 terdapat perintah untuk menampilkan pesan “Hasil Sorting Selection Sort: “. Pada baris ke 223 terdapat perintah untuk menjalankan fungsi printArray dengan argument temporaryArray dan arrayLength. Pada baris ke 224 terdapat perintah untuk menampilkan pesan “Total Langkah Selection Sort: “ dan nilai dari stepCounter. Pada baris ke 225 terdapat perintah untuk menampilkan pesan “Total pertukaran Selection Sort: “ dan nilai dari swapCounter. Pada baris ke 229 terdapat deklarasi fungsi insertionSort dengan parameter int arrayInput[] dan int arrayLength. Di dalamnya, terdapat deklarasi array temporaryArray dengan

Panjang array sama dengan nilai dari `arrayLength` pada baris ke 230, deklarasi variabel integer `j`, `temp`, dan inisialisasi variabel integer `stepCounter` dengan nilai 0 dan variabel integer `swapCounter` dengan nilai 0 pada baris ke 231, perintah untuk menjalankan fungsi `duplicateArray` dengan argument `arrayInput`, `temporaryArray`, dan `arrayLength` pada baris ke 232. Selain itu terdapat perintah untuk menampilkan pesan “ #04 Metode Insertion Sort” pada baris ke 234, perintah untuk menampilkan pesan “Kondisi awal array: “ pada baris ke 235, perintah untuk menjalankan fungsi `printArray` dengan argument `arrayInput`, dan `arrayLength` pada baris ke 236, perintah untuk menampilkan pesan “Proses sorting” pada baris ke 237, dan perulangan `for` pada baris ke 238 dengan kondisi `int i = 1` dan jika `i < arrayLength`. Di dalam perulangan tersebut terdapat inisialisasi nilai `temp = temporaryArray[i]` pada baris ke 239, inisialisasi nilai `j = i - 1` pada baris ke 240, dan perulangan `while` (nested loop), tepatnya pada baris ke 241 dengan kondisi `temporaryArray[j] < temp` dan `j >= 0`, maka lakukan inisialisasi nilai `temporaryArray[j+1]` dengan nilai dari `temporaryArray[j]` pada baris ke 242, decrement `j` pada baris ke 243, perintah untuk menjalankan fungsi `printArray` dengan argument `temporaryArray` dan `arrayLength`, pada baris ke 244, increment `stepCounter` pada baris ke 245 dan increment `swapCounter` pada baris ke 246. Setelah perulangan `while` selesai, pada baris ke 248 terdapat inisialisasi nilai dari `temporaryArray[j+1] = temp`. Pada baris ke 251 terdapat perintah untuk menampilkan pesan “Hasil Sorting Insertion Sort: “. Pada baris ke 252 terdapat perintah untuk menjalankan fungsi `printArray` dengan argument `temporaryArray` dan `arrayLength`. Pada baris ke 253 terdapat perintah untuk menampilkan pesan “Total Langkah Selection Sort: “ dan nilai dari `stepCounter`. Pada baris ke 254 terdapat perintah untuk menampilkan pesan “Total pertukaran Selection Sort: “ dan nilai dari `swapCounter`.

3. Output

Adapun output dari program adalah sebagai berikut :

```
3\Prakt. Struktur Data\P2\Code\" ; if ($?) { g++ bak.cpp -o bak } ; if ($?) { .\bak }

Program Penerapan Metode Sorting.
Ukuran array: 5
elemen-0: 12
elemen-1: 90
elemen-2: 34
elemen-3: 78
elemen-4: 56

Kondisi awal array:
  12 90 34 78 56

=====
Program Penerapan Metode Sorting
=====
1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 1

#01 Metode Bubble Sort
Kondisi awal array:
  12 90 34 78 56
Proses Sorting
  12 34 90 78 56
  12 34 78 90 56
  12 34 78 56 90
  12 34 56 78 90
Hasil Sorting Bubble Sort:
  12 34 56 78 90
Total langkah Bubble Sort: 10
Total pertukaran Bubble Sort: 4

=====
Program Penerapan Metode Sorting
=====
1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 8

Program Selesai.
```

Gambar 3.1 Output bubble sort

Berdasarkan gambar 3.1, dapat dilihat bahwa pertama program menampilkan pesan “Program penerapan metode sorting” yang sesuai dengan perintah pada baris ke 32.

Kemudian program menampilkan pesan “Ukuran array: “ yang sesuai dengan perintah pada baris ke 35. Kemudian user diminta untuk memasukkan nilai yang nantinya akan menjadi Panjang dari array. Pada percobaan ini saya memasukkan 5 nilai. Kemudian program akan menampilkan pesan “elemen-0: “ yang sesuai dengan perintah pada baris ke 115. Kemudian user diminta untuk memasukkan nilai dari tiap elemen yang akan diurutkan. Hal ini sesuai dengan perintah pada baris ke 116. Pada percobaan ini, saya memasukkan nilai 12, 90, 34, 78, dan 56. Kemudian program menampilkan pesan “Kondisi awal array” dan menampilkan kondisi awal dari array yang belum diurutkan yang sesuai dengan perintah pada baris ke 41 dan 42. Kemudian program menampilkan 8 menu, menu 1 untuk bubble sort, menu 2 untuk exchange sort, menu 3 untuk selection sort, menu 4 untuk insertion sort, menu 5 untuk memasukkan ulang array, menu 6 untuk melakukan sorting dengan nilai random, menu 7 untuk melakukan sorting tanpa data, dan menu 8 untuk mengakhiri program. Karena akan melakukan pengurutan dengan bubble sort, maka dipilih menu 1. Kemudian program menampilkan pesan “#01 Metode Bubble Sort” dan menampilkan kondisi array awal seperti sebelumnya. Kemudian, program melakukan proses sorting dan menampilkan data yang ada. Program menampilkan hasil sorting dengan metode terpilih, yaitu 12, 34, 56, 78, dan 90, sesuai dengan perintah pada baris ke 166. Kemudian program menampilkan total Langkah yang dilakukan, sesuai dengan perintah pada baris ke 167 dan total pertukaran yang dilakukan, sesuai dengan perintah pada baris ke 168. Kemudian program menampilkan kembali menu awal, dan dipilih menu 8 akhiri program.

```

3\Prakt. Struktur Data\P2\Code\" ; if ($?) { g++ bak.cpp -o bak } ; if ($?) { .\bak }

Program Penerapan Metode Sorting.
Ukuran array: 5
elemen-0: 12
elemen-1: 90
elemen-2: 34
elemen-3: 78
elemen-4: 56

Kondisi awal array:
  12 90 34 78 56

=====
Program Penerapan Metode Sorting
=====
1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 2

#02 Metode Exchange Sort
Kondisi awal array:
  12 90 34 78 56
Proses Sorting
  90 12 34 78 56
  90 34 12 78 56
  90 78 12 34 56
  90 78 34 12 56
  90 78 56 12 34
  90 78 56 34 12

Hasil Sorting Exchange Sort:
  90 78 56 34 12

Total langkah Exchange Sort: 10
Total pertukaran Exchange Sort: 6

=====
Program Penerapan Metode Sorting
=====
1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 8

Program Selesai.

```

Gambar 3.2 Output Exchange sort

Berdasarkan gambar 3.2, dapat dilihat bahwa pertama program menampilkan pesan “Program penerapan metode sorting” yang sesuai dengan perintah pada baris ke 32. Kemudian program menampilkan pesan “Ukuran array: “ yang sesuai dengan

perintah pada baris ke 35. Kemudian user diminta untuk memasukkan nilai yang nantinya akan menjadi Panjang dari array. Pada percobaan ini saya memasukkan 5 nilai. Kemudian program akan menampilkan pesan “elemen-0: “ yang sesuai dengan perintah pada baris ke 115. Kemudian user diminta untuk memasukkan nilai dari tiap elemen yang akan diurutkan. Hal ini sesuai dengan perintah pada baris ke 116. Pada percobaan ini, saya memasukkan nilai 12, 90, 34, 78, dan 56. Kemudian program menampilkan pesan “Kondisi awal array” dan menampilkan kondisi awal dari array yang belum diurutkan yang sesuai dengan perintah pada baris ke 41 dan 42. Kemudian program menampilkan 8 menu, menu 1 untuk bubble sort, menu 2 untuk exchange sort, menu 3 untuk selection sort, menu 4 untuk insertion sort, menu 5 untuk memasukkan ulang array, menu 6 untuk melakukan sorting dengan nilai random, menu 7 untuk melakukan sorting tanpa data, dan menu 8 untuk mengakhiri program. Karena akan melakukan pengurutan dengan exchange sort, maka dipilih menu 2. Kemudian program menampilkan pesan “#02 Metode Exchange Sort” dan menampilkan kondisi array awal seperti sebelumnya. Kemudian, program melakukan proses sorting dan menampilkan data yang ada. Program menampilkan hasil sorting dengan metode terpilih, yaitu 12, 34, 56, 78, dan 90, sesuai dengan perintah pada baris ke 192. Kemudian program menampilkan total Langkah yang dilakukan, sesuai dengan perintah pada baris ke 193 dan total pertukaran yang dilakukan, sesuai dengan perintah pada baris ke 193. Kemudian program menampilkan kembali menu awal, dan dipilih menu 8 akhiri program.

```

3\Prakt. Struktur Data\P2\Code\" ; if ($?) { g++ bak.cpp -o bak } ; if ($?) { .\bak }

Program Penerapan Metode Sorting.
Ukuran array: 5
elemen-0: 12
elemen-1: 90
elemen-2: 34
elemen-3: 78
elemen-4: 56

Kondisi awal array:
12 90 34 78 56

=====
Program Penerapan Metode Sorting
=====
1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 3

#03 Metode Selection Sort
Kondisi awal array:
12 90 34 78 56
Proses Sorting
90 12 34 78 56
90 78 34 12 56
90 78 56 12 34
90 78 56 34 12
Hasil Sorting Selection Sort:
90 78 56 34 12

Total langkah Selection Sort: 10
Total pertukaran Selection Sort: 4

=====
Program Penerapan Metode Sorting
=====
1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 8

Program Selesai.

```

Gambar 3.3 Output Selection sort

Berdasarkan gambar 3.3, dapat dilihat bahwa pertama program menampilkan pesan “Program penerapan metode sorting” yang sesuai dengan perintah pada baris ke 32. Kemudian program menampilkan pesan “Ukuran array: “ yang sesuai dengan perintah pada baris ke 35. Kemudian user diminta untuk memasukkan nilai yang

nantinya akan menjadi Panjang dari array. Pada percobaan ini saya memasukkan 5 nilai. Kemudian program akan menampilkan pesan “elemen-0: “ yang sesuai dengan perintah pada baris ke 115. Kemudian user diminta untuk memasukkan nilai dari tiap elemen yang akan diurutkan. Hal ini sesuai dengan perintah pada baris ke 116. Pada percobaan ini, saya memasukkan nilai 12, 90, 34, 78, dan 56. Kemudian program menampilkan pesan “Kondisi awal array” dan menampilkan kondisi awal dari array yang belum diurutkan yang sesuai dengan perintah pada baris ke 41 dan 42. Kemudian program menampilkan 8 menu, menu 1 untuk bubble sort, menu 2 untuk exchange sort, menu 3 untuk selection sort, menu 4 untuk insertion sort, menu 5 untuk memasukkan ulang array, menu 6 untuk melakukan sorting dengan nilai random, menu 7 untuk melakukan sorting tanpa data, dan menu 8 untuk mengakhiri program. Karena akan melakukan pengurutan dengan selection sort, maka dipilih menu 3. Kemudian program menampilkan pesan “#03 Metode Selection Sort” dan menampilkan kondisi array awal seperti sebelumnya. Kemudian, program melakukan proses sorting dan menampilkan data yang ada. Program menampilkan hasil sorting dengan metode terpilih, yaitu 12, 34, 56, 78, dan 90, sesuai dengan perintah pada baris ke 223. Kemudian program menampilkan total Langkah yang dilakukan, sesuai dengan perintah pada baris ke 224 dan total pertukaran yang dilakukan, sesuai dengan perintah pada baris ke 225. Kemudian program menampilkan kembali menu awal, dan dipilih menu 8 akhiri program.

```

3\Prakt. Struktur Data\P2\Code\" ; if ($?) { g++ bak.cpp -o bak } ; if ($?) { .\bak }

Program Penerapan Metode Sorting.
Ukuran array: 5
elemen-0: 12
elemen-1: 90
elemen-2: 34
elemen-3: 78
elemen-4: 56

Kondisi awal array:
    12 90 34 78 56

=====
Program Penerapan Metode Sorting
=====

1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 4

#04 Metode Insertion Sort
Kondisi awal array:
    12 90 34 78 56
Proses Sorting
    12 12 34 78 56
    90 12 12 78 56
    90 34 12 12 56
    90 34 34 12 56
    90 78 34 12 12
    90 78 34 34 12
Hasil Sorting Isertion Sort:
    90 78 56 34 12

Total langkah Isertion Sort: 6
Total pertukaran Isertion Sort: 6

=====
Program Penerapan Metode Sorting
=====

1. Bubble Sort
2. Exchange Sort
3. Selection Sort
4. Insertion Sort
5. Input ulang Array
6. Test dengan nilai random
7. Test dengan array kosong
8. AKHIRI PROGRAM
Silahkan pilih antara [1-8]: 8

○ Program Selesai.

```

Gambar 3.4 Output Insertion sort

Berdasarkan gambar 3.4, dapat dilihat bahwa pertama program menampilkan pesan “Program penerapan metode sorting” yang sesuai dengan perintah pada baris ke 32. Kemudian program menampilkan pesan “Ukuran array: “ yang sesuai dengan

perintah pada baris ke 35. Kemudian user diminta untuk memasukkan nilai yang nantinya akan menjadi Panjang dari array. Pada percobaan ini saya memasukkan 5 nilai. Kemudian program akan menampilkan pesan “elemen-0: “ yang sesuai dengan perintah pada baris ke 115. Kemudian user diminta untuk memasukkan nilai dari tiap elemen yang akan diurutkan. Hal ini sesuai dengan perintah pada baris ke 116. Pada percobaan ini, saya memasukkan nilai 12, 90, 34, 78, dan 56. Kemudian program menampilkan pesan “Kondisi awal array” dan menampilkan kondisi awal dari array yang belum diurutkan yang sesuai dengan perintah pada baris ke 41 dan 42. Kemudian program menampilkan 8 menu, menu 1 untuk bubble sort, menu 2 untuk exchange sort, menu 3 untuk selection sort, menu 4 untuk insertion sort, menu 5 untuk memasukkan ulang array, menu 6 untuk melakukan sorting dengan nilai random, menu 7 untuk melakukan sorting tanpa data, dan menu 8 untuk mengakhiri program. Karena akan melakukan pengurutan dengan bubble sort, maka dipilih menu 1. Kemudian program menampilkan pesan “#04 Metode Insertion Sort” dan menampilkan kondisi array awal seperti sebelumnya. Kemudian, program melakukan proses sorting dan menampilkan data yang ada. Program menampilkan hasil sorting dengan metode terpilih, yaitu 12, 34, 56, 78, dan 90, sesuai dengan perintah pada baris ke 252. Kemudian program menampilkan total Langkah yang dilakukan, sesuai dengan perintah pada baris ke 253 dan total pertukaran yang dilakukan, sesuai dengan perintah pada baris ke 254. Kemudian program menampilkan kembali menu awal, dan dipilih menu 8 akhiri program.