

Meeting 8

Avicena Yosanta Muchtar 24/536710/PA/22764

14 November 2024

<https://github.com/TheKurusUGM/Praktikum-Pemograman-UGM-/tree/main>

1 Question 1:

Question:

Sort the data in descending by:

- NISN
- Value

using insertion sort, selection sort, and bubble sort.

Answer:

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  struct Student {
6      long long NISN;
7      std::string Nama;
8      int Value;
9  };
10
11 void displayData(const std::vector<Student>& students) {
12     for (const auto& student : students) {
13         std::cout << student.NISN << " " << student.Nama << " "
14             << student.Value << std::endl;
15     }
16 }
17
18 void insertionSortByNISN(std::vector<Student>& students) {
19     for (size_t i = 1; i < students.size(); i++) {
20         Student key = students[i];
21         int j = i - 1;
22         while (j >= 0 && students[j].NISN < key.NISN) {
23             students[j + 1] = students[j];
```

```

23         j--;
24     }
25     students[j + 1] = key;
26 }
27 }
28
29 void selectionSortByValue(std::vector<Student>& students) {
30     for (size_t i = 0; i < students.size() - 1; i++) {
31         size_t maxIdx = i;
32         for (size_t j = i + 1; j < students.size(); j++) {
33             if (students[j].Value > students[maxIdx].Value) {
34                 maxIdx = j;
35             }
36         }
37         std::swap(students[i], students[maxIdx]);
38     }
39 }
40
41 void bubbleSortByNISN(std::vector<Student>& students) {
42     bool swapped;
43     for (size_t i = 0; i < students.size() - 1; i++) {
44         swapped = false;
45         for (size_t j = 0; j < students.size() - i - 1; j++) {
46             if (students[j].NISN < students[j + 1].NISN) {
47                 std::swap(students[j], students[j + 1]);
48                 swapped = true;
49             }
50         }
51         if (!swapped) break;
52     }
53 }
54
55 void bubbleSortByValue(std::vector<Student>& students) {
56     bool swapped;
57     for (size_t i = 0; i < students.size() - 1; i++) {
58         swapped = false;
59         for (size_t j = 0; j < students.size() - i - 1; j++) {
60             if (students[j].Value < students[j + 1].Value) {
61                 std::swap(students[j], students[j + 1]);
62                 swapped = true;
63             }
64         }
65         if (!swapped) break;
66     }
67 }
68

```

```

69 int main() {
70     std::vector<Student> students = {
71         {9960312699, "Handi Ramadhan", 90},
72         {9963959682, "Rio Alfandra", 55},
73         {9950310962, "Ronaldo Valentino Uneputty", 80},
74         {9970272750, "Achmad Yaumil Fadjri R.", 60},
75         {9970293945, "Alivia Rahma Pramesti", 70},
76         {9952382180, "Ari Lutfianto", 65},
77         {9965653989, "Arief Budiman", 60}
78     };
79
80     std::cout << "Original Data:\n";
81     displayData(students);
82
83     insertionSortByNISN(students);
84     std::cout << "\nSorted by NISN (Descending) using Insertion
85     ↪ Sort:\n";
86     displayData(students);
87
88     selectionSortByValue(students);
89     std::cout << "\nSorted by Value (Descending) using Selection
90     ↪ Sort:\n";
91     displayData(students);
92
93     students = {
94         {9960312699, "Handi Ramadhan", 90},
95         {9963959682, "Rio Alfandra", 55},
96         {9950310962, "Ronaldo Valentino Uneputty", 80},
97         {9970272750, "Achmad Yaumil Fadjri R.", 60},
98         {9970293945, "Alivia Rahma Pramesti", 70},
99         {9952382180, "Ari Lutfianto", 65},
100        {9965653989, "Arief Budiman", 60}
101    };
102
103    bubbleSortByNISN(students);
104    std::cout << "\nSorted by NISN (Descending) using Bubble
105    ↪ Sort:\n";
106    displayData(students);
107
108    students = {
109        {9960312699, "Handi Ramadhan", 90},
110        {9963959682, "Rio Alfandra", 55},
111        {9950310962, "Ronaldo Valentino Uneputty", 80},
112        {9970272750, "Achmad Yaumil Fadjri R.", 60},
113        {9970293945, "Alivia Rahma Pramesti", 70},
114        {9952382180, "Ari Lutfianto", 65},

```

```

112         {9965653989, "Arief Budiman", 60}
113     };
114
115     bubbleSortByValue(students);
116     std::cout << "\nSorted by Value (Descending) using Bubble
    ↪ Sort:\n";
117     displayData(students);
118
119     return 0;
120 }

```

2 Explanation

```

1 struct Student {
2     long long NISN;
3     std::string Nama;
4     int Value;
5 };

```

The struct Student defines the data structure for each student's information.

```

1 void displayData(const std::vector<Student>& students) {
2     for (const auto& student : students) {
3         std::cout << student.NISN << " " << student.Nama << " "
    ↪ << student.Value << std::endl;
4     }
5 }

```

The displayData function displays each student's information from the vector.

```

1 void insertionSortByNISN(std::vector<Student>& students) {
2     for (size_t i = 1; i < students.size(); i++) {
3         Student key = students[i];
4         int j = i - 1;
5         while (j >= 0 && students[j].NISN < key.NISN) {
6             students[j + 1] = students[j];
7             j--;
8         }
9         students[j + 1] = key;
10    }
11 }

```

This function sorts students in descending order based on NISN. For each element in the list, it finds the correct position by constantly changing elements until NISN is in descending order.

```

1 void selectionSortByValue(std::vector<Student>& students) {
2     for (size_t i = 0; i < students.size() - 1; i++) {
3         size_t maxIdx = i;
4         for (size_t j = i + 1; j < students.size(); j++) {
5             if (students[j].Value > students[maxIdx].Value) {
6                 maxIdx = j;
7             }
8         }
9         std::swap(students[i], students[maxIdx]);
10    }
11 }

```

This function sorts students by Value in descending order. It finds the maximum value in the remaining unsorted part of the list and swaps it with the current position.

bubbleSortByNISN and bubbleSortByValue are two separate bubble sorts, one for NISN and one for Value. Bubble Sort repeatedly swaps elements if they are out of order until the entire list is sorted.

```

1 int main() {
2     std::vector<Student> students = {
3         {9960312699, "Handi Ramadhan", 90},
4         {9963959682, "Rio Alfandra", 55},
5         // ... other students ...
6     };
7     // Sort and display results for each sort
8 }

```

Initializes the list of students.

- Calls each sorting function on a copy of students to sort it by NISN and Value in descending order using different algorithms.
- After each sort, it displays the sorted list.

3 Question 2

Question:

Look for data that has NISN 9950310962, then show the value using the binary search.

```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4

```

```

5 struct Student {
6     long long NISN;
7     std::string Nama;
8     int Value;
9 };
10
11 // Binary search function
12 int binarySearch(const std::vector<Student>& students, long long
    ↪ targetNISN) {
13     int left = 0;
14     int right = students.size() - 1;
15     while (left <= right) {
16         int mid = left + (right - left) / 2;
17         if (students[mid].NISN == targetNISN) {
18             return students[mid].Value; // Return the Value if
    ↪ found
19         } else if (students[mid].NISN < targetNISN) {
20             left = mid + 1;
21         } else {
22             right = mid - 1;
23         }
24     }
25     return -1; // Return -1 if NISN is not found
26 }
27
28 int main() {
29     std::vector<Student> students = {
30         {9960312699, "Handi Ramadhan", 90},
31         {9963959682, "Rio Alfandra", 55},
32         {9950310962, "Ronaldo Valentino Uneputty", 80},
33         {9970272750, "Achmad Yaumil Fadjri R.", 60},
34         {9970293945, "Alivia Rahma Pramesti", 70},
35         {9952382180, "Ari Lutfianto", 65},
36         {9965653989, "Arief Budiman", 60}
37     };
38
39     // Sort the students by NISN (ascending) to prepare for
    ↪ binary search
40     std::sort(students.begin(), students.end(), [](const
    ↪ Student& a, const Student& b) {
41         return a.NISN < b.NISN;
42     });
43
44     long long targetNISN = 9950310962;
45     int value = binarySearch(students, targetNISN);
46

```

```

47     if (value != -1) {
48         std::cout << "The value for NISN " << targetNISN << "
           ↳ is: " << value << std::endl;
49     } else {
50         std::cout << "NISN " << targetNISN << " not found." <<
           ↳ std::endl;
51     }
52
53     return 0;
54 }

```

Binary Search Function (binarySearch):

- This function performs binary search on the sorted vector of Student structures.
- If the targetNISN matches a student's NISN, it returns the Value.
- If the NISN is not found, it returns -1.

Sorting by NISN:

- Before applying binary search, the code sort the students vector by NISN in ascending order using std::sort with a lambda function.
- The main function searches for NISN 9950310962 and outputs the associated Value if found. Otherwise, it prints "not found."

4 Question 3

Question:

Change the name of the data that has a value of 60 to Joko. Take advantage of sequential search methods.

```

1  #include <iostream>
2  #include <vector>
3
4  struct Student {
5      long long NISN;
6      std::string Nama;
7      int Value;
8  };
9
10 void displayData(const std::vector<Student>& students) {
11     for (const auto& student : students) {
12         std::cout << student.NISN << " " << student.Nama << " "
           ↳ << student.Value << std::endl;
13     }
14 }
15
16

```

```

17 void updateNameByValue(std::vector<Student>& students, int
   ↪   targetValue, const std::string& newName) {
18     for (auto& student : students) {
19         if (student.Value == targetValue) {
20             student>Nama = newName;
21         }
22     }
23 }
24
25 int main() {
26     std::vector<Student> students = {
27         {9960312699, "Handi Ramadhan", 90},
28         {9963959682, "Rio Alfandra", 55},
29         {9950310962, "Ronaldo Valentino Uneputty", 80},
30         {9970272750, "Achmad Yaumil Fadjri R.", 60},
31         {9970293945, "Alivia Rahma Pramesti", 70},
32         {9952382180, "Ari Lutfianto", 65},
33         {9965653989, "Arief Budiman", 60}
34     };
35
36     std::cout << "Original Data:\n";
37     displayData(students);
38
39
40     updateNameByValue(students, 60, "Joko");
41
42     std::cout << "\nData after updating names with Value 60 to
   ↪     'Joko':\n";
43     displayData(students);
44
45     return 0;
46 }

```

5 Explanation

```

1 struct Student {
2     long long NISN;
3     std::string>Nama;
4     int Value;
5 };

```

Defines the Student struct with NISN,>Nama, and Value fields to represent each student's data.


```

1 void updateNameByValue(std::vector<Student>& students, int
  ↪   targetValue, const std::string& newName) {
2     for (auto& student : students) {
3         if (student.Value == targetValue) {
4             student>Nama = newName;
5         }
6     }
7 }

```

This function searches for students with a Value of 60 and updates their Nama to "Joko".

```

1 int main() {
2     std::vector<Student> students = {
3         {9960312699, "Handi Ramadhan", 90},
4         {9963959682, "Rio Alfandra", 55},
5         {9950310962, "Ronaldo Valentino Uneputty", 80},
6         {9970272750, "Achmad Yaumil Fadjri R.", 60},
7         {9970293945, "Alivia Rahma Pramesti", 70},
8         {9952382180, "Ari Lutfianto", 65},
9         {9965653989, "Arief Budiman", 60}
10    };
11
12    updateNameByValue(students, 60, "Joko");
13
14    displayData(students);
15    return 0;
16 }

```

Initializes the students vector with the given data.
 Calls updateNameByValue to change the name of students with Value of 60 to "Joko".
 Displays data before and after the update.