

LAPORAN PRAKTIKUM

ALGORITMA DAN PEMROGRAMAN



Disusun Oleh:

WAFI FAHRUZZAMAN – 140810200009

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

JATINANGOR

2020

1. Tujuan

- 1.1. Memahami konsep Array dan Sort (Sorting) dalam pemrogram C++ secara benar.
- 1.2. Mampu mengimplementasikan Array dan Sort (Sorting) pada program secara tepat.

2. Materi

2.1. Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun array disebut elemen array(isi), yang masing-masing elemen dapat diakses tersendiri melalui indeks array.

Berikut adalah penulisan notasi array:

$$\sum_{i=0}^{n-1} A[i] = A[0], A[1], A[2], A[3], A[4], \dots, A[n-1]$$

A : Array
n : Panjang array
i : Index array

Array dapat memiliki satu atau banyak dimensi dan secara teori tidak ada batasnya. Faktor yang membatasi adalah memori dari komputer yang menyimpan array itu sendiri. Inisialisasi dimensi array dapat dilakukan dengan menambahkan [] pada variabel. Contohnya: `int matriks[4][4];`

Terdapat cara umum dalam mendeklarasikan variabel array

2.1.1. Deklarasi Secara Langsung

Untuk mendeklarasikan sebuah array dalam C++, menggunakan tanda [] (bracket). Bentuk umum dari pendeklarasian array adalah sebagai berikut:

TipeData namaArray [jumlah_element];

contoh : (misalnya nama LARIK) yang memiliki 10 elemen dengan tipe data int, maka bentuk array nya adalah sebagai berikut:

`int LARIK [10];`

2.1.2. Deklarasi Secara Tidak Langsung

Dalam pendeklarasian secara tidak langsung, terdapat specifier typedef yang digunakan untuk membuat alias tipe data baru.

2.1.3. Deklarasi Sekaligus Inialisasi

Dalam deklarasi sekaligus inialisasi, terdapat fungsi `sizeof()` digunakan untuk mengembalikan nilai alokasi memori suatu variabel. Alokasi memori array yang dideklarasikan akan menyesuaikan size nya.

2.2. Sorting

Merupakan suatu fungsi untuk mengurutkan data berdasarkan nilainya secara ascending atau descending. Ada banyak algoritma sorting yang dapat digunakan dalam berbagai bahasa pemrograman. Perbedaan dari berbagai algoritma tersebut terletak pada efisiensinya/kompleksitasnya. Beberapa contoh algoritma sorting

2.2.1. Bubble Sort

Bubble Sort adalah metode yang membandingkan elemen yang sekarang dengan elemen-elemen berikutnya. Perbandingan elemen dapat dimulai dari awal atau mulai dari paling akhir. Apabila elemen yang sekarang lebih besar (untuk urut menaik) atau lebih kecil (untuk urut menaik) dari elemen berikutnya, maka posisinya ditukar, tetapi jika tidak maka posisinya tetap.

2.2.2. Insertion Sort

Metode Insertion Sort merupakan metode pengurutan dengan cara menyisipkan elemen array pada posisi yang tepat. Pencarian yang tepat dilakukan dengan melakukan pencarian beruntun di dalam array. Selama pencarian posisi yang tepat dilakukan pergeseran elemen array. Algoritma pengurutan ini tepat untuk persoalan menyisipkan elemen baru ke dalam array yang sudah terurut. Misalnya dalam permainan kartu, kartu yang dicabut biasanya disisipkan oleh pemain pada posisi yang tepat sehingga penambahan kartu tersebut membuat semua kartu tetap terurut.

2.2.3. Selection Sort

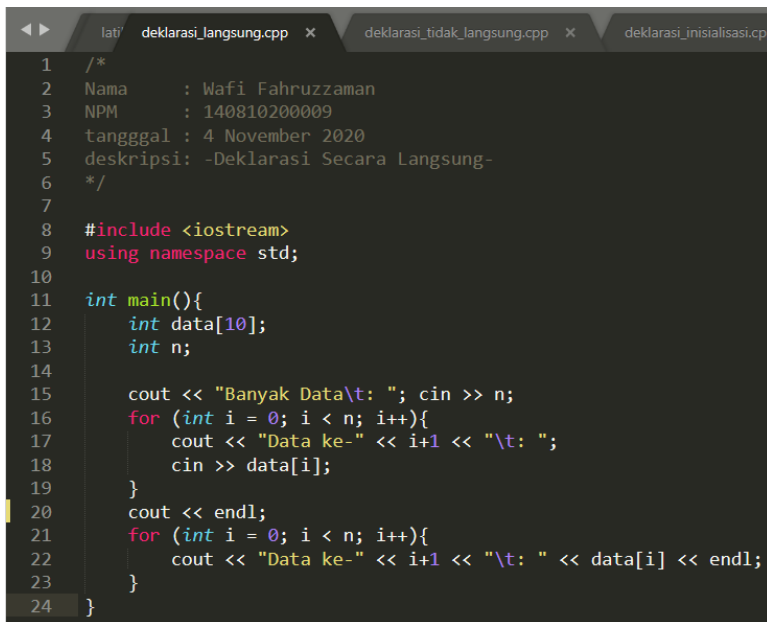
Program metode selection sort c++ ini merupakan kombinasi antara sorting dan searching. Jadi setiap proses akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil atau terbesar akan dipertukarkan ke posisi yang tepat di dalam array.

2.2.4. Merge Sort

Metode ini memanfaatkan keteraturan yang diperoleh dari hasil merging dua buah array. Suatu array Nilai yang mempunyai N elemen (Nilai[0..N-1]) dianggap terdiri dari N array yang masing-masing terdiri dari satu elemen. Untuk pasangan array yang berdekatan kita lakukan merging sehingga diperoleh N/2 buah array yang masing-masing array memiliki 2 elemen (jika N ganjil, akan terdapat sebuah array dengan 1 elemen). Pada saat melakukan proses merging dilakukan pengaturan posisi dengan cara elemen yang lebih kecil diletakkan di posisi awal (untuk pengurutan secara menaik) dan elemen yang lebih besar diletakkan di posisi awal (untuk pengurutan secara menurun). Kemudian dilakukan merging kembali untuk setiap pasangan array seperti cara di atas sehingga kita peroleh N/2 buah array yang masing-masing array memiliki 4 elemen. Langkah ini kita teruskan hingga kita memperoleh sebuah array yang sudah dalam keadaan terurut.

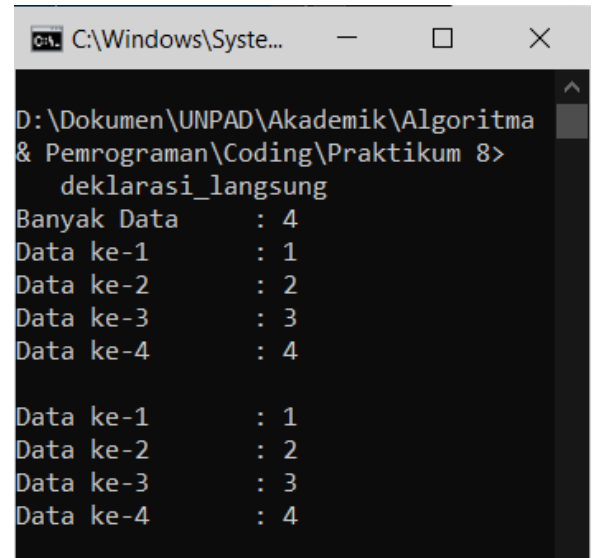
3. Latihan

3.1. Deklarasi Secara Langsung



```
1  /*
2  Nama      : Wafi Fahrurzzaman
3  NPM       : 140810200009
4  tanggal  : 4 November 2020
5  deskripsi : -Deklarasi Secara Langsung-
6  */
7
8  #include <iostream>
9  using namespace std;
10
11 int main(){
12     int data[10];
13     int n;
14
15     cout << "Banyak Data\t: "; cin >> n;
16     for (int i = 0; i < n; i++){
17         cout << "Data ke-" << i+1 << "\t: ";
18         cin >> data[i];
19     }
20     cout << endl;
21     for (int i = 0; i < n; i++){
22         cout << "Data ke-" << i+1 << "\t: " << data[i] << endl;
23     }
24 }
```

Gambar 3.1.a. coding (.cpp)

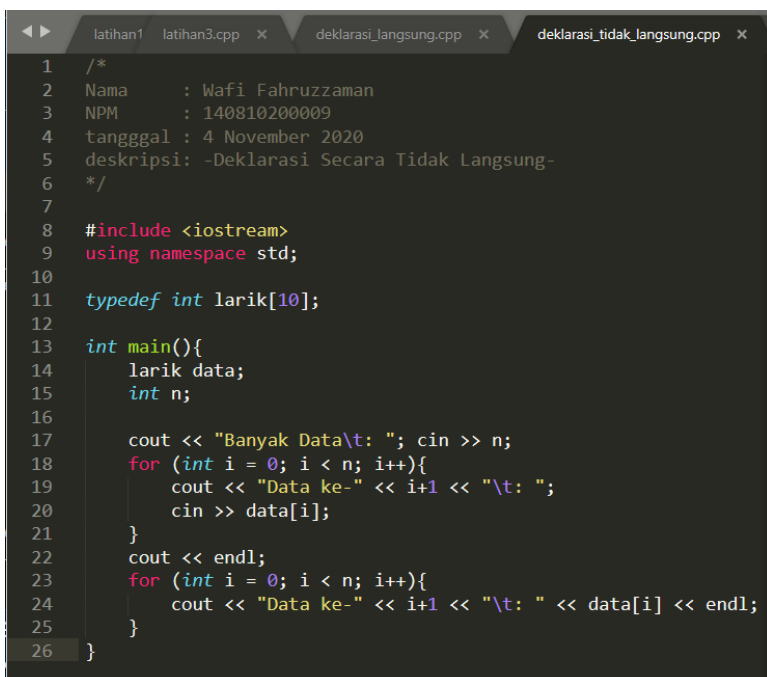


```
C:\Windows\System32\cmd.exe
D:\Dokumen\UNPAD\Akademik\Algoritma & Pemrograman\Coding\Praktikum 8>
deklarasi_langsung
Banyak Data      : 4
Data ke-1        : 1
Data ke-2        : 2
Data ke-3        : 3
Data ke-4        : 4

Data ke-1        : 1
Data ke-2        : 2
Data ke-3        : 3
Data ke-4        : 4
```

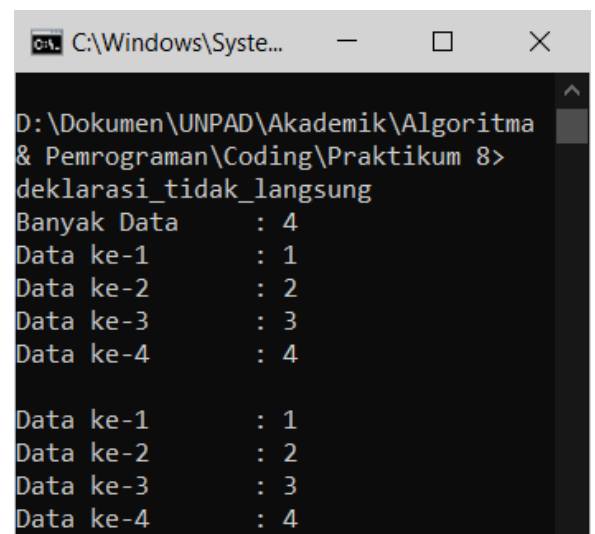
Gambar 3.1.b. program/hasil (.exe)

3.2. Deklarasi Secara Tidak Langsung



```
1  /*
2  Nama      : Wafi Fahrurzzaman
3  NPM       : 140810200009
4  tanggal  : 4 November 2020
5  deskripsi : -Deklarasi Secara Tidak Langsung-
6  */
7
8  #include <iostream>
9  using namespace std;
10
11 typedef int larik[10];
12
13 int main(){
14     larik data;
15     int n;
16
17     cout << "Banyak Data\t: "; cin >> n;
18     for (int i = 0; i < n; i++){
19         cout << "Data ke-" << i+1 << "\t: ";
20         cin >> data[i];
21     }
22     cout << endl;
23     for (int i = 0; i < n; i++){
24         cout << "Data ke-" << i+1 << "\t: " << data[i] << endl;
25     }
26 }
```

Gambar 3.2.a. coding (.cpp)

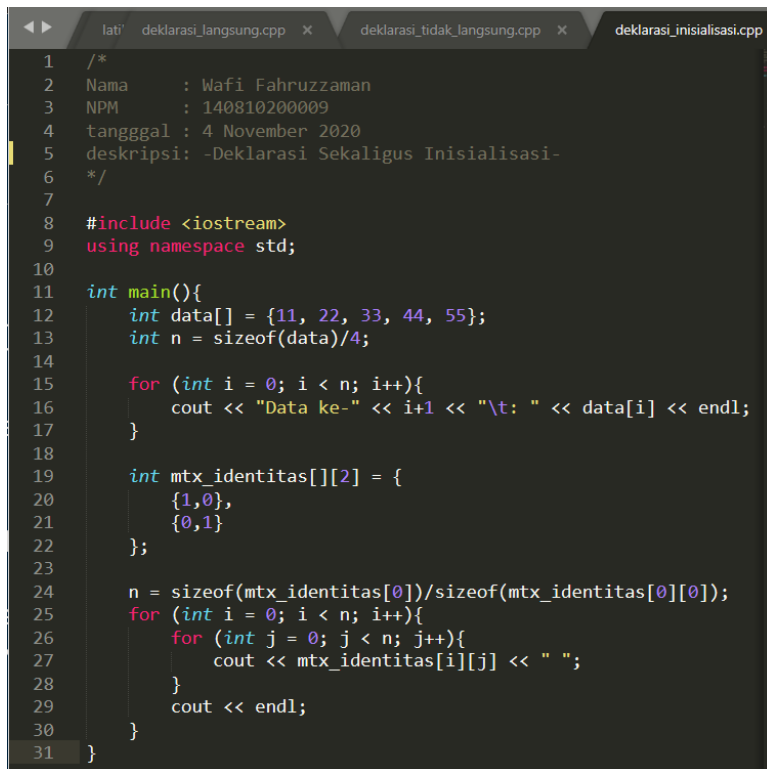


```
C:\Windows\System32\cmd.exe
D:\Dokumen\UNPAD\Akademik\Algoritma & Pemrograman\Coding\Praktikum 8>
deklarasi_tidak_langsung
Banyak Data      : 4
Data ke-1        : 1
Data ke-2        : 2
Data ke-3        : 3
Data ke-4        : 4

Data ke-1        : 1
Data ke-2        : 2
Data ke-3        : 3
Data ke-4        : 4
```

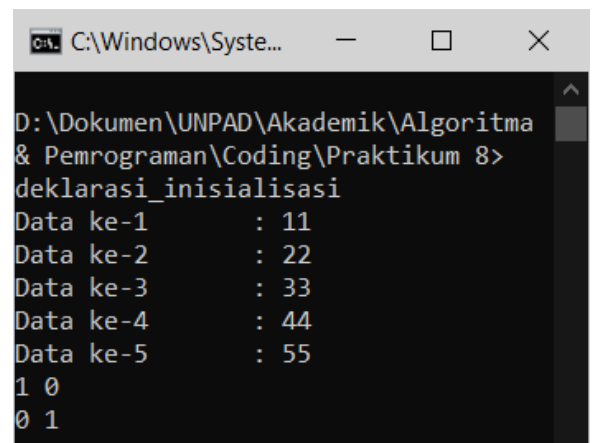
Gambar 3.2.b. program/hasil (.exe)

3.3. Deklarasi Sekaligus Inisialisasi



```
1  /*
2  Nama      : Wafi Fahrurzzaman
3  NPM       : 140810200009
4  tanggal  : 4 November 2020
5  deskripsi: -Deklarasi Sekaligus Inisialisasi-
6  */
7
8  #include <iostream>
9  using namespace std;
10
11 int main(){
12     int data[] = {11, 22, 33, 44, 55};
13     int n = sizeof(data)/4;
14
15     for (int i = 0; i < n; i++){
16         cout << "Data ke-" << i+1 << "\t: " << data[i] << endl;
17     }
18
19     int mtx_identitas[][2] = {
20         {1,0},
21         {0,1}
22     };
23
24     n = sizeof(mtx_identitas[0])/sizeof(mtx_identitas[0][0]);
25     for (int i = 0; i < n; i++){
26         for (int j = 0; j < n; j++){
27             cout << mtx_identitas[i][j] << " ";
28         }
29         cout << endl;
30     }
31 }
```

Gambar 3.3.a. coding (.cpp)



```
C:\Windows\Syste...
D:\Dokumen\UNPAD\Akademik\Algoritma
& Pemrograman\Coding\Praktikum 8>
deklarasi_inisialisasi
Data ke-1      : 11
Data ke-2      : 22
Data ke-3      : 33
Data ke-4      : 44
Data ke-5      : 55
1 0
0 1
```

Gambar 3.3.b. program/hasil (.exe)

3.4. Latihan 1

```
1  /*
2  Nama      : Wafi Fahrurrazman
3  NPM       : 140810200009
4  tanggal  : 4 November 2020
5  deskripsi: -Latihan 1-
6  */
7
8  #include <iostream>
9  using namespace std;
10
11 void banyakData(int& n);
12 void isilarik(int a[], int n);
13 void printLarik(int a[], int n);
14
15 int main(){
16     int x[10];
17     int n;
18     banyakData(n);
19     isilarik(x, n);
20     printLarik(x, n);
21 }
22
23
24 void banyakData(int& n){
25     cout << "Banyak Data : "; cin >> n;
26 }
27
28 void isilarik(int a[], int n){
29     cout << "Silahkan masukkan data larik:" << endl;
30     for (int i = 0; i < n; i++){
31         cout << "Data ke-" << i+1 << " = "; cin >> a[i];
32     }
33 }
34
35 void printLarik(int a[], int n){
36     cout << "\nData yang telah dimasukkan ke larik:" << endl;
37     for (int i = 0; i < n; i++){
38         cout << "Data ke-" << i+1 << " = " << a[i] << endl;
39     }
40 }
```

Gambar 3.4.a. coding (.cpp)

```
C:\Windows\System32\cmd.exe
D:\Dokumen\UNPAD\Akademik\Algoritma
& Pemrograman\Coding\Praktikum 8>
latihan1
Banyak Data : 5
Silahkan masukkan data larik:
Data ke-1 = 3
Data ke-2 = 7
Data ke-3 = 4
Data ke-4 = 8
Data ke-5 = 2

Data yang telah dimasukkan ke larik:
Data ke-1 = 3
Data ke-2 = 7
Data ke-3 = 4
Data ke-4 = 8
Data ke-5 = 2
```

Gambar 3.4.b. program/hasil (.exe)

3.5. Latihan 2

```
1  /*
2  Nama      : Wafi Fahrurrazaman
3  NPM       : 140810200009
4  tanggal  : 4 November 2020
5  deskripsi : -Latihan 2-
6  */
7
8  #include <iostream>
9  using namespace std;
10
11  typedef int matriks[10][10];
12
13  void banyakData(int& a, int& b);
14  void isiMatriks(matriks x, int a, int b);
15  void cetakMatriks(matriks x, int a, int b);
16
17  int main(){
18      matriks x;
19      int nBaris, nKolom;
20      banyakData(nBaris, nKolom);
21      isiMatriks(x, nBaris, nKolom);
22      cetakMatriks(x, nBaris, nKolom);
23  }
24
25  void banyakData(int& a, int& b){
26      cout << "Banyak Baris : "; cin >> a;
27      cout << "Banyak Kolom : "; cin >> b;
28  }
29
30  void isiMatriks(matriks x, int a, int b){
31      for (int i = 0; i < a; i++){
32          for (int j = 0; j < b; j++){
33              cout << "Baris " << i+1 << ", Kolom " << j+1 << " = ";
34              cin >> x[i][j];
35          }
36      }
37  }
38
39  void cetakMatriks(matriks x, int a, int b){
40      for (int i = 0; i < a; i++){
41          for (int j = 0; j < b; j++){
42              cout << x[i][j] << "\t";
43          }
44          cout << endl;
45      }
46  }
```

Gambar 3.5.a. coding (.cpp)

```
C:\Windows\System32\cmd.exe
D:\Dokumen\UNPAD\Akademik\Algoritma
& Pemrograman\Coding\Praktikum 8>
latihan2
Banyak Baris : 2
Banyak Kolom : 2
Baris 1, Kolom 1 = 2
Baris 1, Kolom 2 = 1
Baris 2, Kolom 1 = 4
Baris 2, Kolom 2 = 3
2      1
4      3
```

Gambar 3.5.b. program/hasil (.exe)

3.6. Latihan 3

```
latihan3.cpp
1  /*
2  Nama      : Wafi Fahrizzaman
3  NPM       : 140810200009
4  tanggal  : 4 November 2020
5  deskripsi: -Latihan 3-
6  */
7
8  #include <iostream>
9  using namespace std;
10
11 void swap(int& a, int& b);
12 void bubbleSort(int arr[], int n);
13 void printArr(int arr[], int n);
14
15 int main(){
16     int larik[] = {12, 36, 5, 19, 8};
17     int n = sizeof(larik)/sizeof(larik[0]);
18     cout << "Array sebelum sorting:\n"; printArr(larik, n);
19     bubbleSort(larik, n);
20     cout << "\nArray setelah sorting:\n"; printArr(larik, n);
21 }
22
23 void swap(int& a, int& b){
24     int temp = a;
25     a = b;
26     b = temp;
27 }
28
29 void bubbleSort(int arr[], int n){
30     for (int i = 0; i < n-1; i++){
31         for (int j = 0; j < n-i-1; j++){
32             if (arr[j] > arr[j+1])
33                 swap(arr[j], arr[j+1]);
34         }
35     }
36 }
37
38 void printArr(int arr[], int n){
39     for (int i = 0; i < n; i++){
40         cout << "Data ke-" << i+1 << " = " << arr[i] << endl;
41     }
42 }
```

Gambar 3.6.a. coding (.cpp)

```
C:\Windows\Syste...
D:\Dokumen\UNPAD\Akademik\Algoritma
& Pemrograman\Coding\Praktikum 8>
latihan3
Array sebelum sorting:
Data ke-1 = 12
Data ke-2 = 36
Data ke-3 = 5
Data ke-4 = 19
Data ke-5 = 8

Array setelah sorting:
Data ke-1 = 5
Data ke-2 = 8
Data ke-3 = 12
Data ke-4 = 19
Data ke-5 = 36
```

Gambar 3.6.b program/hasil (.exe)

4. Tugas

4.1. Tugas 1

```
1.  /*
2.   Nama      : Wafi Fahrurzzaman
3.   NPM       : 140810200009
4.   tanggal  : 4 November 2020
5.   deskripsi: -Tugas 1-
6.  */
7.
8.  #include <iostream>
9.  using namespace std;
10.
11. typedef int matriks[10][10];
12. void header(string isi);
13. void menu(int& pilihan);
14. void ukuranMatriks(int& a, int& b);
15. void ukuranMatriks2(int& c, int& d);
16. void inputMatriks(matriks x, int a, int b);
17. void printMatriks(matriks x, int a, int b);
18. void penambahan(matriks hasilTambah, matriks x, matriks y, int a, int b);
19. void perkalian(matriks hasilKali, matriks x, matriks y, int a, int b, int c, int d);
20. void transpose(matriks x, int a, int b);
21.
22. int main(){
23.     matriks satu, dua, hasilTambah, hasilKali;
24.     int nBaris, nKolom, nBaris1, nKolom1, nBaris2, nKolom2;
25.     int pilihan = 0;
26.     while (pilihan != 4){
27.         menu(pilihan);
28.         switch(pilihan){
29.             case 1 :
30.                 cout << endl;
31.                 header ("-Penjumlahan Matriks-");
32.                 ukuranMatriks(nBaris, nKolom);
33.                 cout << "\n== Matriks 1 ==\n";
34.                 inputMatriks(satu, nBaris, nKolom);
35.                 cout << "\n== Matriks 2 ==\n";
36.                 inputMatriks(dua, nBaris, nKolom);
37.                 cout << endl;
38.                 printMatriks(satu, nBaris, nKolom);
39.                 cout << "\n      +\n\n";
40.                 printMatriks(dua, nBaris, nKolom);
41.                 penambahan(hasilTambah, satu, dua, nBaris, nKolom);
42.                 cout << endl;
43.                 break;
44.         }
```

```

45.         case 2 :
46.             perkalian :
47.             cout << endl;
48.             header ("-Perkalian Matriks-");
49.             cout << "Syarat: Banyak Kolom Mstriks 1 = Banyak Baris Matriks 2";
50.             cout << "\n\n== Matriks 1 ==\n";
51.             ukuranMatriks(nBaris1, nKolom1);
52.             cout << "\n== Matriks 2 ==\n";
53.             ukuranMatriks2(nBaris2, nKolom2);
54.
55.             if (nKolom1 != nBaris2){
56.                 cout << "\n Matriks 1 & Matriks 2 Tidak Dapat Dikali\n";
57.                 cout << " PERHATIKAN SYARAT!!!\n\n";
58.                 goto perkalian;
59.
60.             } else {
61.                 cout << "\n--Input Elemen Matriks--";
62.                 cout << "\n\n== Matriks 1 ==\n";
63.                 inputMatriks(satu, nBaris1, nKolom1);
64.                 cout << "\n== Matriks 2 ==\n";
65.                 inputMatriks(dua, nBaris2, nKolom2);
66.                 cout << endl;
67.                 printMatriks(satu, nBaris1, nKolom1);
68.                 cout << "\n      *\n\n";
69.                 printMatriks(dua, nBaris2, nKolom2);
70.                 perkalian(hasilKali, satu, dua, nBaris1, nKolom1, nBaris2, nKolom2);
71.                 cout << endl;
72.             }
73.             break;
74.
75.         case 3 :
76.             cout << endl;
77.             header ("-Transpose Matriks-");
78.             ukuranMatriks(nBaris, nKolom);
79.             cout << "\n== Input Elemen Matriks ==\n";
80.             inputMatriks(satu, nBaris, nKolom);
81.             cout << "\n== Matriks ==\n";
82.             printMatriks(satu, nBaris, nKolom);
83.             cout << "\n== Hasil Transpose ==\n";
84.             transpose(satu, nKolom, nBaris);
85.             cout << endl;
86.             break;
87.
88.         case 4 :
89.             cout << endl;
90.             header ("-PROGRAM SELESAI-");
91.             cout << endl;
92.             break;
93.

```

```

94.         default:
95.             system("cls");
96.
97.             cout << "\nMenu tidak ada. Cek kembali daftar Menu!\n";
98.         }
99.     }
100. }
101.
102. void header(string isi){
103.     for (int i=1; i<= isi.length(); i++){
104.         cout << "=";
105.
106.         if (i == isi.length()){
107.             cout << endl;
108.         }
109.     }
110.
111.     cout << isi << endl;
112.
113.     for (int i=1; i <= isi.length(); i++){
114.         cout << "=";
115.
116.         if (i == isi.length()){
117.             cout << endl;
118.         }
119.     }
120. }
121.
122. void menu(int& pilihan){
123.     header("-----MENU-----");
124.     cout << "1. Penjumlahan Matriks\n";
125.     cout << "2. Perkalian Matriks\n";
126.     cout << "3. Transpose Matriks\n";
127.     cout << "4. Selesai\n";
128.     cout << "-----\n";
129.     cout << "Pilih Menu [1-4] : ";
130.     cin >> pilihan;
131. }
132.
133. void ukuranMatriks(int& a, int& b){
134.     cout << "Banyak Baris\t: "; cin >> a;
135.     cout << "Banyak Kolom\t: "; cin >> b;
136. }
137.
138. void ukuranMatriks2(int& c, int& d){
139.     cout << "Banyak Baris\t: "; cin >> c;
140.     cout << "Banyak Kolom\t: "; cin >> d;
141. }
142.

```

```

143.     void inputMatriks(matriks x, int a, int b){
144.         for (int i = 0; i < a; i++){
145.             for (int j = 0; j < b; j++){
146.                 cout << "Baris [" << i+1 <<"] Kolom [" << j+1 << " ] : ";
147.                 cin >> x[i][j];
148.             }
149.         }
150.     }
151.
152.     void printMatriks(matriks x, int a, int b){
153.         for (int i = 0; i < a; i++){
154.             cout << "[ ";
155.             for (int j = 0; j < b; j++){
156.                 cout << x[i][j] << " ";
157.             }
158.             cout << "]" << endl;
159.         }
160.     }
161.
162.     void penambahan(matriks hasilTambah, matriks x, matriks y, int a, int b){
163.         cout << "\n      =\n\n";
164.         for (int i = 0; i < a; i++){
165.             cout << "[ ";
166.             for (int j = 0; j < b; j++){
167.                 hasilTambah[i][j] = x[i][j] + y[i][j];
168.                 cout << hasilTambah[i][j] << " ";
169.             }
170.             cout << "]" << endl;
171.         }
172.     }
173.
174.     void perkalian(matriks hasilKali, matriks x, matriks y,
175. int a, int b, int c, int d){
176.         int temp=0;
177.         for(int i = 0; i < a; i++){
178.             for(int j = 0; j < d; j++){
179.                 hasilKali[i][j] = 0;
180.                 for(int k = 0; k < c; k++){
181.                     temp = x[i][k] * y[k][j];
182.                     hasilKali[i][j] = hasilKali[i][j] + temp;
183.                 }
184.             }
185.         }
186.         cout << "\n      =\n\n";
187.         for (int i = 0; i < a; i++){
188.             cout << "[ ";
189.             for (int j = 0; j < d; j++){
190.                 cout << hasilKali[i][j] << " ";
191.             }

```

```

192.             cout << "]" << endl;
193.         }
194.     }
195.
196.     void transpose(matriks x, int a, int b){
197.         for (int i = 0; i < a; i++){
198.             cout << "[ ";
199.             for (int j = 0; j < b; j++){
200.                 cout << x[j][i] << " ";
201.             }
202.             cout << "]" << endl;
203.         }
204.     }

```

Hasil Program Tugas 1 :

```

D:\Dokumen\UNPAD\Akademik\Algoritma & Pemrograman\Coding
\Praktikum 8> tugas1
=====
-----MENU-----
=====
1. Penjumlahan Matriks
2. Perkalian Matriks
3. Transpose Matriks
4. Selesai
-----
Pilih Menu [1-4] : 1

=====
-Penjumlahan Matriks-
=====
Banyak Baris    : 2
Banyak Kolom    : 2

== Matriks 1 ==
Baris [1] Kolom [1] : 1
Baris [1] Kolom [2] : 2
Baris [2] Kolom [1] : 3
Baris [2] Kolom [2] : 4

== Matriks 2 ==
Baris [1] Kolom [1] : 1
Baris [1] Kolom [2] : 2
Baris [2] Kolom [1] : 3
Baris [2] Kolom [2] : 4

[ 1 2 ]
[ 3 4 ]

+

[ 1 2 ]
[ 3 4 ]

=

[ 2 4 ]
[ 6 8 ]

```

```

=====
-----MENU-----
=====
1. Penjumlahan Matriks
2. Perkalian Matriks
3. Transpose Matriks
4. Selesai
-----
Pilih Menu [1-4] : 2

=====
-Perkalian Matriks-
=====
Syarat: Banyak Kolom Matriks 1 = Banyak Baris Matriks 2

== Matriks 1 ==
Banyak Baris    : 2
Banyak Kolom     : 3

== Matriks 2 ==
Banyak Baris    : 3
Banyak Kolom     : 3

--Input Elemen Matriks--

== Matriks 1 ==
Baris [1] Kolom [1] : 2
Baris [1] Kolom [2] : 1
Baris [1] Kolom [3] : 2
Baris [2] Kolom [1] : 1
Baris [2] Kolom [2] : 2
Baris [2] Kolom [3] : 1

== Matriks 2 ==
Baris [1] Kolom [1] : 2
Baris [1] Kolom [2] : 1
Baris [1] Kolom [3] : 4
Baris [2] Kolom [1] : 1
Baris [2] Kolom [2] : 2
Baris [2] Kolom [3] : 3
Baris [3] Kolom [1] : 2
Baris [3] Kolom [2] : 2
Baris [3] Kolom [3] : 1

[ 2  1  2 ]
[ 1  2  1 ]

      *

[ 2  1  4 ]
[ 1  2  3 ]
[ 2  2  1 ]

      =

[ 9  8  13 ]
[ 6  7  11 ]

```

```

=====
-----MENU-----
=====
1. Penjumlahan Matriks
2. Perkalian Matriks
3. Transpose Matriks
4. Selesai
-----
Pilih Menu [1-4] : 3

=====
-Transpose Matriks-
=====
Banyak Baris    : 3
Banyak Kolom    : 2

--Input Elemen Matriks--

Baris [1] Kolom [1] : 1
Baris [1] Kolom [2] : 2
Baris [2] Kolom [1] : 3
Baris [2] Kolom [2] : 4
Baris [3] Kolom [1] : 5
Baris [3] Kolom [2] : 6

== Matriks ==
[ 1 2 ]
[ 3 4 ]
[ 5 6 ]

== Hasil Transpose ==
[ 1 3 5 ]
[ 2 4 6 ]

=====
-----MENU-----
=====
1. Penjumlahan Matriks
2. Perkalian Matriks
3. Transpose Matriks
4. Selesai
-----
Pilih Menu [1-4] : 4

=====
-PROGRAM SELESAI-
=====

```

4.2. Tugas 2

```
1.  /*
2.  Nama      : Wafi Fahrurzzaman
3.  NPM       : 140810200009
4.  tanggal  : 4 November 2020
5.  deskripsi: -Tugas 2-
6.  */
7.
8.  #include <iostream>
9.  using namespace std;
10.
11. int data[10];
12. int n = sizeof(data)/sizeof(data[0]);
13. void header(string isi);
14. void menu(int& pilihan);
15. void inputData(int data[], int& n);
16. void printData(int data[], int n);
17. void insertion(int data[], int n);
18. void selection(int data[], int n);
19. void merge(int data[], int kecil, int tengah, int besar);
20. void mergeSort(int data[], int kecil, int besar);
21.
22. int main(){
23.     int pilihan = 0;
24.     while (pilihan != 4){
25.         menu(pilihan);
26.         switch(pilihan){
27.             case 1 :
28.                 cout << endl;
29.                 header ("-INSERTION SORT-");
30.                 inputData(data, n);
31.                 cout << "\n== SEBELUM INSERTION SORT ==\n";
32.                 printData(data, n); cout << "\n";
33.                 insertion(data, n);
34.                 cout << "\n\n== HASIL INSERTION SORT ==\n";
35.                 printData(data, n); cout << "\n\n";
36.                 cout << endl;
37.                 break;
38.
39.             case 2 :
40.                 cout << endl;
41.                 header ("-SELECTION SORT-");
42.                 inputData(data, n);
43.                 cout << "\n== SEBELUM SELECTION SORT ==\n";
44.                 printData(data, n); cout << "\n";
45.                 selection(data, n);
46.                 cout << "\n== HASIL SELECTION SORT ==\n";
```



```

47.         printData(data, n); cout << "\n\n";
48.         break;
49.
50.     case 3 :
51.         cout << endl;
52.         header ("-MERGE SORT-");
53.         inputData(data, n);
54.         cout << "\n\n== SEBELUM MERGE SORT ==\n";
55.         printData(data, n); cout << "\n";
56.         cout << "\n== PROSES MERGE SORT ==\n";
57.         printData(data, n);
58.         mergeSort(data, 0, n - 1); cout << "\n";
59.         cout << "\n== HASIL MERGE SORT ==\n";
60.         printData(data, n); cout << "\n\n";
61.         break;
62.
63.     case 4 :
64.         cout << endl;
65.         header ("-PROGRAM SELESAI-");
66.         cout << endl;
67.         break;
68.
69.     default:
70.         system("cls");
71.         cout << "\nMenu tidak ada. Cek kembali daftar Menu!\n";
72.     }
73. }
74. }
75.
76. void header(string isi){
77.     for (int i=1; i<= isi.length(); i++){
78.         cout << "=";
79.         if (i == isi.length()){
80.             cout << endl;
81.         }
82.     }
83.     cout << isi << endl;
84.     for (int i=1; i <= isi.length(); i++){
85.         cout << "=";
86.
87.         if (i == isi.length()){
88.             cout << endl;
89.         }
90.     }
91. }
92.
93. void menu(int& pilihan){
94.     header("-----MENU-----");
95.     cout << "1. Insertion Sort\n";

```

```

96.     cout << "2. Selection Sort\n";
97.     cout << "3. Merge Sort\n";
98.     cout << "4. Selesai\n";
99.     cout << "-----\n";
100.    cout << "Pilih Menu [1-4] : ";
101.    cin >> pilihan;
102.    }
103.
104.    void inputData(int data[], int& n){
105.        cout << "Input Banyak Data : "; cin >> n;
106.        for (int i = 0; i < n; i++){
107.            cout << "Data Ke-" << i+1 << " : "; cin >> data[i];
108.        }
109.    }
110.
111.    void printData(int data[], int n){
112.        cout << endl;
113.        for (int i = 0; i < n; i++){
114.            cout << data[i] << " ";
115.        }
116.    }
117.
118.    void insertion(int data[], int n){
119.        int temp, j;
120.        cout << "\n== PROSES INSERTION SORT ==\n";
121.        for(int i = 0; i < n; i++){
122.            temp = data[i];
123.            j = i-1;
124.            while((temp < data[j]) && (j >= 0)){
125.                data[j+1]= data[j];
126.                j = j-1;
127.            }
128.            data[j+1]=temp;
129.            printData(data, n);
130.        }
131.    }
132.
133.    void selection(int data[], int n){
134.        int iMin, temp;
135.        cout << "\n== PROSES SELECTION SORT ==\n";
136.        for(int i = 0; i < n-1; i++){
137.            iMin = i;
138.            printData(data, n);
139.            for(int j=i+1; j < n; j++){
140.                if(data[j] <= data[iMin]){
141.                    iMin = j;
142.                }
143.            }
144.

```

```

145.         if (iMin != i) {
146.             temp = data[i];
147.             data[i] = data[iMin];
148.             data[iMin] = temp;
149.         }
150.     }
151.     cout << endl;
152. }
153.
154. void merge(int data[], int kecil, int tengah, int besar){
155.     int temp[besar + 1];
156.     int i = kecil;
157.     int j = tengah + 1;
158.     int k = 0;
159.     while (i <= tengah && j <= besar)
160.     {
161.         if (data[i] <= data[j])
162.             temp[k++] = data[i++];
163.         else
164.             temp[k++] = data[j++];
165.     }
166.     while (i <= tengah)
167.         temp[k++] = data[i++];
168.     while (j <= besar)
169.         temp[k++] = data[j++];
170.     k--;
171.     while (k >= 0)
172.     {
173.         data[k + kecil] = temp[k];
174.         k--;
175.     }
176. }
177.
178. void mergeSort(int data[], int kecil, int besar){
179.     int tengah;
180.     if (kecil < besar)
181.     {
182.         tengah = (kecil + besar) / 2;
183.         mergeSort(data, kecil, tengah);
184.         mergeSort(data, tengah + 1, besar);
185.         merge(data, kecil, tengah, besar);
186.         printData(data, n);
187.     }
188. }

```

Hasil Program Tugas 2 :

```
C:\Wind...  —  □  ×

D:\Dokumen\UNPAD\Akademik\Algoritma & Pemrograman\Coding\Praktikum 8>
tugas2
=====
-----MENU-----
=====
1. Insertion Sort
2. Selection Sort
3. Merge Sort
4. Selesai
-----
Pilih Menu [1-4] : 1

=====
-INSERTION SORT-
=====
Input Banyak Data : 10
Data Ke-1 : 10
Data Ke-2 : 7
Data Ke-3 : 5
Data Ke-4 : 6
Data Ke-5 : 8
Data Ke-6 : 9
Data Ke-7 : 2
Data Ke-8 : 3
Data Ke-9 : 1
Data Ke-10 : 4

== SEBELUM INSERTION SORT ==

10 7 5 6 8 9 2 3 1 4

== PROSES INSERTION SORT ==

10 7 5 6 8 9 2 3 1 4
7 10 5 6 8 9 2 3 1 4
5 7 10 6 8 9 2 3 1 4
5 6 7 10 8 9 2 3 1 4
5 6 7 8 10 9 2 3 1 4
5 6 7 8 9 10 2 3 1 4
2 5 6 7 8 9 10 3 1 4
2 3 5 6 7 8 9 10 1 4
1 2 3 5 6 7 8 9 10 4
1 2 3 4 5 6 7 8 9 10

== HASIL INSERTION SORT ==

1 2 3 4 5 6 7 8 9 10
```

```
=====
-----MENU-----
=====
1. Insertion Sort
2. Selection Sort
3. Merge Sort
4. Selesai
-----
Pilih Menu [1-4] : 2

=====
-SELECTION SORT-
=====
Input Banyak Data : 10
Data Ke-1 : 10
Data Ke-2 : 7
Data Ke-3 : 5
Data Ke-4 : 6
Data Ke-5 : 8
Data Ke-6 : 9
Data Ke-7 : 2
Data Ke-8 : 3
Data Ke-9 : 1
Data Ke-10 : 4

== SEBELUM SELECTION SORT ==

10 7 5 6 8 9 2 3 1 4

== PROSES SELECTION SORT ==

10 7 5 6 8 9 2 3 1 4
1 7 5 6 8 9 2 3 10 4
1 2 5 6 8 9 7 3 10 4
1 2 3 6 8 9 7 5 10 4
1 2 3 4 8 9 7 5 10 6
1 2 3 4 5 9 7 8 10 6
1 2 3 4 5 6 7 8 10 9
1 2 3 4 5 6 7 8 10 9
1 2 3 4 5 6 7 8 10 9

== HASIL SELECTION SORT ==

1 2 3 4 5 6 7 8 9 10
```

```
=====
-----MENU-----
=====
1. Insertion Sort
2. Selection Sort
3. Merge Sort
4. Selesai
-----
Pilih Menu [1-4] : 3

=====
-MERGE SORT-
=====
Input Banyak Data : 10
Data Ke-1 : 10
Data Ke-2 : 7
Data Ke-3 : 5
Data Ke-4 : 6
Data Ke-5 : 8
Data Ke-6 : 9
Data Ke-7 : 2
Data Ke-8 : 3
Data Ke-9 : 1
Data Ke-10 : 4

== SEBELUM MERGE SORT ==

10 7 5 6 8 9 2 3 1 4

== PROSES MERGE SORT ==

10 7 5 6 8 9 2 3 1 4
7 10 5 6 8 9 2 3 1 4
5 7 10 6 8 9 2 3 1 4
5 7 10 6 8 9 2 3 1 4
5 6 7 8 10 9 2 3 1 4
5 6 7 8 10 2 9 3 1 4
5 6 7 8 10 2 3 9 1 4
5 6 7 8 10 2 3 9 1 4
5 6 7 8 10 1 2 3 4 9
1 2 3 4 5 6 7 8 9 10

== HASIL MERGE SORT ==

1 2 3 4 5 6 7 8 9 10
```

5. Kesimpulan

Pada praktikum Algoritma dan Pemrograman ke 8, telah mempelajari dan mengimplementasi-kan materi yang diberikan yaitu

1. Array
 - a. Deklarasi secara langsung
 - b. Deklarasi secara tidak langsung
 - c. Deklarasi sekaligus inisialisasi
2. Sort (Sorting)
 - a. Bubble Sort
 - b. Insertion Sort
 - c. Selection Sort
 - d. Merge Sort

Selain mempelajari materi-materi tersebut, juga telah mengimplementasikan dengan cara membuat program sederhana. Pada praktikum ini membuat program sederhana yaitu membuat Program Modular Matriks 2D. Program tersebut dapat meng-input dan print matriks, selain itu juga dapat menjumlah, mengali 2 matriks, serta dapat transpose suatu matriks. User dapat menentukan panjang dan lebar matriks sendiri. Juga, program dijalankan dengan menu.

Selain itu, mengeksplorasi bagaimana cara sorting di c++ kemudian membuat program yang memuat Insertion Sort, Selection Sort, dan Merge Sort.