

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

MODUL IV



Disusun Oleh :

NAMA : Fadhel Yussie Ramadhan

NIM : 2311102322

Dosen

Wahyu Andi Saputra, S.Pd., M.Eng

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024

A. Dasar Teori

Materi yang di bahas adalah tentang array.

Array merupakan struktur data yang paling dasar. Sebelum mempelajari struktur data lainnya, alangkah baiknya, kita perlu mempelajari Array terlebih dahulu. Berikut ini adalah beberapa jenis array :

1. Array Satu Dimensi Array satu dimensi adalah tipe variabel yang terdiri dari kumpulan data dengan tipe yang sama yang disusun dalam satu baris atau satu dimensi. Setiap elemen di dalam array memiliki sebuah indeks atau nomor yang digunakan untuk mengakses elemen tersebut. Indeks dimulai dari 0 dan berakhir pada jumlah elemen dikurangi satu. Contohnya, sebuah array satu dimensi yang berisi bilangan bulat {1, 2, 3, 4, 5} memiliki lima elemen dan indeksnya dimulai dari 0. Indeks 0 merujuk pada elemen pertama, indeks 1 merujuk pada elemen kedua, dan seterusnya hingga indeks 4 merujuk pada elemen kelima.
2. Array Dua Dimensi Array dua dimensi adalah variable yang terdiri dari kumpulan array satu dimensi dengan tipe yang sama yang disusun dalam baris dan kolom. Dalam array dua dimensi, setiap elemen memiliki dua indeks, yaitu indeks baris dan indeks kolom. Indeks baris menunjukkan posisi elemen dalam baris, sementara indeks kolom menunjukkan posisi elemen dalam kolom.
3. Array MultiDimensi
4. Array Empat dimensi
5. Array lima dimensi

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

--

```

#include <iostream> using
namespace std;

    struct
Node
{
    int data;
    Node *next;
};

Node *head;
Node *tail;
    void
init()
{
    head =
NULL;    tail =
NULL;
} bool
isEmpty()
{
    return head ==
NULL;
} void insertDepan(int
nilai)
{
    Node *baru = new Node;
baru->data = nilai;
baru->next = NULL;    if
(isEmpty())
    {
        head = tail =
baru;
    }    else    {
baru->next = head;
head = baru;
    }
} void insertBelakang(int
nilai)

```

```

{
    Node *baru = new Node;
    baru->data = nilai;
    baru->next = NULL;    if
    (isEmpty())
    {
        head = tail =
    baru;
    }    else    {
    tail->next = baru;
    tail = baru;
    }
} int
hitungList()

{
    Node *hitung = head;
    int jumlah = 0;    while
    (hitung != NULL)

    {
        jumlah++;
    hitung = hitung->next;
    }    return
    jumlah;
} void insertTengah(int data, int
posisi)
{    if (posisi < 1 || posisi >
hitungList())
    {
        cout << "Posisi diluar jangkauan" <<
endl;

    }    else if (posisi
== 1)
    {
        cout << "Posisi bukan posisi tengah" <<
endl;
    }
else
    {
        Node *baru = new Node();
        baru->data = data;        Node
        *bantu = head;        int nomor =
        1;        while (nomor < posisi -
        1)

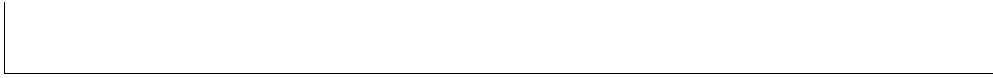
```



```

        {
            bantu =
bantu->next;
nomor++;
        }
        baru->next =
bantu->next;
        bantu->next
= baru;
    }
} void
hapusDepan()
{
    if
(!isEmpty())
    {
        Node *hapus = head;
if (head->next != NULL)
        {
            head =
head->next;
        }
        else
        {
            head = tail =
NULL;
        }
delete hapus;
    }
    else
    {
        cout <<
"List kosong!" << endl;
    }
} void
hapusBelakang()
{
    if
(!isEmpty())
    {
        Node *hapus = tail;
if (head != tail)
        {
            Node *bantu = head;
            while (bantu->next != tail)
            {
                bantu = bantu->next;
            }
            tail
= bantu;
            tail->next = NULL;

```



```

        } else
    {
        head = tail =
NULL;
    }
    delete hapus;
    } else { cout <<
"List kosong!" << endl;
    }
} void hapusTengah(int
posisi)
{ if (posisi < 1 || posisi >
hitungList())

    {
        cout << "Posisi di luar jangkauan" <<
endl;
    } else if (posisi
== 1)

    {
        cout << "Posisi bukan posisi tengah" <<
endl;
    }
else

    {
        Node *bantu = head;
        Node *hapus;
        Node *sebelum = NULL;
        int nomor = 1;
        while (nomor < posisi)
        {
            sebelum =
bantu;
            bantu =
bantu->next;
            nomor++;
        }
        hapus =
bantu;
        if (sebelum
!= NULL)
        {
            sebelum->next =
bantu->next;
        } else
        {
            head = bantu-
>next;
        }
    }
}

```




```

        delete hapus;
    }
} void ubahDepan(int
data)
{    if
(!isEmpty())
    {        head->data =
data;
    }    else    {        cout << "List
masih kosong!" << endl;
    }
} void ubahTengah(int data, int
posisi)
{    if
(!isEmpty())
    {        if (posisi < 1 || posisi >
hitungList())
        {            cout << "Posisi di luar
jangkauan" << endl;
        }        else if
(posisi == 1)
        {            cout << "Posisi bukan posisi
tengah" << endl;
        }
    else
    {
        Node *bantu = head;
int nomor = 1;

        while (nomor < posisi)
        {            bantu
= bantu->next;
nomor++;
        }
bantu->data = data;
    }    }    else    {        cout
<< "List masih kosong!" << endl;    }
}

```



```

} void ubahBelakang(int
data)
{
    if
(!isEmpty())
    {
        tail->data =
data;
    }
    else
    {
        cout << "List
masih kosong!" << endl;
    }
} void
clearList()
{
    Node *bantu = head;
Node *hapus;
    while (bantu != NULL)
    {
        hapus = bantu;
bantu = bantu->next;
delete hapus;
    }
    head = tail = NULL;
    cout <<
"List berhasil terhapus!" << endl;
} void
tampil()
{
    Node *bantu = head;
if (!isEmpty())

    {
        while (bantu !=
NULL)
        {
            cout << bantu->
data << " ";
            bantu =
bantu->next;
        }
        cout << endl;
    }
    else
    {
        cout << "List
masih kosong!" << endl;
    }
}

```

```
}  
int main()  
{    init();  
insertDepan(3);  
tampil();  
insertBelakang(5);  
tampil();  
insertDepan(2);  
tampil();  
insertDepan(1);  
tampil();  
hapusDepan();  
tampil();  
hapusBelakang();  
tampil();  
insertTengah(7, 2);  
tampil();  
hapusTengah(2);  
tampil();  
ubahDepan(1);  
tampil();  
ubahBelakang(8);  
tampil();  
ubahTengah(11, 2);  
tampil();  
    return  
0;  
}
```

Screenshots Output

```
3
35
235
1235
235
23
273
23
13
18
Posisi bukan posisi tengah
111
PS C:\Users\USER\AppData\Local\Temp>
```

Deskripsi:

Program meminta user menginputkan array sebanyak 233 elemen lalu datanya akan ditampilkan beserta input array nya.

E. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 2

```

#include <iostream>

using namespace std;
struct Node {
string data;
Node *next;
};

Node *head, *tail, *baru, *bantu, *hapus;
void init() {
head = NULL;
tail = head;
} int
isEmpty() {
return head == NULL;
} void buatNode(string
data) { baru = new Node;
baru->data = data; baru-
>next = NULL;
} int hitungList() {
bantu = head; int jumlah
= 0; while (bantu !=
NULL) { jumlah++;
bantu = bantu->next;
} return
jumlah;
} void insertDepan(string
data) { buatNode(data);
if (isEmpty()) { head =
baru; tail = head;
baru->next = head;
} else {

```

```

        while (tail->next != head) {
tail = tail->next;
        }        baru->
>next = head;        head
= baru;        tail->next
= head;
        }
} void insertBelakang(string
data) {        buatNode(data);        if
(isEmpty()) {        head = baru;
tail = head;        baru->next =
head;
        } else {        while (tail->
>next != head) {        tail =
tail->next;
        }        tail->
>next = baru;
baru->next = head;
        }
} void insertTengah(string data, int
posisi) {        if (isEmpty()) {        head
= baru;        tail = head;        baru->
>next = head;
        } else {        baru->data =
data;        int nomor = 1;
bantu = head;        while (nomor <
posisi - 1) {        bantu =
bantu->next;        nomor++;
        }        baru->next =
bantu->next;        bantu->next
= baru;
        }
} void hapusDepan()
{        if (!isEmpty())
{

```



```

        hapus = head;          tail =
head;          if (hapus->next == head) {
head = NULL;          tail = NULL;
delete hapus;          } else {
while (tail->next != hapus) {
tail = tail->next;
        }          head
= head->next;          tail-
>next = head;
hapus->next = NULL;
delete hapus;
        } } else {          cout << "List
masih kosong!" << endl;
    }
} void hapusBelakang() {    if
(!isEmpty()) {          hapus = head;
tail = head;          if (hapus->next ==
head) {          head = NULL;
tail = NULL;          delete hapus;
} else {          while (hapus->next
!= head) {          hapus = hapus-
>next;
        }          while (tail-
>next != hapus) {          tail =
tail->next;
        }          tail-
>next = head;
hapus->next = NULL;
delete hapus;
    }
} else {

    cout << "List masih kosong!" << endl;
}
}

```

```

    init();
insertDepan("Ayam");
tampil();
insertDepan("Bebek");
tampil();
insertBelakang("Cicak");
tampil();
insertBelakang("Domba");
tampil();
hapusBelakang();
tampil();    hapusDepan();
tampil();
insertTengah("Sapi", 2);
tampil();
hapusTengah(2);
tampil();    return 0;
}

```

Screenshot output:

The screenshot shows a Windows command prompt window with the following output:

```

PS C:\Users\USER> cd C:\Users\USER\Documents\C++\unguided
Ayam
Bebek Ayam
Bebek Ayam Cicak
Bebek Ayam Cicak Domba
Bebek Ayam Cicak
Ayam Cicak
Ayam Sapi Cicak
PS C:\Users\USER\Documents

```

Overlaid on the command prompt is a Notepad window titled 'Fadhel Yussie Ramadhan' containing the text '2311102322'. The status bar at the bottom of the Notepad window indicates 'Ln 2, Col 11 | 33 of 33 character | 100% | Window | UTF-8'.

Deskripsi:

Program meminta user menginputkan 10 angka array yang kemudian di data dan di klasifikasikan ke dalam angka ganjil dan genap.

F. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```

#include <iostream>
#include <iomanip> using
namespace std;
    struct
Node
{
    string
nama;    string
nim;    Node
*next;
};
string tmp_nama;
Node *head;
Node *tail;
    void
init()
{
    head =
NULL;    tail =
NULL;
} bool
isEmpty()

{
    return head ==
NULL;
} void insertDepan(string nama, string
nim) {
    Node *baru = new Node;
baru->nama = nama;
baru->nim = nim;    baru-
>next = NULL;    if
(isEmpty())
    {
        head = tail =
baru;
    }    else    {
baru->next = head;
head = baru;

```

```

    }
} void insertBelakang(string nama, string
nim)
{
    Node *baru = new Node;
    baru->nama = nama;
    baru->nim = nim;    baru-
>next = NULL;    if
(isEmpty())
    {
        head = tail =
baru;
    }    else    {
tail->next = baru;
tail = baru;
    }
} int
hitungList()
{
    Node *hitung = head;
    int jumlah = 0;    while
(hitung != NULL)
    {
        jumlah++;
        hitung = hitung->next;
    }    return
jumlah;
} void insertTengah(string nama, string nim, int
posisi)
{
    if (posisi < 1 || posisi >
hitungList())
    {
        cout << "Posisi diluar jangkauan" <<
endl;
    }    else if (posisi
== 1)
    {
        cout << "Posisi bukan posisi tengah" << endl;
    }
else    {

```

```

        Node *baru = new Node();
baru->nama = nama;        baru->
>nim = nim;        Node *bantu =
head;        int nomor = 1;
while (nomor < posisi - 1)
    {
        bantu =
baru->next;
nomor++;
    }        baru->next =
baru->next;        bantu->next
= baru;

    }
} void
hapusDepan()
{
    if
(!isEmpty())
    {
        Node *hapus = head;
tmp_nama = head->nama;        if
(head->next != NULL)
        {
            head =
head->next;

        }        else
{
            head = tail =
NULL;

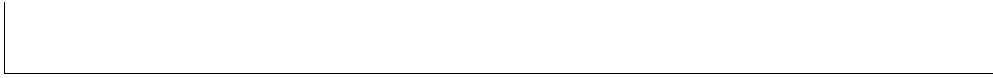
        }        delete hapus;        cout << "Data " <<
tmp_nama << " telah dihapus." << endl;        }        else {
cout << "List kosong!" << endl;
    }
} void
hapusBelakang()
{
    if (!isEmpty())
    {
        Node *hapus = tail;

```

```

        while (nomor <
posisi)
        {
            sebelum =
bantu;
            bantu =
bantu->next;
            nomor++;
        }
        hapus = bantu;
string tmp_nama = bantu->nama;
if (sebelum != NULL)
    {
        sebelum->next =
bantu->next;
    }
    else
    {
        head = bantu-
>next;
    }
    delete hapus;
    cout << "Data " <<
tmp_nama << " telah dihapus." << endl;
}
} void ubahDepan(string nama, string
nim)
{
    if
(!isEmpty())
    {
        tmp_nama = head->nama;
        head->nama =
nama;
        head->nim = nim;
        cout << "Data " <<
tmp_nama << " telah diganti dengan data " << nama << endl;
    }
    else
    {
        cout << "List
masih kosong!" << endl;
    }
}
} void ubahTengah(string nama, string nim, int
posisi)
{
    if (!isEmpty())
    {
        if (posisi < 1 || posisi >
hitungList())

```



```

        {
            cout << "Posisi di luar
jangkauan" << endl;
        }
        else if
        (posisi == 1)
        {
            cout << "Posisi bukan posisi
tengah" << endl;
        }
    else
    {
        Node *bantu = head;
        int nomor = 1;
        while (nomor < posisi)
        {
            bantu
            = bantu->next;
            nomor++;
        }
        tmp_nama = bantu->nama;
        bantu->nama = nama;
        bantu->nim = nim;
        cout << "Data " << tmp_nama << " telah diganti dengan data
" << nama << endl;
    }
    else {
        cout
        << "List masih kosong!" << endl;
    }
}
void ubahBelakang(string nama, string
nim)
{
    if
    (!isEmpty())
    {
        tmp_nama = tail->nama;
        tail->nama =
nama;
        tail->nim = nim;
        cout << "Data " <<
tmp_nama << " telah diganti dengan data " << nama << endl;
    }
    else
    {
        cout << "List masih kosong!" << endl;
    }
}
}

```




```

void
clearList()
{
    Node *bantu = head;
    Node *hapus;
    while (bantu != NULL)
    {
        hapus = bantu;
        bantu = bantu->next;
        delete hapus;
    }
    head = tail = NULL;
    cout <<
    "List berhasil terhapus!" << endl;
}
void
tampil()
{
    Node *bantu = head;
    if (!isEmpty())
    {
        cout << "\nDATA MAHASISWA\n\n";
        cout <<
        setw(37) << setfill('-') << "-" << setfill(' ')
        << endl;
        cout << "| " << setw(20) << left
        << "Nama"

        << " | " << setw(10) << "NIM"
        << " |" << endl;
        cout << setw(37) <<
        setfill('-') << "-" << setfill(' ')
        << endl;
        while (bantu
        != NULL)
        {
            cout << "| " << setw(20) << left <<
            bantu->nama <<
            " | " << setw(10) << bantu->nim << " |" << endl;
            bantu = bantu->next;
        }
        cout << setw(37) << setfill('-') << "-" <<
        setfill(' ')
        << endl;
        cout
        << endl;
    }
    else
    {
        cout << "List masih kosong!" << endl;
    }
}

```



```

int main()
{
    int operasi,
    posisi;    string nama;
    string nim;
    cout <<
endl
    << "===== " <<
endl;    cout << "  PROGRAM SINGLE LINKED LIST NON-CIRCULAR
" << endl;    cout <<
"===== " << endl;
cout << "1. Tambah Data di Depan" << endl;    cout << "2.
Tambah Data di Belakang" << endl;    cout << "3. Tambah Data
di Tengah" << endl;    cout << "4. Ubah Data di Depan" <<
endl;    cout << "5. Ubah Data di Belakang" << endl;
cout << "6. Ubah Data di Tengah" << endl;    cout << "7.
Hapus Data di Depan" << endl;    cout << "8. Hapus Data di
Belakang" << endl;    cout << "9. Hapus Data di Tengah" <<
endl;    cout << "10. Hapus Seluruh Data" << endl;    cout
<< "11. Tampilkan Data" << endl;    cout << "0. Keluar" <<
endl;    cout <<
"===== " << endl;
do    {        cout << "Pilih Operasi: ";        cin >>
operasi;        switch (operasi)
    {
case 1:
        cout << endl
        << "=== Tambah Depan ===" << endl
        << endl;
        cout << "Masukkan Nama : ";
        cin.ignore();        getline(cin,
nama);        cout << "Masukkan
NIM : ";        cin >> nim;
        insertDepan(nama, nim);
        cout << endl

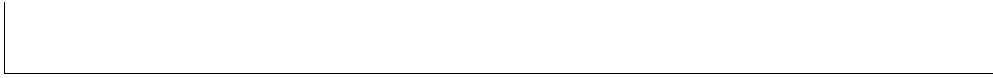
```



```

        << endl
        << "Data telah ditambahkan." << endl
        << endl;
break;    case 2:
        cout << endl
        << "=== Tambah Belakang ===" << endl
        << endl;
        cout << "Masukkan Nama : ";
        cin.ignore();          getline(cin,
        nama);                cout << "Masukkan
        NIM : ";              cin >> nim;
        insertBelakang(nama, nim);
        cout << endl          << endl
        << "Data telah ditambahkan." << endl
        << endl;
break;    case 3:
        cout << endl
        << "=== Tambah Tengah ===" << endl
        << endl;                cout
        << "Masukkan Nama : ";
        cin.ignore();          getline(cin,
        nama);                cout << "Masukkan NIM :
        ";                    cin >> nim;          cout
        << "Masukkan Posisi : ";          cin >>
        posisi;                insertTengah(nama, nim,
        posisi);                cout << endl
        << endl
        << "Data telah ditambahkan." << endl
        << endl;
break;    case 4:
        cout << endl
        << "=== Ubah Depan ===" << endl
        << endl;
        cout << "Masukkan Nama : ";
        cin.ignore();          getline(cin,
        nama);                cout << "Masukkan
        NIM : ";

```



```

        cin >> nim;
ubahDepan(nama, nim);
break;        case 5:

        cout << endl
                << "=== Ubah Belakang ===" << endl
                << endl;
cout << "Masukkan Nama : ";
cin.ignore();        getline(cin,
nama);        cout << "Masukkan
NIM : ";        cin >> nim;
ubahBelakang(nama, nim);
break;        case 6:
        cout << endl
                << "=== Ubah Tengah ===" << endl
                << endl;        cout
<< "Masukkan Nama : ";
cin.ignore();        getline(cin,
nama);        cout << "Masukkan NIM :
";        cin >> nim;
cout << "Masukkan Posisi : ";
cin >> posisi;
ubahTengah(nama, nim, posisi);
break;        case 7:
        cout << endl
                << "=== Hapus Depan ===" << endl
                << endl;
hapusDepan();
break;        case 8:
        cout << endl
                << "=== Hapus Belakang ===" << endl
                << endl;
hapusBelakang();
break;        case 9:

        cout << endl
                << "=== Hapus Tengah ===" << endl
                << endl;
cout << "Masukkan Posisi : ";

```



```
        cin >> posisi;
hapusTengah(posisi);
break;        case 10:

        cout << endl
        << "=== Hapus List ===" << endl
        << endl;
clearList();
break;        case 11:
tampil();
break;
        default:
break;
    }
} while (operasi != 0);
}
```

Screenshots Output

```

PS C:\Users\USER\GitHub\Struktur-Data-1> cd "c:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided\" ; if ($?)
{ g++ unguided2.cpp -o unguided2 } ; if ($?) { .\unguided2 }
Masukkan ukuran dari array 3 dimensi
masukkan jumlah baris: 2
masukkan jumlah kolom: 2
masukkan jumlah layer: 1
Input Array[0][0][0] = 1
Input Array[0][0][1] = 2
Input Array[0][1][0] = 3
Input Array[0][1][1] = 1

Data Array[0][0][0] = 1
Data Array[0][0][1] = 2
Data Array[0][1][0] = 3
Data Array[0][1][1] = 1

1 2
3 1

PS C:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided>

```

Deskripsi:

Program meminta user memasukkan jumlah baris kolom dan layer pada array tiga dimensi lalu user menginputkan masing masing array lalu program mendatanya dan memberi hasil nya.

G. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya) Unguided 3

```

I.    #include <iostream>
J.    using namespace std;
K.
L.    int min(int angka[], int ukuran) {
M.    int nilaimin = angka[0];
N.
O.        for (int i = 1; i < ukuran; i++) {
P.            if (angka[i] < nilaimin) {
Q.                nilaimin = angka[i];
R.            }
S.        }
T.
U.    return nilaimin;
V.    }
W.
X.    int maks(int angka[], int ukuran) {
Y.    int nilaimaks = angka[0];
Z.
AA.    for (int i = 1; i < ukuran; i++) {
BB.    if (angka[i] > nilaimaks) {
CC.    nilaimaks = angka[i];
DD.    } EE.    } FF.
GG.    return nilaimaks; HH. } II.
JJ.    int main() {
KK.    int angka[10]; LL.    int pilihan; MM.
NN.    cout << "Masukkan 10 angka: ";
OO.    for (int i = 0; i < 10; i++) {
PP.    cin >> angka[i]; QQ.    } RR.
SS.    cout << "\nMenu:" << endl;
TT.    cout << "1. Cari Nilai Minimum" << endl;
UU.    cout << "2. Cari Nilai Maksimum" << endl;
VV.    cout << "3. Hitung Rata rata" << endl;
WW.    cout << "Masukkan pilihan Anda: "; XX.
YY.    cin >> pilihan;
ZZ.
AAA.        switch (pilihan) {
BBB.            case 1: {
CCC.                int nilaiMin = min(angka, 10);

```

```

DDD. cout << "Nilai Minimum: " << nilaiMin << endl;
EEE. break;
FFF. }
GGG. case 2: {
HHH. int nilaiMaks = maks(angka, 10);
III. cout << "Nilai Maksimum: " << nilaiMaks << endl;
JJJ. break;
KKK. }
LLL. case 3: {
MMM. float jumlah = 0;
NNN. for (int i = 0; i < 10; i++) {
OOO. jumlah += angka[i];
PPP. }
QQQ. float rerata = jumlah / 10;
RRR. cout << "Rata-Rata: " << rerata << endl;
SSS. break;
TTT. }
UUU. default:
VVV. cout << "gagal!" << endl;
WWW.      }
XXX. YYY.      return 0; ZZZ. }
AAAA.

```

Screenshots Output

```

PS C:\Users\USER\GitHub\Struktur-Data-1> cd "c:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided\" ; if ($?)
{ g++ unguided3.cpp -o unguided3 } ; if ($?) { .\unguided3 }
Masukkan 10 angka: 3
4
3
4
3
4
3
4
3
4
3

Menu:
1. Cari Nilai Minimum
2. Cari Nilai Maksimum
3. Hitung Rata rata
Masukkan pilihan Anda: 1
Nilai Minimum: 3
PS C:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided>

```

```

PS C:\Users\USER\GitHub\Struktur-Data-1> cd "c:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided\" ; if ($?)
{ g++ unguided3.cpp -o unguided3 } ; if ($?) { .\unguided3 }
Masukkan 10 angka: 2
3
2
3
2
3
2
3
2
3
3

Menu:
1. Cari Nilai Minimum
2. Cari Nilai Maksimum
3. Hitung Rata rata
Masukkan pilihan Anda: 2
Nilai Maksimum: 3
PS C:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided>

```

```

PS C:\Users\USER\GitHub\Struktur-Data-1> cd "c:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided\" ; if ($?)
{ g++ unguided3.cpp -o unguided3 } ; if ($?) { .\unguided3 }
Masukkan 10 angka: 2
3
2
3
2
3
2
3
2
3
3

Menu:
1. Cari Nilai Minimum
2. Cari Nilai Maksimum
3. Hitung Rata rata
Masukkan pilihan Anda: 3
Rata-Rata: 2.5
PS C:\Users\USER\GitHub\Struktur-Data-1\Modul 2\unguided>

```

Deskripsi:

Membuat program mencari nilai minimum maksimum dan nilai rata rata dari 10 angka inputan user. Menggunakan control switch case.

Kesimpulan modul 1 ini mempelajari tentang 3

jenis array yaitu: -array satu dimensi yang di

susun dalam satu baris

-array dua dimensi yang terdiri dari baris dan kolom

-array multi dimensi yang terdiri dari 3 elemen atau lebih.

Jadi kesimpulannya, array adalah struktur data yang situasional (karena ada 3 array berbeda) tergantung kepada program angka yang akan di kerjakan.

Referensi

Modul IV tipe data praktikum algoritma struktur data IF 11 B Whatsapp group praktikum strukdat IF 11 B.