

WORKSHEET – MODUL 6

Identitas mahasiswa:

Nama : Ilham Yusuf Wi'am
NIM : 24/539979/TK/59890
Sesi praktikum: Senin B3

Intruksi: Kerjakan semua percobaan dan *check points* yang ada pada Modul 6 dan tuliskan jawabannya pada worksheet ini.

Percobaan 1: ARRAY STATIS SATU DIMENSI

1. Apakah keluaran dari program di atas tersebut? Jelaskan

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int larik[5] = {10, 20, 30, 40, 50};
6      int i;
7      cout<<"Elemen larik\t\t"<<"Nilai elemen larik\n";
8      for(i=0; i<5; i++)
9      {
10         cout<<"larik["<<i<<"]\t\t\t"<<larik[i]<<"\n";
11     }
12     return 0;
13 }
```

Output :

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Elemen larik          Nilai elemen larik
larik[0]              10
larik[1]              20
larik[2]              30
larik[3]              40
larik[4]              50
```

Penjelasan :

Kode ini bertujuan untuk menampilkan nilai dari elemen array yang ditunjukkan dari indeksnya, index 0 atau elemen pertama bernilai 10, dan seterusnya.

Percobaan 2: ARRAY STATIS SATU DIMENSI

1. Berapa indeks array A yang paling besar ?

Indeks array A paling besar adalah 4

2. Berapa ukuran tipe data dari elemen array A ?

Ukuran tipe data dari elemen array A adalah 4 byte

3. Berapa bertambah ukuran memori yang menyimpan tiap nilai elemen dari array A ?

Penambahan setiap menyimpan nilai adalah 4 byte

Percobaan 3: ARRAY STATIS SATU DIMENSI UNTUK DATA *STRUCTURE*

1. Apakah keluaran dari program di atas?

```
1  #include <iostream>
2  using namespace std;
3
4  typedef struct Person
5  {
6      string name;
7      int NIM;
8      int age;
9  }Student;
10
11
12 int main () {
13     int i, n;
14     cout << "Masukkan jumlah data mahasiswa :";
15     cin >> n;
16     cout << endl;
17     Student s_array[n];
18     cout << "Masukkan Data Mahasiswa \n";
19     for (i=0; i<n; i++){
20         cout << "Masukkan Nama mahasiswa :\n";
21         cin >> s_array[i].name;
22         cin.ignore();
23         cout << "Masukkan NIM mahasiswa :\n";
24         cin >> s_array[i].NIM;
25         cout << "Masukkan Umur mahasiswa :\n";
26         cin >> s_array[i].age;
27     }
28
29     cout << "\nDaftar Mahasiswa\n";
30     cout << "\nNo. \t\tNama\t\tNIM\t\tUmur\n";
31     for(i=0;i<n;i++){
32         cout << i+1 << "\t\t"<<s_array[i].name << "\t\t\t" << s_array[i].NIM << "\t\t" << s_array[i].age << endl;
33     }
34 }
```

Output :

```
Masukkan jumlah data mahasiswa :2
Masukkan Data Mahasiswa
Masukkan Nama mahasiswa :
Ilham Yusuf Wi'am
Masukkan NIM mahasiswa :
Masukkan Umur mahasiswa :
Masukkan Nama mahasiswa :
Masukkan NIM mahasiswa :
Masukkan Umur mahasiswa :

Daftar Mahasiswa

No.      Nama      NIM      Umur
1        Ilham      0        10
2        0        0
```

Ada kesalahan

Output yang benar :

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Masukkan jumlah data mahasiswa :2
```

```
Masukkan Data Mahasiswa
Masukkan Nama mahasiswa :
Iam
Masukkan NIM mahasiswa :
24/539979/TK/59890
Masukkan Umur mahasiswa :
19
Masukkan Nama mahasiswa :
V
Masukkan NIM mahasiswa :
24/xxxxxx/EK/yyyyy
Masukkan Umur mahasiswa :
18
```

Daftar Mahasiswa

No.	Nama	NIM	Umur
1	Iam	24/539979/TK/59890	19
2	V	24/xxxxxx/EK/yyyyy	18

- Apakah ada kesalahan ? Jika ada silahkan diperbaiki sampai menghasilkan keluaran yang sesuai

Ada kesalahan,

Kode perbaikan :

```
1  #include <iostream>
2  using namespace std;
3
4  typedef struct Person
5  {
6      string name, NIM;
7      int age;
8  }Student;
9
10
11 int main () {
12     int i, n;
13     cout << "Masukkan jumlah data mahasiswa :";
14     cin >> n;
15     cout << endl;
16     Student s_array[n];
17     cout << "Masukkan Data Mahasiswa \n";
18     for (i=0; i<n; i++){
19         cout << "Masukkan Nama mahasiswa :\n";
20         cin >> s_array[i].name;
21         cout << "Masukkan NIM mahasiswa :\n";
22         cin >> s_array[i].NIM;
23         cin.ignore();
24         cout << "Masukkan Umur mahasiswa :\n";
25         cin >> s_array[i].age;
26     }
27
28     cout << "\nDaftar Mahasiswa\n";
29     cout << "\nNo. \t\tNama\t\tNIM\t\tUmur\n";
30     for(i=0;i<n;i++){
31         cout << i+1 << "\t\t" << s_array[i].name << "\t\t" << s_array[i].NIM << "\t\t" << s_array[i].age << endl;
32     }
33 }
```

Output :

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Masukkan jumlah data mahasiswa :2

Masukkan Data Mahasiswa
Masukkan Nama mahasiswa :
Ilham
Masukkan NIM mahasiswa :
539979
Masukkan Umur mahasiswa :
19
Masukkan Nama mahasiswa :
Wikan
Masukkan NIM mahasiswa :
52980
Masukkan Umur mahasiswa :
20

Daftar Mahasiswa
```

No.	Nama	NIM	Umur
1	Ilham	539979	19
2	Wikan	52980	20

Jika NIMnya berstring 24/xxxxxx/TK/yyyyy

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Masukkan jumlah data mahasiswa :2

Masukkan Data Mahasiswa
Masukkan Nama mahasiswa :
Iam
Masukkan NIM mahasiswa :
24/539979/TK/59890
Masukkan Umur mahasiswa :
19
Masukkan Nama mahasiswa :
V
Masukkan NIM mahasiswa :
24/xxxxxx/EK/yyyyy
Masukkan Umur mahasiswa :
18

Daftar Mahasiswa
```

No.	Nama	NIM	Umur
1	Iam	24/539979/TK/59890	19
2	V	24/xxxxxx/EK/yyyyy	18

Percobaan 4: ARRAY STATIS DUA DIMENSI

1. Apakah keluaran dari program di atas ?

```
1  #include <iostream>
2  using namespace std;
3
4  int main () {
5      int A[5][5], i, j, k, l; // deklarasi array A bertipe integer dan memiliki ukuran baris 5 dan kolom 5
6      // inisialisasi indeks dari array A dimulai dari indeks [0][0]
7      for ( int i = 0; i < 5; i++ ){
8          for ( int j = 0; j < 5; j++ )
9              // set nilai elemen dimulai dari array A[0][0], baris ke 0 akan diset ke nilai 100 selanjutnya kenaikan baris akan bertambah 1
10                 A[i][j] = i+100;
11      }
12
13      cout << "Indeks\t\t" << "Nilai Elemen\t" << "Alamat"<<endl;
14      // menampilkan indeks dan nilai dari tiap elemen array A serta lokasi alamat memorinya
15      for ( int k = 0; k < 5; k++ ){
16          for ( int l = 0; l < 5; l++ )
17              cout << "[" <<k<<"[" <<l<<"[" <<"\t\t"<<A[k][l]<<"\t\t\t"<<6A[k][l]<< endl;
18      }
19  }
```

Output :

```
Praktikan@CDSR3-33 MSYS /c/iam
```

```
$ ./a.exe
```

Indeks	Nilai Elemen	Alamat
[0][0]	100	0x7ffffcb0
[0][1]	100	0x7ffffcb4
[0][2]	100	0x7ffffcb8
[0][3]	100	0x7ffffcbcc
[0][4]	100	0x7ffffcbd0
[1][0]	101	0x7ffffcbd4
[1][1]	101	0x7ffffcbd8
[1][2]	101	0x7ffffcbdc
[1][3]	101	0x7ffffcbe0
[1][4]	101	0x7ffffcbe4
[2][0]	102	0x7ffffcbe8
[2][1]	102	0x7ffffcbec
[2][2]	102	0x7ffffcbf0
[2][3]	102	0x7ffffcbf4
[2][4]	102	0x7ffffcbf8
[3][0]	103	0x7ffffcbfc
[3][1]	103	0x7ffffcc00
[3][2]	103	0x7ffffcc04
[3][3]	103	0x7ffffcc08
[3][4]	103	0x7ffffcc0c
[4][0]	104	0x7ffffcc10
[4][1]	104	0x7ffffcc14
[4][2]	104	0x7ffffcc18
[4][3]	104	0x7ffffcc1c
[4][4]	104	0x7ffffcc20

2. Mengapa diperlukan dua buah perintah “for” untuk iterasinya ?

diperlukan dua buah perintah for dalam iterasinya karena array di atas merupakan array dua dimensi yang terdiri dari row dan column. For pertama untuk mengiterasi row dan for kedua untuk mengiterasi column.

3. Modifikasi program di atas dengan mengganti perintah iterasi “for” dengan perintah “while”

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int A[5][5], i, j, k, l; // Declaration of array A and indices
6      i = 0; // Initialization of row index
7
8      // Populating array using nested while loops
9      while (i < 5) {
10         j = 0; // Initialize column index
11         while (j < 5) {
12             A[i][j] = i + 100; // Assign values to array elements
13             j++;
14         }
15         i++;
16     }
17
18     cout << "Indeks\t\t" << "Nilai Elemen\t" << "Alamat" << endl;
19
20     // Displaying array data using nested while loops
21     k = 0; // Initialize row index for display
22     while (k < 5) {
23         l = 0; // Initialize column index for display
24         while (l < 5) {
25             cout << "[" << k << "]" << l << "]"
26                 << "\t\t" << A[k][l]
27                 << "\t\t\t" << &A[k][l] << endl;
28             l++;
29         }
30         k++;
31     }
32
33     return 0;
34 }

```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Indeks      Nilai Elemen  Alamat
[0][0]      100           0x7ffffc00
[0][1]      100           0x7ffffc04
[0][2]      100           0x7ffffc08
[0][3]      100           0x7ffffc0c
[0][4]      100           0x7ffffc10
[1][0]      101           0x7ffffc14
[1][1]      101           0x7ffffc18
[1][2]      101           0x7ffffc1c
[1][3]      101           0x7ffffc20
[1][4]      101           0x7ffffc24
[2][0]      102           0x7ffffc28
[2][1]      102           0x7ffffc2c
[2][2]      102           0x7ffffc30
[2][3]      102           0x7ffffc34
[2][4]      102           0x7ffffc38
[3][0]      103           0x7ffffc3c
[3][1]      103           0x7ffffc40
[3][2]      103           0x7ffffc44
[3][3]      103           0x7ffffc48
[3][4]      103           0x7ffffc4c
[4][0]      104           0x7ffffc50
[4][1]      104           0x7ffffc54
[4][2]      104           0x7ffffc58
[4][3]      104           0x7ffffc5c
[4][4]      104           0x7ffffc60

```

- Modifikasi program di atas dengan mengganti perintah iterasi “for” dengan perintah “do while”

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int A[5][5], i, j, k, l; // Declaration of array A and indices
6      i = 0; // Start index for row
7
8      // Populating the array using nested do-while loops
9      do {
10         j = 0; // Start index for column
11         do {
12             A[i][j] = i + 100; // Assign values to array elements
13             j++;
14         } while (j < 5);
15         i++;
16     } while (i < 5);
17
18     cout << "Indeks\t\t" << "Nilai Elemen\t" << "Alamat" << endl;
19
20     // Displaying the array data using nested do-while loops
21     k = 0; // Start index for row in display
22     do {
23         l = 0; // Start index for column in display
24         do {
25             cout << "[" << k << "]"[" << l << "]"
26                 << "\t\t" << A[k][l]
27                 << "\t\t\t\t" << &A[k][l] << endl;
28             l++;
29         } while (l < 5);
30         k++;
31     } while (k < 5);
32
33     return 0;
34 }

```

Praktikan@CDSR3-33 MSYS /c/iam

\$./a.exe

Indeks	Nilai Elemen	Alamat
[0][0]	100	0x7ffffcb0
[0][1]	100	0x7ffffcb4
[0][2]	100	0x7ffffcb8
[0][3]	100	0x7ffffcbc
[0][4]	100	0x7ffffcbd
[1][0]	101	0x7ffffcbd4
[1][1]	101	0x7ffffcbd8
[1][2]	101	0x7ffffcbd0
[1][3]	101	0x7ffffcbe0
[1][4]	101	0x7ffffcbe4
[2][0]	102	0x7ffffcbe8
[2][1]	102	0x7ffffcbec
[2][2]	102	0x7ffffcbf0
[2][3]	102	0x7ffffcbf4
[2][4]	102	0x7ffffcbf8
[3][0]	103	0x7ffffcbfc
[3][1]	103	0x7ffffcc00
[3][2]	103	0x7ffffcc04
[3][3]	103	0x7ffffcc08
[3][4]	103	0x7ffffcc0c
[4][0]	104	0x7ffffcc10
[4][1]	104	0x7ffffcc14
[4][2]	104	0x7ffffcc18
[4][3]	104	0x7ffffcc1c
[4][4]	104	0x7ffffcc20

Percobaan 5: ARRAY STATIS DUA DIMENSI

1. Apakah keluaran dari program di atas ? Jelaskan

```
1  #include<iostream>
2  using namespace std;
3
4  int main(){
5      int s[3][3];
6      int i, j;
7      cout<<"\nMasukkan sks Mahasiswa untuk 3 Angkatan yang berbeda \n";
8      for(i=0;i<3;i++){
9          cout<<"\n Angkatan 202"<<i<<"\t";
10         for(j=0;j<3;j++){
11             cout<<"\nskks Mahasiswa ["<<i<<"]["<<j<<"]=" ";
12             cin>>s[i][j];
13         }
14     }
15     cout<<endl;
16     cout<<"\nTampilkan data sks Mahasiswa 3 Angkatan\n";
17     cout<<"\n Angkatan\t\t"<<"sks";
18     for(i=0;i<3;i++){
19         cout<<"\n 202"<<i<<"\t";
20         for(j=0;j<3;j++){
21             cout<<"\t"<<s[i][j];
22         }
23     }
24     cout<<endl;
25 }
```

Praktikan@CDSR3-33 MSYS /c/iam
\$./a.exe

Masukkan sks Mahasiswa untuk 3 Angkatan yang berbeda

Angkatan 2020
sks Mahasiswa [0][0]= 24
sks Mahasiswa [0][1]= 21
sks Mahasiswa [0][2]= 20

Angkatan 2021
sks Mahasiswa [1][0]= 23
sks Mahasiswa [1][1]= 24
sks Mahasiswa [1][2]= 21

Angkatan 2022
sks Mahasiswa [2][0]= 23
sks Mahasiswa [2][1]= 22
sks Mahasiswa [2][2]= 19

Tampilkan data sks Mahasiswa 3 Angkatan

Angkatan	sks		
2020	24	21	20
2021	23	24	21
2022	23	22	19

2. Modifikasi program di atas sehingga dapat mengeluarkan data sks mahasiswa untuk 6 angkatan dimulai dari angkatan 1990 dan tiap angkatan ada 4 data sks mahasiswa.

```
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5      int s[6][4]; // Mengubah ukuran array menjadi 6x4
6      int i, j;
7      cout << "\nMasukkan SKS Mahasiswa untuk 6 Angkatan dimulai dari angkatan 1990\n";
8      for(i = 0; i < 6; i++) {
9          cout << "\n Angkatan " << 1990 + i << "\t";
10         for(j = 0; j < 4; j++) {
11             cout << "\nSKS Mahasiswa [" << i << "][" << j << "] = ";
12             cin >> s[i][j];
13         }
14     }
15     cout << endl;
16     cout << "\nTampilkan data SKS Mahasiswa untuk 6 Angkatan\n";
17     cout << "\n Angkatan\t\t" << "SKS";
18     for(i = 0; i < 6; i++) {
19         cout << "\n " << 1990 + i << "\t";
20         for(j = 0; j < 4; j++) {
21             cout << "\t" << s[i][j];
22         }
23         cout << endl;
24     }
25 }
```

Praktikan@CDSR3-33 MSYS /c/iam
\$./a.exe

Masukkan SKS Mahasiswa untuk 6 Angkatan dimulai dari angkatan 1990

Angkatan 1990
SKS Mahasiswa [0][0] = 3
SKS Mahasiswa [0][1] = 4
SKS Mahasiswa [0][2] = 7
SKS Mahasiswa [0][3] = 9

Angkatan 1991
SKS Mahasiswa [1][0] = 20
SKS Mahasiswa [1][1] = 24
SKS Mahasiswa [1][2] = 23
SKS Mahasiswa [1][3] = 21

Angkatan 1992
SKS Mahasiswa [2][0] = 16
SKS Mahasiswa [2][1] = 17
SKS Mahasiswa [2][2] = 19
SKS Mahasiswa [2][3] = 18

Angkatan 1993
SKS Mahasiswa [3][0] = 23
SKS Mahasiswa [3][1] = 24
SKS Mahasiswa [3][2] = 21
SKS Mahasiswa [3][3] = 24

Angkatan 1994
SKS Mahasiswa [4][0] = 17
SKS Mahasiswa [4][1] = 15
SKS Mahasiswa [4][2] = 14
SKS Mahasiswa [4][3] = 13

Angkatan 1995
SKS Mahasiswa [5][0] = 24
SKS Mahasiswa [5][1] = 24
SKS Mahasiswa [5][2] = 24
SKS Mahasiswa [5][3] = 24

Tampilkan data SKS Mahasiswa untuk 6 Angkatan

Angkatan	SKS			
1990	3	4	7	9
1991	20	24	23	21
1992	16	17	19	18
1993	23	24	21	24
1994	17	15	14	13
1995	24	24	24	24

3. Gantilah perintah “for” dengan “while” untuk perintah pada poin kedua

```
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5      int s[6][4]; // Menggunakan array untuk 6 angkatan dan 4 data SKS per angkatan
6      int i = 0, j;
7
8      cout << "\nMasukkan SKS Mahasiswa untuk 6 Angkatan dimulai dari angkatan 1990\n";
9
10     // Input data menggunakan while loop
11     while (i < 6) {
12         cout << "\n Angkatan " << 1990 + i << "\t";
13         j = 0; // Reset nilai j untuk setiap angkatan
14         while (j < 4) {
15             cout << "\nSKS Mahasiswa [" << i << "][" << j << "] = ";
16             cin >> s[i][j];
17             j++;
18         }
19         i++;
20     }
21
22     cout << endl;
23     cout << "\nTampilkan data SKS Mahasiswa untuk 6 Angkatan\n";
24     cout << "\n Angkatan\t\t" << "SKS";
25
26     i = 0; // Reset nilai i untuk output
27     // Output data menggunakan while loop
28     while (i < 6) {
29         cout << "\n " << 1990 + i << "\t";
30         j = 0; // Reset nilai j untuk setiap angkatan
31         while (j < 4) {
32             cout << "\t" << s[i][j];
33             j++;
34         }
35         cout << endl;
36         i++;
37     }
38 }
```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe

Masukkan SKS Mahasiswa untuk 6 Angkatan dimulai dari angkatan 1990

Angkatan 1990
SKS Mahasiswa [0][0] = 1
SKS Mahasiswa [0][1] = 2
SKS Mahasiswa [0][2] = 3
SKS Mahasiswa [0][3] = 4

Angkatan 1991
SKS Mahasiswa [1][0] = 2
SKS Mahasiswa [1][1] = 3
SKS Mahasiswa [1][2] = 4
SKS Mahasiswa [1][3] = 5

Angkatan 1992
SKS Mahasiswa [2][0] = 3
SKS Mahasiswa [2][1] = 4
SKS Mahasiswa [2][2] = 5
SKS Mahasiswa [2][3] = 6

Angkatan 1993
SKS Mahasiswa [3][0] = 4
SKS Mahasiswa [3][1] = 5
SKS Mahasiswa [3][2] = 6
SKS Mahasiswa [3][3] = 7

Angkatan 1994
SKS Mahasiswa [4][0] = 5
SKS Mahasiswa [4][1] = 6
SKS Mahasiswa [4][2] = 7
SKS Mahasiswa [4][3] = 8

Angkatan 1995
SKS Mahasiswa [5][0] = 6
SKS Mahasiswa [5][1] = 7
SKS Mahasiswa [5][2] = 8
SKS Mahasiswa [5][3] = 9

```

Tampilkan data SKS Mahasiswa untuk 6 Angkatan

Angkatan	1	2	3	4
1990	1	2	3	4
1991	2	3	4	5
1992	3	4	5	6
1993	4	5	6	7
1994	5	6	7	8
1995	6	7	8	9

Percobaan 6: STRUCTURE DINAMIS

1. Apakah keluaran dari program di atas ?

```
1  #include <iostream>
2  using namespace std;
3
4  struct rectangular {
5      int length;
6      int width;
7  };
8  int main() {
9      rectangular *ptr, r;
10     int area;
11     ptr = &r;
12     cout << "Enter length: "<<endl;
13     cin >> (*ptr).length;
14     cout << "Enter width"<<endl;
15     cin >> (*ptr).width;
16     area = (*ptr).length * (*ptr).width;
17     cout << "Displaying information Rectangular" << endl;
18     cout << "Length = " << (*ptr).length <<" \tWidth " << (*ptr).width <<endl;
19     cout << "Area = " << area<<endl;
20     return 0;
21 }
```

```
Praktikan@CDSR3-33 MSYS /c/iam
$ g++ p6.cpp

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Enter length:
5
Enter width
10
Displaying information Rectangular
Length = 5      Width 10
Area = 50
```

2. Apa beda antara structure dinamis dan statis ?

struktur statis memiliki alokasi memori tetap yang ditentukan saat kompilasi, sehingga lebih mudah dikelola dan efisien dalam penggunaan memori. Sebaliknya, struktur dinamis memungkinkan alokasi memori saat runtime, memberikan fleksibilitas yang lebih tinggi, tetapi memerlukan pengelolaan memori manual untuk menghindari kebocoran. Struktur statis cocok untuk data yang ukurannya diketahui sebelumnya, sedangkan struktur dinamis lebih sesuai untuk kebutuhan yang berubah.

Percobaan 7: STRUCTURE DINAMIS

1. Jelaskan apa keluaran dari program tersebut

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i,n;
6      cout<<"Enter total number of elements:"<<"\n";
7      cin>>n;
8      int *a = new int(n);
9      cout<<"Enter "<<n<<" elements"<<endl;
10     for(i = 0;i<n;i++) {
11         cin>>a[i];
12     }
13     cout<<"Entered elements are: ";
14     for(i = 0;i<n;i++) {
15         cout<<a[i]<<" ";
16     }
17     cout<<endl;
18     delete (a);
19     return 0;
20 }
```

Praktikan@CDSR3-33 MSYS /c/iam
\$./a.exe
Enter total number of elements:
3
Enter 3 elements
1 2 3
Entered elements are: 1 2 3

2. Apa perbeda antara structure dinamis dan statis ?

struktur statis memiliki alokasi memori tetap yang ditentukan saat kompilasi, sehingga lebih mudah dikelola dan efisien dalam penggunaan memori. Sebaliknya, struktur dinamis memungkinkan alokasi memori saat runtime, memberikan fleksibilitas yang lebih tinggi, tetapi memerlukan pengelolaan memori manual untuk menghindari kebocoran. Struktur statis cocok untuk data yang ukurannya diketahui sebelumnya, sedangkan struktur dinamis lebih sesuai untuk kebutuhan yang berubah.

Percobaan 8: ARRAY DINAMIS

1. Jelaskan keluaran program di atas ?

```
1  #include <iostream>
2  using namespace std;
3
4  #define N 10
5  // Dynamically allocate memory for 1D Array in C++
6  int main(){
7  // dynamically allocate memory of size `N`
8      int* A = new int[N];
9  // assign values to the allocated memory
10     for (int i = 0; i < N; i++) {
11         A[i] = i + 1;
12     }
13     // print the 1D array
14     for (int i = 0; i < N; i++) {
15         cout << A[i] << " "; // or *(A + i)
16     }
17     // deallocate memory
18     delete[] A;
19     return 0;
20 }
```

Praktikan@CDSR3-33 MSYS /c/iam
\$./a.exe
1 2 3 4 5 6 7 8 9 10

2. Apa bedanya dengan array statis ?

Array dapat bersifat statis atau dinamis. Array statis akan menyimpan jumlah elemen data yang tetap, sesuai pada saat dideklarasikan. Sedangkan array dinamis jumlah elemennya tidak tetap dan dapat berkembang.

Check poin 1

Screenshot dan masukkan ke *worksheet* hasil kerja anda.

```
1.
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int arr[10];
5      int i;
6      int sum=0, avg=0;
7      cout<<"Masukkan 10 Elemen Array bertipe integer \n";
8      for(i=0; i<10; i++){
9          cin>>arr[i];
10         sum = sum + arr[i];
11     }
12     cout<<"\nElemen Array adalah : \n";
13     for(i=0; i<10; i++){
14         cout<<arr[i]<<" ";
15     }
16     cout<<"\n\nHasil penjumlahan dari semua elemen array adalah : "<<sum;
17     avg = sum/10;
18     cout<<"\nRerata elemen array adalah : "<<avg;
19     return 0;
20 }
```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Masukkan 10 Elemen Array bertipe integer
1
2
3
4
5
6
7
8
9
10

Elemen Array adalah :
1 2 3 4 5 6 7 8 9 10

Hasil penjumlahan dari semua elemen array adalah : 55
Rerata elemen array adalah : 5

```

2.

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  const int EMPLOYEE_COUNT = 3; // Ganti nama supaya tidak konflik
6
7  struct Employee {
8      string name;
9      int emp_no;
10     float salary;
11 };
12
13 int main() {
14     Employee emp[EMPLOYEE_COUNT];
15     cout << "Masukkan Data Pegawai\n";
16
17     for (int i = 0; i < EMPLOYEE_COUNT; i++) {
18         cout << " Pegawai Nomor " << i + 1 << endl;
19         cout << " Nama: ";
20         cin.ignore();
21         getline(cin, emp[i].name);
22         cout << " Kode: ";
23         cin >> emp[i].emp_no;
24         cout << " Gaji: ";
25         cin >> emp[i].salary;
26     }
27
28     cout << "\nData Pegawai\n";
29     cout << "No.\tNama\tKode\tGaji\n";
30
31     for (int i = 0; i < EMPLOYEE_COUNT; i++) {
32         cout << i + 1 << "\t"
33             << emp[i].name << "\t"
34             << emp[i].emp_no << "\t"
35             << emp[i].salary << endl;
36     }
37
38     return 0;
39 }
40

```

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Masukkan Data Pegawai
Pegawai Nomor 1
Nama: iam
Kode: 13
Gaji: 100
Pegawai Nomor 2
Nama: wik
Kode: 32
Gaji: 10
Pegawai Nomor 3
Nama: toni
Kode: 10
Gaji: 8
```

```
Data Pegawai
No.    Nama    Kode    Gaji
1      am     13     100
2      wik     32     10
3      toni    10     8
```

3.

```
1  #include <iostream>
2  using namespace std;
3
4  typedef int* IntPtr;
5
6  int main() {
7      IntPtr p;
8      int a[10];
9      int index;
10
11     for (index = 0; index < 10; index++)
12         a[index] = index;
13
14     p = a;
15
16     for (index = 0; index < 10; index++)
17         cout << p[index] << " ";
18     cout << endl;
19
20     for (index = 0; index < 10; index++)
21         p[index] = p[index] + 1;
22
23     for (index = 0; index < 10; index++)
24         cout << a[index] << " ";
25     cout << endl;
26
27     return 0;
28 }
29
```

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
```

Penjelasan :

Program ini mendeklarasikan array a berisi 10 elemen dan sebuah pointer p bertipe int*. Array a diisi dengan angka 0 hingga 9, lalu pointer p diarahkan ke array tersebut. Program

mencetak isi array melalui pointer p, kemudian menambahkan 1 ke setiap elemen array juga lewat pointer. Karena p menunjuk langsung ke array a, perubahan yang dilakukan melalui p akan mempengaruhi isi array a. Akhirnya, program mencetak isi array a yang sudah bertambah satu di setiap elemennya.

Check poin 2

Screenshot dan masukkan ke *worksheet* hasil kerja anda.

1.

Program tersebut awalnya mengurutkan lima angka yang sudah ditentukan menggunakan algoritma *insertion sort* secara menaik. Setelah dimodifikasi, program memungkinkan pengguna untuk memasukkan lima angka secara interaktif melalui input keyboard, kemudian mengurutkannya dari nilai terkecil ke terbesar dan menampilkannya kembali ke layar.

```
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5      int myarray[5]; // Array 5 elemen
6
7      cout << "Masukkan 5 angka:\n";
8      for (int i = 0; i < 5; i++) {
9          cout << "Elemen ke-" << i + 1 << ": ";
10         cin >> myarray[i];
11     }
12
13     cout << "\nInput list adalah:\n";
14     for (int i = 0; i < 5; i++) {
15         cout << myarray[i] << "\t";
16     }
17
18     // Proses insertion sort
19     for (int k = 1; k < 5; k++) {
20         int temp = myarray[k];
21         int j = k - 1;
22         while (j >= 0 && temp <= myarray[j]) {
23             myarray[j + 1] = myarray[j];
24             j = j - 1;
25         }
26         myarray[j + 1] = temp;
27     }
28
29     cout << "\nOutput list setelah diurutkan:\n";
30     for (int i = 0; i < 5; i++) {
31         cout << myarray[i] << "\t";
32     }
33     cout << endl;
34
35     return 0;
36 }
37
```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Masukkan 5 angka:
Elemen ke-1: 3
Elemen ke-2: 2
Elemen ke-3: 5
Elemen ke-4: 4
Elemen ke-5: 1

Input list adalah:
3      2      5      4      1
Output list setelah diurutkan:
1      2      3      4      5

```

2.

Selain mencari umur terbesar, kita juga menyimpan index siswa tertua (indexMax). Setelah loop selesai, kita gunakan vr[indexMax].name untuk menampilkan nama siswa yang

```

1  #include<iostream>
2  using namespace std;
3
4  // struct declaration outside main()
5  struct
6  {
7      int roll, age;
8      string name;
9  } vr[4];
10
11 int main()
12 {
13     int i, j, k, maxAge, indexMax;
14
15     for (i = 0; i < 4; i++)
16     {
17         cout << "Enter " << i + 1 << " Student Record\n";
18         cout << "Enter Roll no: ";
19         cin >> vr[i].roll;
20         cout << "Enter age: ";
21         cin >> vr[i].age;
22         cout << "Enter Name: ";
23         cin >> vr[i].name;
24     }
25
26     cout << endl;
27     for (j = 0; j < 4; j++)
28     {
29         cout << "\nDisplay " << j + 1 << " Student Record";
30         cout << "\nRoll no.: " << vr[j].roll;
31         cout << "\nAge: " << vr[j].age;
32         cout << "\nName: " << vr[j].name << endl;
33     }
34
35     maxAge = vr[0].age;
36     indexMax = 0; // Simpan index siswa dengan umur maksimum
37
38     for (k = 1; k < 4; k++)
39     {
40         if (vr[k].age > maxAge)
41         {
42             maxAge = vr[k].age;
43             indexMax = k; // Update index siswa tertua
44         }
45     }
46
47     cout << "\nThe oldest student's age is: " << maxAge << endl;
48     cout << "The oldest student is: " << vr[indexMax].name << endl;
49
50     return 0;
51 }
52

```

umurnya tertua.

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Enter 1 Student Record
Enter Roll no: 1
Enter age: 30
Enter Name: aan
Enter 2 Student Record
Enter Roll no: 2
Enter age: 25
Enter Name: ron
Enter 3 Student Record
Enter Roll no: 4
Enter age: 13
Enter Name: cyan
Enter 4 Student Record
Enter Roll no: 5
Enter age: 45
Enter Name: josh

Display 1 Student Record
Roll no.: 1
Age: 30
Name: aan

Display 2 Student Record
Roll no.: 2
Age: 25
Name: ron

Display 3 Student Record
Roll no.: 4
Age: 13
Name: cyan

Display 4 Student Record
Roll no.: 5
Age: 45
Name: josh

The oldest student's age is: 45
The oldest student is: josh
```

3.

```

1  #include<iostream>
2  using namespace std;
3
4  // declare functions
5  double add(double arr[], int size);
6  double avg(double sum, int n);
7
8  // main function
9  int main()
10 {
11     // declare variables
12     double numbers[3]; // Array untuk menampung tiga angka
13     double sum, average;
14
15     // take input from end-user
16     cout << "Enter three Numbers : ";
17     for (int i = 0; i < 3; i++) {
18         cin >> numbers[i]; // Input angka ke dalam array
19     }
20
21     // calculate sum value
22     sum = add(numbers, 3); // Menghitung jumlah angka dalam array
23
24     // calculate average value
25     average = avg(sum, 3); // Menghitung rata-rata
26
27     // display result
28     cout << "Sum = " << sum << endl;
29     cout << "Average = " << average << endl;
30
31     return 0;
32 }
33
34 // function to add the numbers in the array
35 double add(double arr[], int size)
36 {
37     double sum = 0;
38     for (int i = 0; i < size; i++) {
39         sum += arr[i]; // Menambahkan semua elemen array
40     }
41     return sum;
42 }
43
44 // function to calculate average of n number
45 double avg(double sum, int n)
46 {
47     return sum / n;
48 }
49

```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Enter three Numbers : 1
2
3
Sum = 6
Average = 2

```

4.

```

1  #include<iostream>
2  using namespace std;
3
4  // declare functions
5  double add(double arr[], int size);
6  double avg(double sum, int n);
7
8  // main function
9  int main()
10 {
11     // declare variables
12     double numbers[3]; // Array untuk menampung tiga angka
13     double sum, average;
14
15     // take input from end-user
16     cout << "Enter three Numbers : ";
17     for (int i = 0; i < 3; i++) {
18         cin >> numbers[i]; // Input angka ke dalam array
19     }
20
21     // calculate sum value
22     sum = add(numbers, 3); // Menghitung jumlah angka dalam array
23
24     // calculate average value
25     average = avg(sum, 3); // Menghitung rata-rata
26
27     // display result
28     cout << "Sum = " << sum << endl;
29     cout << "Average = " << average << endl;
30
31     return 0;
32 }
33
34 // function to add the numbers in the array
35 double add(double arr[], int size)
36 {
37     double sum = 0;
38     for (int i = 0; i < size; i++) {
39         sum += arr[i]; // Menambahkan semua elemen array
40     }
41     return sum;
42 }
43
44 // function to calculate average of n number
45 double avg(double sum, int n)
46 {
47     return sum / n;
48 }
49

```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Enter three Numbers : 3
6
9
Sum = 18
Average = 6

```

5.

```

1  #include<iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6      cout << "Enter the size of the matrix: ";
7      cin >> n; // Ukuran matriks
8
9      // Matriks diagonal
10     cout << "\nDiagonal Matrix:" << endl;
11     char diagonalMatrix[n][n];
12
13     // Mengisi dan menampilkan matriks diagonal
14     for (int i = 0; i < n; i++) {
15         for (int j = 0; j < n; j++) {
16             if (i == j) {
17                 diagonalMatrix[i][j] = 'X';
18             } else {
19                 diagonalMatrix[i][j] = ' ';
20             }
21             cout << diagonalMatrix[i][j] << " ";
22         }
23         cout << endl;
24     }
25
26     // Matriks segitiga bawah
27     cout << "\nLower Triangular Matrix:" << endl;
28     char lowerTriangularMatrix[n][n];
29
30     // Mengisi dan menampilkan matriks segitiga bawah
31     for (int i = 0; i < n; i++) {
32         for (int j = 0; j < n; j++) {
33             if (i >= j) {
34                 lowerTriangularMatrix[i][j] = 'X';
35             } else {
36                 lowerTriangularMatrix[i][j] = ' ';
37             }
38             cout << lowerTriangularMatrix[i][j] << " ";
39         }
40         cout << endl;
41     }
42
43     return 0;
44 }
45

```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Enter the size of the matrix: 3

Diagonal Matrix:
X
  X
    X

Lower Triangular Matrix:
X
X X
X X X

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Enter the size of the matrix: 4

Diagonal Matrix:
X
  X
    X
      X

Lower Triangular Matrix:
X
X X
X X X
X X X X

```

Check poin 3

Screenshot dan masukkan ke *worksheet* hasil kerja anda.

1.

Program yang diberikan berfungsi untuk mengakses elemen array berdasarkan indeks dan mencari elemen dalam array. Namun, masalah utama yang ada adalah tidak adanya pemeriksaan validitas indeks, sehingga jika pengguna memasukkan indeks yang lebih besar dari ukuran array, program akan menghasilkan error atau perilaku tak terduga. Untuk menyempurnakan program, kita dapat memecahnya menjadi tiga fungsi modular: `fillArray()` untuk mengisi array, `accessElement()` untuk mengakses elemen berdasarkan indeks dengan validasi, dan `searchElement()` untuk mencari elemen dalam array. Fungsi `accessElement()` melakukan pemeriksaan apakah indeks yang dimasukkan valid (dalam rentang 0 hingga $n-1$). Dengan cara ini, program menjadi lebih aman karena menghindari kesalahan akses array dan lebih modular dalam pemrogramannya, memberikan pesan yang sesuai jika pengguna memasukkan indeks yang tidak valid.

```

1  #include<iostream>
2  using namespace std;
3
4  // Fungsi untuk mengisi array
5  void fillArray(int arr[], int n) {
6      cout << "Enter elements of array" << endl;
7      for(int i = 0; i < n; i++) {
8          cin >> arr[i];
9      }
10 }
11
12 // Fungsi untuk mengakses elemen berdasarkan indeks
13 void accessElement(int arr[], int n, int l) {
14     if (l >= 0 && l < n) {
15         cout << "Element present at index " << l << " is " << arr[l] << endl;
16     } else {
17         cout << "Invalid index! Please enter a valid index between 0 and " << n-1 << endl;
18     }
19 }
20
21 // Fungsi untuk mencari elemen dalam array
22 void searchElement(int arr[], int n, int k) {
23     int ans = -1;
24     for(int i = 0; i < n; i++) {
25         if(arr[i] == k) {
26             ans = i;
27             break;
28         }
29     }
30     if(ans != -1)
31         cout << "The element " << k << " is present at index " << ans << endl;
32     else
33         cout << "The element " << k << " is not there in the array" << endl;
34 }
35
36 int main() {
37     int n, l, k;
38
39     cout << "Enter size of array" << endl;
40     cin >> n;
41
42     int arr[n];
43
44     // Memanggil fungsi untuk mengisi array
45     fillArray(arr, n);
46
47     // Mengakses elemen berdasarkan indeks
48     cout << "Accessing element at index. Enter the index: ";
49     cin >> l;
50     accessElement(arr, n, l); // Memanggil fungsi accessElement
51
52     // Mencari elemen dalam array
53     cout << "Enter element to be searched: ";
54     cin >> k;
55     searchElement(arr, n, k); // Memanggil fungsi searchElement
56
57     return 0;
58 }
59

```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Enter size of array
3
Enter elements of array
3
2
1
Accessing element at index. Enter the index: 0
Element present at index 0 is 3
Enter element to be searched: 2
The element 2 is present at index 1

```

2.


```

1  #include<iostream>
2  using namespace std;
3
4  // Fungsi untuk meng-inputkan array
5  void inputArray(int arr[], int size) {
6      cout << "Input list is:" << endl;
7      for (int i = 0; i < size; i++) {
8          cin >> arr[i];
9      }
10 }
11
12 // Fungsi untuk mengurutkan array dan menampilkan hasilnya
13 void sortAndDisplay(int arr[], int size) {
14     // Menampilkan array sebelum diurutkan
15     cout << "Input list is" << endl;
16     for (int i = 0; i < size; i++) {
17         cout << arr[i] << "\t";
18     }
19     cout << endl;
20
21     // Mengurutkan array menggunakan Insertion Sort
22     for (int k = 1; k < size; k++) {
23         int temp = arr[k];
24         int j = k - 1;
25         while (j >= 0 && temp <= arr[j]) {
26             arr[j + 1] = arr[j];
27             j = j - 1;
28         }
29         arr[j + 1] = temp;
30     }
31
32     // Menampilkan array setelah diurutkan
33     cout << "\nOutput list is:" << endl;
34     for (int i = 0; i < size; i++) {
35         cout << arr[i] << "\t";
36     }
37     cout << endl;
38 }
39
40 int main() {
41     int myarray[5];
42
43     // Menginputkan array
44     inputArray(myarray, 5);
45
46     // Mengurutkan array dan menampilkan hasilnya
47     sortAndDisplay(myarray, 5);
48
49     return 0;
50 }
51

```

Praktikan@CDSR3-33 MSYS /c/iam

\$./a.exe

Input list is:

2

5

7

1

4

Input list is

2 5 7 1 4

Output list is:

1 2 4 5 7

3.

```

1  #include<iostream>
2  using namespace std;
3
4  // Fungsi untuk menampilkan elemen matriks
5  void displayMatrix(int matrix[3][3], int rows, int cols) {
6      cout << "Matrix elements are:" << endl;
7      for (int i = 0; i < rows; i++) {
8          for (int j = 0; j < cols; j++) {
9              cout << matrix[i][j] << "\t";
10             }
11             cout << endl;
12         }
13     }
14
15     // Fungsi untuk mencari nilai terkecil dalam matriks
16     void findMinElement(int matrix[3][3], int rows, int cols) {
17         int minValue = matrix[0][0];
18         int minRow = 0, minCol = 0;
19
20         // Mencari nilai terkecil dan indeksinya
21         for (int i = 0; i < rows; i++) {
22             for (int j = 0; j < cols; j++) {
23                 if (matrix[i][j] < minValue) {
24                     minValue = matrix[i][j];
25                     minRow = i;
26                     minCol = j;
27                 }
28             }
29         }
30
31         // Menampilkan hasil
32         cout << "The smallest value in the matrix is " << minValue << endl;
33         cout << "It is located at index [" << minRow << "][" << minCol << "]" << endl;
34     }
35
36     int main() {
37         int matrix[3][3] = {
38             {12, 4, 7},
39             {9, 1, 5},
40             {6, 3, 8}
41         };
42
43         // Menampilkan elemen-elemen matriks
44         displayMatrix(matrix, 3, 3);
45
46         // Mencari nilai terkecil dan indeksinya
47         findMinElement(matrix, 3, 3);
48
49         return 0;
50     }
51

```

```

Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Matrix elements are:
12      4      7
9       1      5
6       3      8
The smallest value in the matrix is 1
It is located at index [1][1]

```

4.

Program di atas menggunakan struktur (struct) dan alokasi dinamis dengan operator new untuk membuat array dari objek Employee. Setiap elemen dari array DynArray adalah objek dari struktur Employee, yang berisi dua anggota data, yaitu Name (tipe data string) dan Age (tipe data int).

Tipe data yang digunakan dalam program ini adalah:

1. struct Employee: Struktur data yang berisi dua atribut yaitu Name (string) dan Age (integer).
2. Employee* DynArray: Pointer yang menunjuk ke array dinamis yang berisi objek Employee.

3. `new (nothrow) Employee[3]`: Alokasi memori untuk array yang berisi 3 objek `Employee`.

```
Praktikan@CDSR3-33 MSYS /c/iam
$ ./a.exe
Displaying the Array Content
Name: Agus      Age: 33
Name: Sigit     Age: 26
Name: Sulis     Age: 52
```

Untuk menambah data dalam array dinamis, kita perlu mengalokasikan ulang memori array tersebut dengan ukuran yang lebih besar. Proses ini dimulai dengan membuat array baru dengan ukuran yang lebih besar menggunakan operator `new`. Kemudian, kita menyalin elemen-elemen dari array lama ke array baru. Setelah itu, data baru dapat dimasukkan ke dalam array baru di posisi yang tepat, yaitu pada indeks terakhir yang kosong. Setelah selesai, kita harus menghapus array lama menggunakan `delete[]` untuk mencegah kebocoran memori, dan mengarahkan pointer array lama ke array yang baru. Proses ini memungkinkan kita untuk memperluas array dinamis meskipun ukurannya telah ditentukan sebelumnya saat alokasi pertama.

Kode :

```
1  #include <iostream>
2  using namespace std;
3
4  struct Employee {
5      string Name;
6      int Age;
7  };
8
9  int main() {
10     int size = 3;
11     Employee* DynArray;
12     DynArray = new (nothrow) Employee[size];
13
14     DynArray[0].Name = "Agus";
15     DynArray[0].Age = 33;
16     DynArray[1].Name = "Sigit";
17     DynArray[1].Age = 26;
18     DynArray[2].Name = "Sulis";
19     DynArray[2].Age = 52;
20
21     // Menambah satu elemen baru
22     size++; // Menambah ukuran array
23     Employee* tempArray = new (nothrow) Employee[size]; // Alokasi array baru dengan ukuran lebih besar
24
25     // Menyalin data dari DynArray ke tempArray
26     for (int i = 0; i < size - 1; i++) {
27         tempArray[i] = DynArray[i];
28     }
29
30     // Menambah data baru pada elemen terakhir
31     tempArray[size - 1].Name = "Budi";
32     tempArray[size - 1].Age = 40;
33
34     delete[] DynArray; // Menghapus array lama
35     DynArray = tempArray; // Mengarahkan pointer DynArray ke tempArray yang baru
36
37     // Menampilkan data setelah penambahan
38     cout << "Displaying the Array Content" << endl;
39     for (int i = 0; i < size; i++) {
40         cout << "Name: " << DynArray[i].Name << "\tAge: " << DynArray[i].Age << endl;
41     }
42
43     delete[] DynArray; // Menghapus array yang sudah tidak digunakan
44     return 0;
45 }
46
```