# 

# BAB II

# VARIABEL, TIPE DATA, DAN ARRAY

## 2.1. Tujuan

1. Praktikan dapat memahami konsep variabel, tipe data, dan array.
2. Praktikan dapat mengetahui penggunaan variabel, tipe data, dan array

dengan bahasa pemrograman Python, C++, Java, dan PHP.

1. Praktikan mampu mengimplementasikan variabel, tipe data dan array dalam

bahasa pemrograman Python, C++, Java, dan PHP.

1. Praktikan mampu memahami fungsi variabel,tipe data,array dalam bahasa pemrograman Python,C++,Java,dan PHP.
2. Praktikan mampu membedakan penempatan variabel,tipe data,array dalam bahasa pemrograman Pyhton,C++,Java,dan PHP.

## 2.2. Dasar Teori

### 2.2.1. Variabel

* Pengertian Variabel

Variabel adalah lokasi atau tempat di memori yang digunakan untuk menyimpan informasi atau data dari sebuah *program* untuk selanjutnya dapat diproses. Ketika kita membuat sebuah variabel, berarti kita memerintahkan sebuah memori untuk menyediakan alokasi *space* atau ruang bagi variabel. Ruang atau *space* yang dialokasikan tergantung dari tipe data dari variabel tersebut.

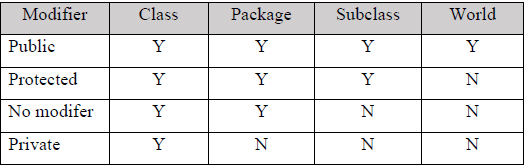
* Jenis - Jenis Variabel

Dalam bahasa Java, terdapat 3 jenis variabel yang dapat dikategorikan berdasarkan jangkauan ruang lingkupnya. Ruang lingkup yang dimaksud disini merujuk ke lokasi dimana variabel masing dapat digunakan atau tidak. Berikut ini merupakan 3 jenis variabel berdasarkan ruang lingkupnya, yakni sebagai berikut :

1. *Local Variable*Variabel yang dideklarasikan dalam sebuah lingkup kecil (lokal) seperti *method*/*constructor*/*blocks*. Berarti, setiap variabel yang dideklarasikan dalam *method* disebut lokal variabel. Variabel lokal mempunyai ruang lingkup yang kecil juga. Karena dideklarasikan di sebuah method, maka variabel tersebut juga hanya bisa diakses dalam sebuah *method* juga.
2. *Instance Variable*Variabel yang dideklarasikan dalam sebuah *class* namun diluar *method*. Java adalah bahasa pemrograman berbasis *object*, setiap kita menulis kode program java, secara tidak langsung kita sudah membuat *class*. Pendeklarasikan variabel langsung di dalam *class* tanpa dibungkus method sering disebut dengan *instance variable*. *Instance variable* juga dapat kita beri *access modifier*-nya masing-masing.
3. *Static Variable*Variabel yang dideklarasikan dengan *keyword static*, biasanya digunakan untuk mendeklarasikan “sesuatu” yang bersifat *live*, yang artinya variabel ini dapat diakses langsung tanpa membuat *object* dari *class* terlebih dahulu. Juga, *static variable* akan membuat sebuah “dunia” sendiri dan akan dipakai bersama-sama dengan *object* lain yang terbentuk. Penjelasan lebih lengkap akan dijelaskan ketika kita sudah memasuki materi *class* dan *object*.

* Macam - Macam Modifier

Secara umum ada 3 macam *modifier* yang digunakan dalam Java yaitu *public*, *private*, dan *protected*. Apabila kita tidak menggunakan tiga kata kunci tersebut, maka member atau *class* itu tidak menggunakan *modifier* (*no modifier*). Masing-masing *modifier* akan menentukan di mana saja member bisa diakses. Berikut ini merupakan tabel jangkauan untuk masing – masing *modifier* :



Gambar 2. 1 Tabel *modifier*

Sumber : <https://www.duniait.web.id/2020/08/pengertian-dan-jenis-variabel-java.html>

### 2.2.2. Tipe Data

### Pengertian Tipe Data

Tipe data ialah klasifikasi data yang mengenalkan kompilator atau penerjemah bagaimana programmer bermaksud untuk menggunakan data. Sebagian besar bahasa pemrograman mendukung berbagai macam data,sebagai contoh,bilangan riil,integer atau boolean. Sebuah tipe data menyediakan sekumpulan nilai-nilai dari ekspresi (yaitu variabel,fungsi) yang dapat mengambil nilai tersebut. Tipe data ini mendefinisikan operasi yang akan dilakukan pada data,maksud dari data disimpan.

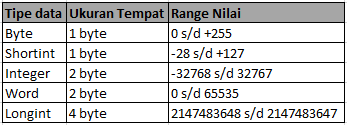
### Jenis-jenis Tipe Data

Jenis — jenis tipe data dibagi menjadi 2 yaitu Tipe data *primitive* (sederhana) dan tipe data *composite.*

1. Tipe data primitive (sederhana)

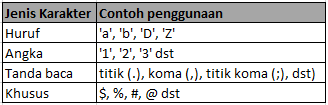
Tipe data primitive adalah tipe data yang hanya mampu menyimpan satu nilai pada setiap satu variabel. tipe data ini ini merupakan tipe data dasar yang sering digunakan program. Contoh tipe data primitif, yaitu : Tipe Numerik, tipe numerik ini digunakan pada variabel untuk menyimpan nilai dalam bentuk angka. Tipe data ini terbagi menjadi “Integer” dan “Real”.

* Tipe Numerik, tipe numerik ini digunakan pada variabel untuk menyimpan nilai dalam bentuk angka. Tipe data ini terbagi menjadi “Integer” dan “Real”.Integer *(int)*, merupakan tipe data bilangan bulat.



Gambar 2. 2 Tabel kategori integer

* Karakter (*char*), merupakan tipe data yang menyimpan hanya satu (1) digit karakter, karena ukuran satu digit itu satu *byte* (dan 1 *byte* itu = 8 bit). untuk penulisan karakter menggunakan petik tunggal ( ‘ ) di depan dan belakang karakter yang ditulis. contoh-nya seperti ini :



Gambar 2. 3 Tabel kategori karakter

* *Boolean*, merupakan tipe data logika yang hanya bernilai *true* (benar) dan *false* (salah). FYI, tipe data ini memakai memori paling kecil.



Gambar 2. 4 Contoh penggunaan *boolean*

2. Tipe data *composite*

Tipe data ini lawan dari tipe data *primitive*, tipe data *composite* dapat menampung banyak nilai. Antara lain :

* Tipe data ini lawan dari tipe data *primitive*, tipe data *composite* dapat menampilkan *Array*, atau disebut larik. Adalah tipe data yang terstruktur, meskipun sederhana *array* mampu menyimpan data dengan tipe yang sama (homogen) di dalam sebuah variabel.
* *Record* atau Struct, Seperti *array, record* atau *struct* termasuk data komposit. *Record* biasa dikenal pada bahasa pemrograman Pascal atau *Delphi* sedangkan *struct* lebih dikenal di C++. Lain hal dengan *array*, tipe data *record* menampung tipe data yang berbeda-beda (heterogen).
* *Image*, tipe data ini adalah tipe data grafik seperti contohnya grafik perkembangan suatu populasi. pada bahasa pemrograman modern yang berbasis visual, tipe data ini sudah sangat didukung.
* *Date Time*, nilai data *date* dan *time* disimpan dalam format yang spesifik. Dengan variabel atau konstanta yang dideklarasikan dapat disimpan dengan baik dalam bentuk tipe data “*date*” maupun “*time*”. Tipe data ini termasuk kelompok tipe data komposit, karena terbentuk dari beberapa tipe data.
* *Object*, tipe data ini menyimpan yang berhubungan dengan objek-obyek yang biasa ada pada bahasa pemrograman VB, *Delphi* dan bahasa pemrograman lain yang berbasis GUI (*Graphical User Interface*). Contoh-nya : jika sebuah form memiliki *control Command button*, kita beri nama *Command1*.
* *Subrange*, Tipe data bilangan yang mempunyai *range* nilai tertentu yang sudah ditentukan oleh seorang *programmer*. Biasanya, tipe data ini memiliki batas maksimum dan minimum.
* Enumerasi, Tipe data ini mempunyai elemen-elemen yang harus disebutkan satu per-satu dan memiliki nilai konstanta berupa integer yang sesuai dengan urutannya. Pada nilai konstanta-nya ini mewakili suatu nama variable yang ditulis didalam kurung. Tipe data ini sering ditemui dalam bahasa Delphi dan SQL.

Sumber : [https://codedevelopcoffee.wordpress.com/2019/04/15/pengertian-tipe-data-jenis-jenis-tipe-data-pada-pemrograman/#:~:text=Pengertian%20Tipe%20Data%20Dalam%20ilmu%20komputer%20dan%20pemrograman,data%2C%20sebagai%20contoh%2C%20bilangan%20riil%2C%20integer%20atau%20boolean.](https://codedevelopcoffee.wordpress.com/2019/04/15/pengertian-tipe-data-jenis-jenis-tipe-data-pada-pemrograman/" \l ":~:text=Pengertian%20Tipe%20Data%20Dalam%20ilmu%20komputer%20dan%20pemrograman,data%2C%20sebagai%20contoh%2C%20bilangan%20riil%2C%20integer%20atau%20boolean.)

### 2.2.3. Array

* Pengertian

Array (larik) ialah sekumpulan variabel-variabel yang memiliki nama dan tipe data yang sama satu dengan lainnya. Juga bisa diartikan sebagai sekelompok data sejenis yang tersimpan di dalam variabel dengan nama sama. maka dari itu Untuk dapat membedakan antara array satu dengan lainnya dapat dengan memberikan indeks pada masing-masing variable tersebut.

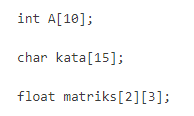
* Struktur data paling mendasar

Dalam menyusun fungsi array terdapat 3 struktur yang paling mendasar, antara lain

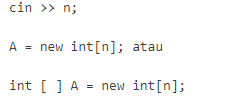
1. Kumpulan data dengan tipe sama yang jumlahnya tertentu
2. Akses setiap elemen menggunakan indeks
3. Disimpan pada tempat yang saling bersambungan (contiguous)

* Deklarasi

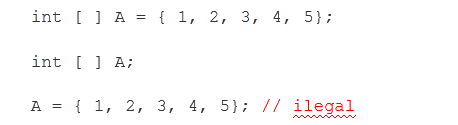
Untuk dapat memasukan array dalam sebuah program, kita harus memahami masing-masing pendeklarasian dari array tersebut, berikut ini deklarasi array :



Gambar 2.5 Array Statis



Gambar 2.5 Array Dinamis



Gambar 2.6 Inisialisasi

* Abstraksi

Abstraksi merupakan proses representasi data dan program yang sama dengan pengertiannya dengan menyembunyikan detail implementasi. hal ini bertujuan agar para programmer dapat lebih berfokus pada konsep tertentu saja dalam satu waktu. selain itu mengembangkan mekanisme abstrak untuk keperluan komputasi tertentu pada level yang lebih tinggi. ada 2 lapisan abstaksi yaitu:

1. Mendefinisikan objek yang ingin dimanipulasi (representasi data dalam struktur data).
2. Mendefinisikan operasi yang akan dilakukan pada struktur data (mengimplementasikan algoritma).

* TIPE DATA ABSTRAK (TDA)

Tipe Data abstrak atau sering juga disebut bstract data type (ADT) merupakan tipe data yang hanya dapat diakses lewat interface. Mengacu ke program dalam penggunaan TDA sebagai client untuk program yang menentukan tipe data. sebagai contoh dalam implementasi sebagai berikut:

⊕ Interface adalah opaque = client tidak dapat melihat implementasi

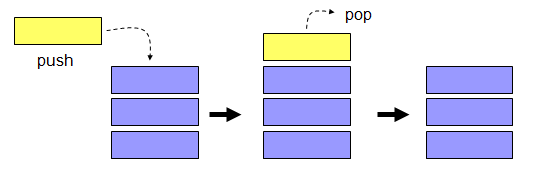
* KEUNTUNGAN TIPE DATA ABSTRAK (TDA)

penggunaan tipe data abstrak sangat berguna dalam melakukan pemrograman, karena dengan programmer dapat lebih berfokus pada satu program dan hasil yang memuaskan, berikut ada 4 keuntungan menggunakan TDA :

1. Menyediakan mekanisme yang efektif untuk mengorganisasikan sistem software yang besar
2. Menyediakan cara untuk membatasi ukuran dan kompleksitas interface antara algoritma dan
3. Struktur data yang bersesuaian dan program yang menggunakan algoritma dan struktur data
4. Interface TDA mendefinisikan arti yang “pas” dari komunikasi

* PUSHDOWN STACK TDA

Salah satu jenis TDA yang paling sering digunakan ialah pushdown stack. Pushdown Stack TDA mempunyai fungsi untuk menyediakan dua operasi dasar yaitu insert (push) item baru dan delete (pop) item yang paling akhir disisipkan agar dapat memahasi dengan mudah serta menggunakan sistem LIFO (Last in- first out) berikut contoh gambar dalam jenis TDA ini :



Gambar 2.7 Contoh Jenis TDA

* INTERFACES TDA PUSHDOWN-STACK

Interfaces TDA Pushdown-stack Digunakan untuk evaluasi ekspresi aritmetika :

⊕ Ekspresi Infiks

Operator berada di antara operan, Misalnya 4 + 5 (4 dan 5 adalah operan)

⊕ Ekspresi Posfiks

Operator berada setelah operan, misalnya 4 + 5 ditulis sebagai 4 5 +

⊕ Interface : Client bisa menggunakan empat operasi

Disimpan dalam STACK.h

* NOTASI POSFIKS

berikut ini contoh penggunaan notasi Posfiks :

⊗ Apa notasi posfiks untuk ekspresi infiks :

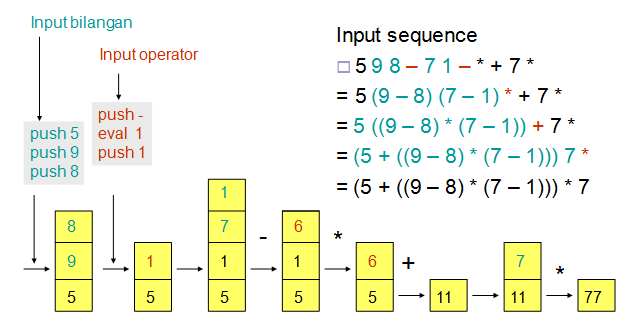
6 + 5 \* 9 ?

⊗ Apa notasi infiks untuk ekspresi posfiks :

5 9 8 + 4 6 \* \* 7 + \* ?

5 9 8 – 7 1 – \* + 7 \* ?

* NOTASI POSFIKS DAN PUSHDOWN STACK



Gambar 2.8 Contoh penggunaan Notasi posfiks dan Pushdown Stack

* TIPE DATA ABSTRAK (TDA) ARRAY

Tipe data abstrak array mempunya sifat setiap instan objek data adalah himpunan pasangan berbentuk (indeks, harga) dan tidak ada pasangan himpunan dengan indeks sama

* FUNGSI-FUNGSI YANG DIBERLAKUKAN PADA ARRAY :

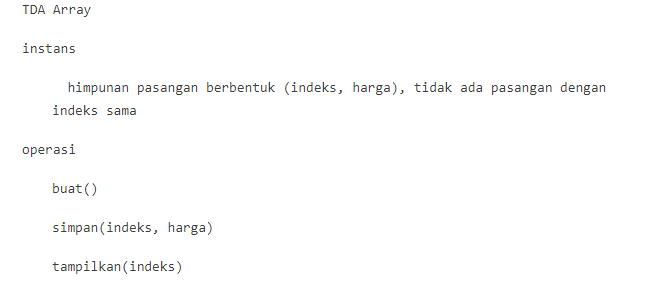
a. Buat = membuat array kosong awal.

b. Simpan = menambah pasangan berbentuk (indeks, harga) ke dalam himpunan, dan jika terdapat.

c. Pasangan dengan indeks sama = hapus pasangan lama.

d. Tampilkan = tampilkan harga dari pasangan bila diberikan indeksnya.

Sebagai contoh, berikut ini tertuang TDA array pada sebuah program :



Gambar 2.9 Contoh TDA array programaan

Sumber : <http://matkul.xyz/pengertian-dan-fungsi-array-dalam-pemrograman/>

## 2.3. Hasil Percobaan dan Analisis

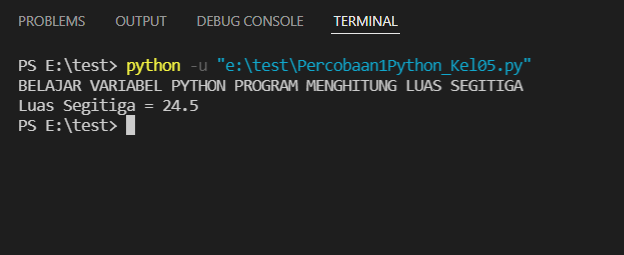
### 2.3.1 Deklarasi Variabel dan Tipe data pada Python

Berikut adalah *source code* yang digunakan pada percobaan Python ini:

*Source code* Python:

|  |
| --- |
| print('BELAJAR VARIABEL PYTHON PROGRAM MENGHITUNG LUAS SEGITIGA')  alas = 7  tinggi = 7  luas = 0.5 \* alas \* tinggi  print ('Luas Segitiga =', + luas) |

Dan berikut adalah *output* yang dihasilkan *source code* diatas:



Gambar 2. 10 Output Percobaan 1 Python

Pada *source code* dimulai dengan fungsi print(), dimana Python akan mencetak 'Belajar Variabel Python Program Menghitung Luas Segitiga’ sebagai hasil *output.* Kemudian diikuti dengan deklarasi variabel alas dan tinggi berupa Integer yang bernilai masing-masing 7. Yang dimana kedua nilai tersebut akan dioperasikan untuk menghasilkan nilai bagi variabel luas, yang kemudian akan di *print* menjadi hasil *output.*

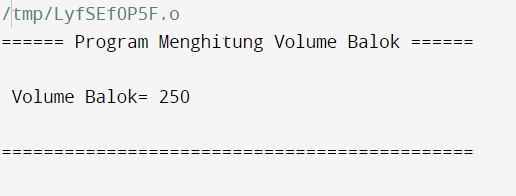
### 2.3.2. Deklarasi Variabel dan Tipe data pada C++

Berikut adalah *source code* yang digunakan pada percobaan C++ ini:

*Source code* C++:

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int volume, panjang, lebar, tinggi;  panjang = 5;  lebar = 5;  tinggi = 10;  volume = panjang \* lebar \* tinggi;  cout << "\n ====== Program Menghitung Volume Balok ======\n\n";  cout << " Volume Balok= " << volume <<endl;  cout << "\n============================================= \n\n";  return 0;} |

Dan berikutadalah hasil *output* dari *source code* di atas:



Gambar 2. 11 Output Percobaan 1 C++

*Source code* bekerja dengan mendeklarasikan 4 variabel bertipe integer (volume, panjang, lebar, tinggi), kemudian nilai dari 3 variabel akan dikalikan untuk mendapatkan nilai dari variabel volume. Dan hasil dari nilai volume akan ditampilkan menggunakan cout sebagai *output* dari program, *cout* adalah salah satu objek dari *file header* *iostream* yang berfungsi untuk menampilkan *output*.

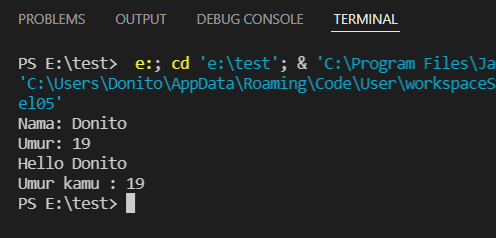
### 2.3.3. Deklarasi Variabel dan Tipe data pada Java

Berikut adalah *source code* yang digunakan pada percobaan Java ini:

*Source code* C++:

|  |
| --- |
| import java.util.Scanner;  public class Percobaan1Java\_Kel05 {  public static void main(String[] args) {  Scanner input = new Scanner( System.in );  System.out.print( "Nama: " );  String nama = input.next();  System.out.print( "Umur: " );  int umur = input.nextInt();  System.out.println("Hello "+nama);  System.out.println("Umur kamu : "+umur);  input.close();  }  } |

Dan berikutadalah hasil *output* dari *source code* diatas:



Gambar 2. 12 Output Percobaan 1 Java

*Source code* dimulai dengan meng-*import java.util.Scanner* yang merupakan *class* yang menyediakan fungsi-fungsi untuk mengambil *input* dari *keyboard*. Program bekerja dengan menampilkan kolom “Nama: “ dan “Umur: “ menggunakan System.out.print. Kemudian *user* diminta untuk memasukan nilai kepada variable nama dan umur (int) menggunakan keyboard dengan nilai berupa *input.*next()dan *input.nextInt*(). Program kemudian mencetak kembali nilai yang sudah di *input* menggunakan *System.out.println*.

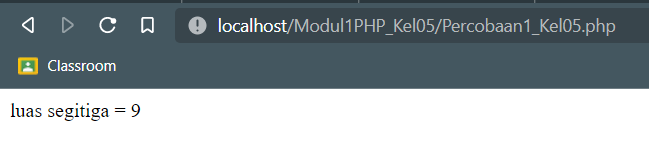
### 2.3.4. Deklarasi Variabel dan Tipe data pada PHP

Berikut adalah source code yang digunakan pada percobaan PHP ini:

*Source code* PHP:

|  |
| --- |
| <html>  <head>  <body>  <?php  $alas=6;  $tinggi=3;  $luas= 0.5 \* ($alas\*$tinggi) ;  echo "luas segitiga = $luas";  ?>  </body>  </head>  </html> |

Dan berikutadalah hasil *output* dari *source code* diatas:



Gambar 2. 13 Output Percobaan 1 PHP

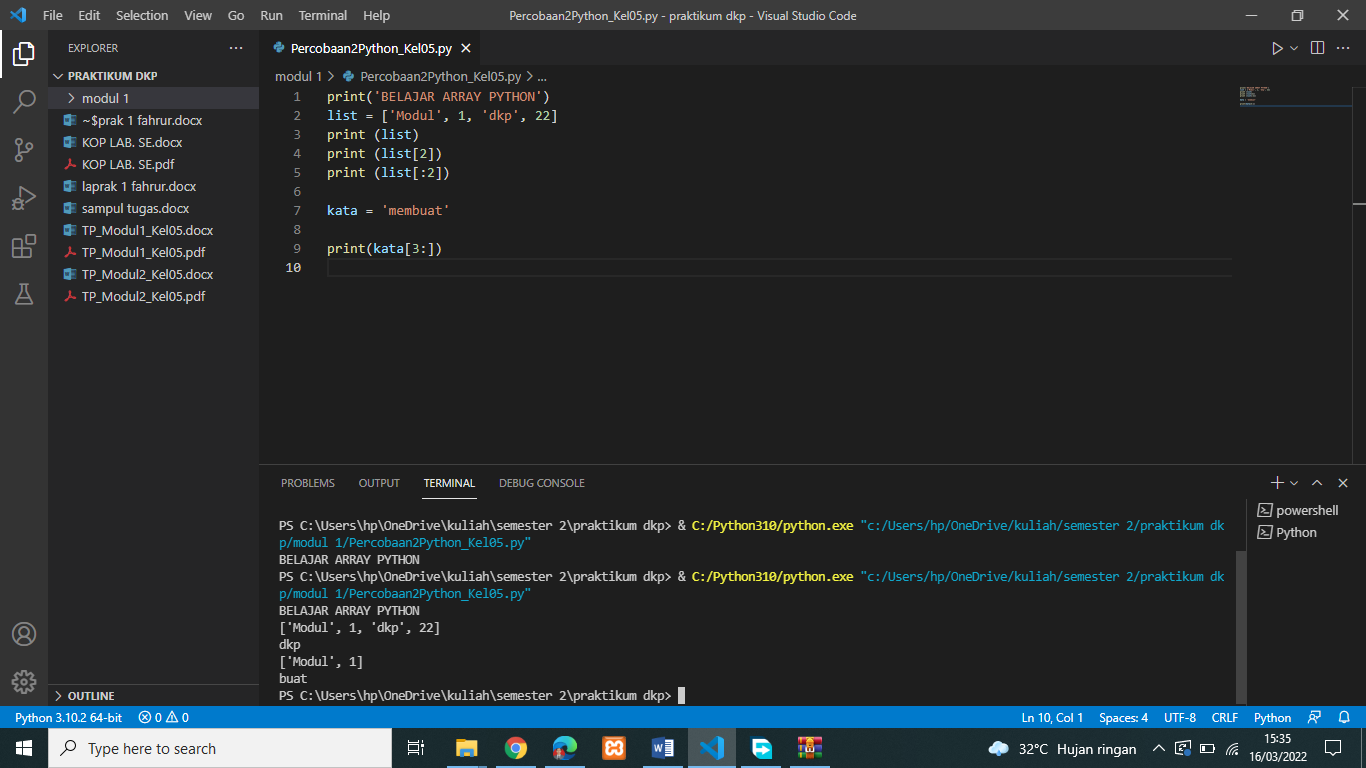
*Source code* dimulai dengan mendeklarasi 3 variabel ($alas, $tinggi, dan $luas), dimana variable $luas mendapatkan nilainya dari hasil pengoperasian menggunakan nilai $alas dan $tinggi. Kemudian digunakan fungsi echo untuk menampilkan nilai $tinggi ke layar.

### 2.3.5.     Array pada Python

*Source Code*:

|  |
| --- |
| print('BELAJAR ARRAY PYTHON')  list = ['Modul', 1, 'dkp', 22]  print (list)  print (list[2])  print (list[:2])  kata = 'membuat'  print(kata[3:]) |

*Output*:



Gambar 2. 14 Output array pada Python

Analisis:

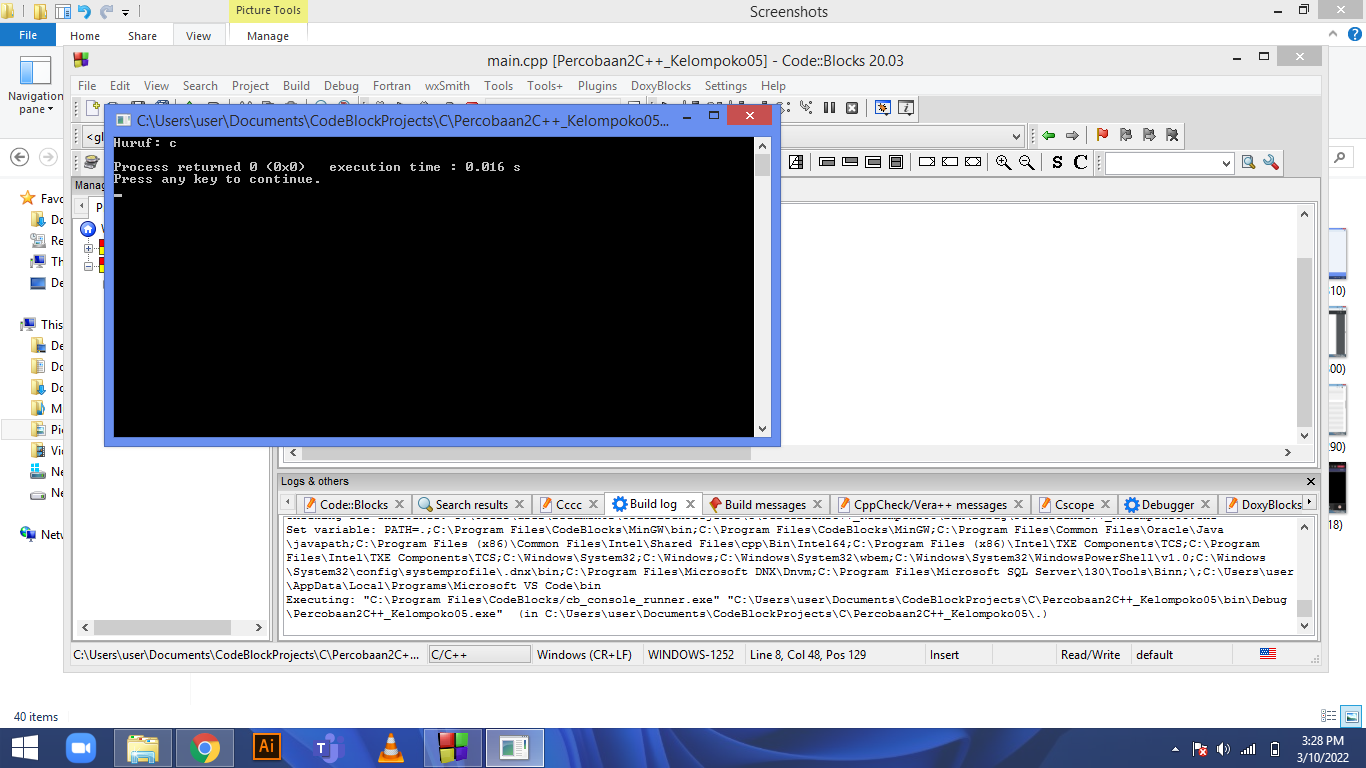
Pada *source code* Python di atas, *code* baris pertama berisi *print*(). *Code* ini tersebut berfungsi untuk menampilakan pada layar apa yang tertulis di dalam tutup kurung buka kurung. Dalam *code* pertama berfungsi untuk menampilkan kalimat BELAJAR *ARRAY* *PYTHON*. Code baris kedua adalah *variable* *list*, code ini berfungsi untuk menyimpan data dalam array. Didalam list terdapat dua kata tipe *String* yaitu 'Modul' dan 'dkp'. Terdapat juga data tipe *Integer* yaitu 1 dan 2.Dalam bahasa *Python* tipe data dapat menyesuaikan sendiri tipe data yang digunakan, sehingga kita tidak perlu mendeklarasikan tipe data tersebut. *Code Print (list)* digunakan untuk menampilkan seluruh isi dari *list*. *Code* print (list[2]) digunakan untuk menampilkan elemen *array list* nomer dua. *Code* print (list[:2]) digunakan untuk menampilkan isi elemen *array* *list* sebelum nomer dua. *Code* print (kata[3:]) digunakan untuk menampilkan isi *array kata* setelah nomer tiga.

### 2.3.6.     Array pada C++

*Source Code*:

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  int main() {      char huruf[5] = { 'a', 'b', 'c','d', 'e' };      //mengambil data pada array      cout << "Huruf: " << huruf[2] << endl;      return 0;  } |

*Output:*



Gambar 2. 15 Output array satu dimensi pada C++

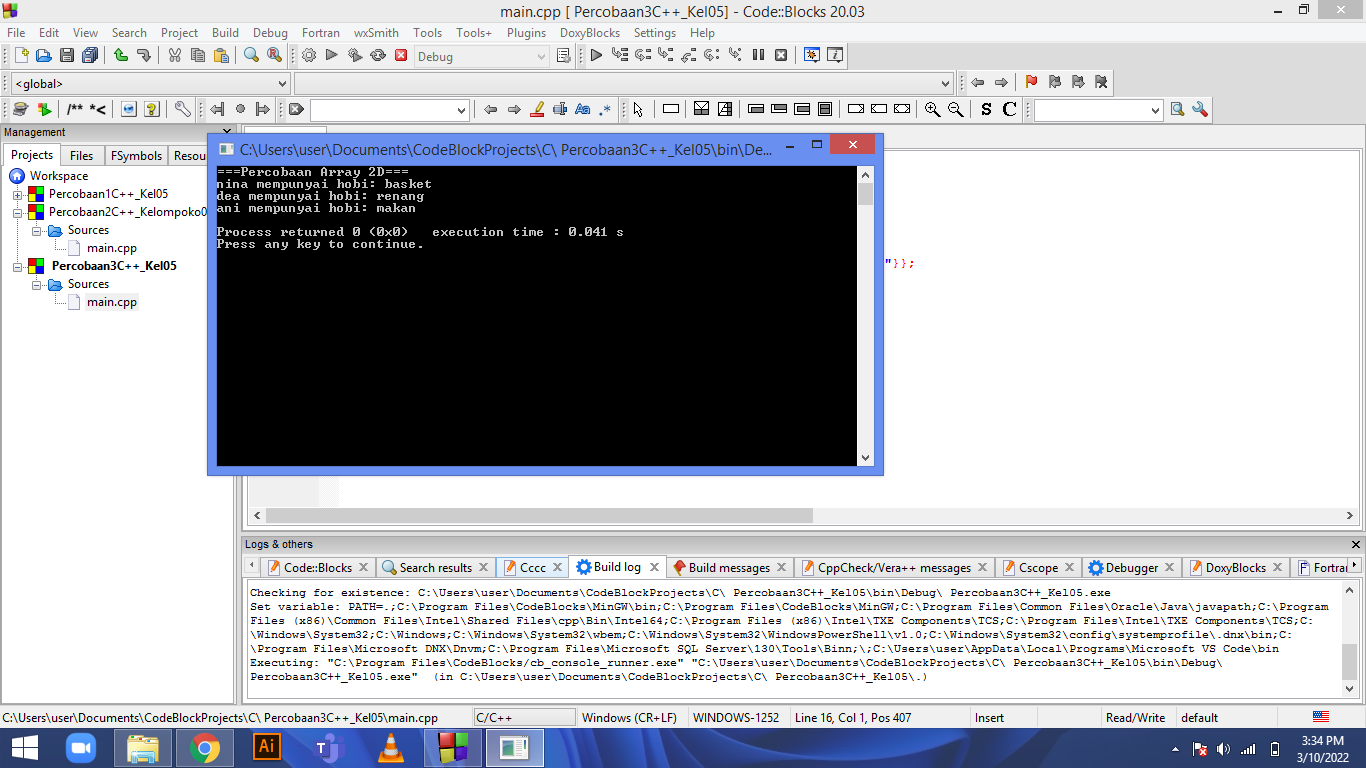
Analisis

Dalam *source code* ini #include <iostream> berfungsi untuk meng-*impor*t fungsi *input* dan fungsi *output* pada bahasa C++. #include <string> berfungsi untuk meng-*import* fungsi tipe data *String*. *using namespace std*; berfungsi menambahkan *namespace* saat kita menggunakan fungsi *cin* atau fungsi *cout. int main()* berfungsi memberikan nilai balik menurut tipe datanya. char huruf[5] = merupakan *array* satu dimensi karena hanya memiliki satu baris saja. *Cout* berfungsi untuk menampilkan *output*  dari *source code.* Pada *source code* ini yang ditampilkan adalah isi dari *array* nomer dua. Dikarenakan perhitungan *array* dimulai dari 0 sehingga nilai yang mucul adalah c.

*Source Code*:

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {    string arr[2][3] = {{"nina","dea","ani"},{"basket","renang","makan"}};    cout << "===Percobaan Array 2D===" << endl;    cout << arr[0][0] <<" mempunyai hobi: "<< arr[1][0] <<" "<<endl;    cout << arr[0][1] <<" mempunyai hobi: "<< arr[1][1] <<" "<<endl;    cout << arr[0][2] <<" mempunyai hobi: "<< arr[1][2] <<" "<<endl;    return 0;  } |

*Output:*



Gambar 2. 16 Output array dua dimensi pada C++

Analisis

Dalam *source code* ini #include <iostream> berfungsi untuk meng-*impor*t fungsi *input* dan fungsi *output* pada bahasa C++. using namespace std; berfungsi menambahkan *namespace* saat kita menggunakan fungsi cin atau fungsi cout. int main() berfungsi memberikan nilai balik menurut tipe datanya. string arr[2][3]= {{"nina", "dea", "ani"}, {"basket", "renang", "makan"}}; merupakan *array* dua dimensi karena memiliki dua baris *code*. Cout berfungsi untuk menampilkan *output*  dari *source code.* Pada *source code* cout << arr[0][0] <<" mempunyai hobi: "<< arr[1][0] <<" "<<endl; yang ditampilkan adalah nina memiliki hobi: basket.

Dikarenakan cout << arr[0][0] menampilkan *array* baris pertama data kenol yaitu nina. arr[1][0] menampilkan *array* baris kedua data kenol yaitu basket. Endl mengakhiri cout dan berganti baris pada *output.*

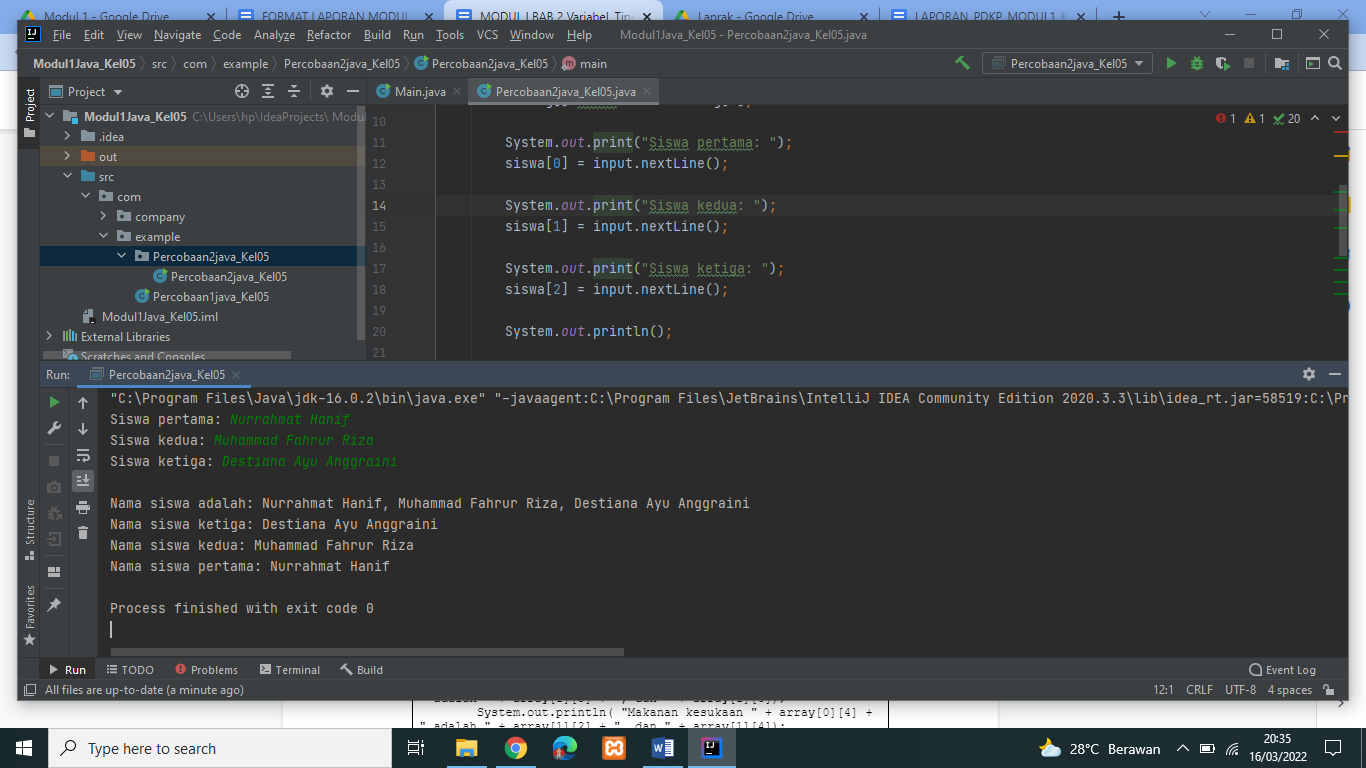
### 2.3.7.     Array pada Java

*Array* satu dimensipada bahasa Java.

*Source Code*

|  |
| --- |
| package com.example;  import java.util.Scanner; public class Percobaan2java\_Kel05 {    public static void main(String args[]){        Scanner input = new Scanner(System.in);        String[] siswa = new String[3];        System.out.print("Siswa pertama: ");       siswa[0] = input.nextLine();        System.out.print("Siswa kedua: ");       siswa[1] = input.nextLine();        System.out.print("Siswa ketiga: ");       siswa[2] = input.nextLine();        System.out.println();        System.out.print("Nama siswa adalah: ");       System.out.println(siswa[0]+", "+siswa[1]+", "+siswa[2]);       System.out.print( "Nama siswa ketiga: " );       System.out.println( siswa[2] );       System.out.print( "Nama siswa kedua: " );       System.out.println( siswa[1] );       System.out.print( "Nama siswa pertama: " );       System.out.println( siswa[0] );     } } |

*Output:*



Gambar 2. 17 Output array satu dimensi pada Java

Analisis

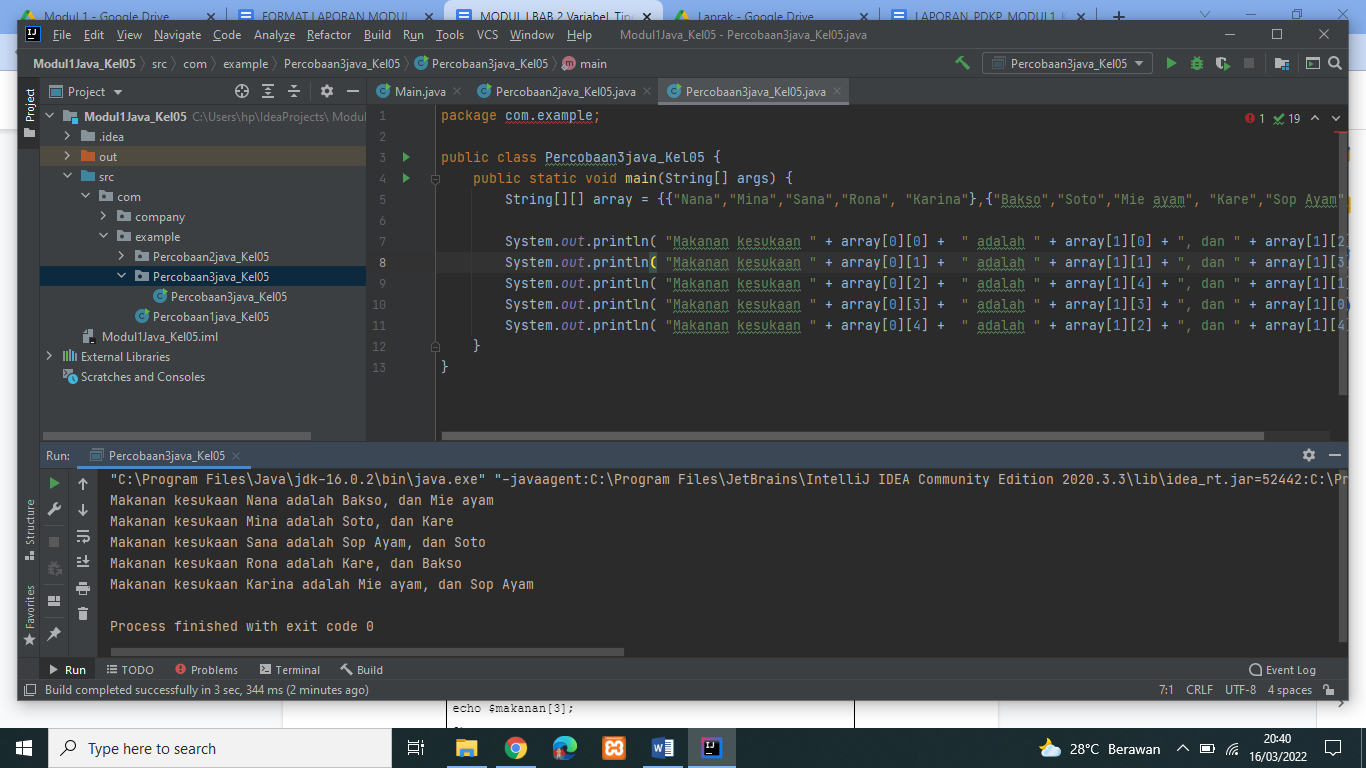
Pada *source code* ini import java.util.Scanner; merupakan merupakan *code* untuk memasukkan *packet scanner*.  *public static void main* adalah *main method* yang merupakan metode utama untuk menjalankan program. *scanner input = new Scanner(System.in)* merupakan perintah instansiasi *scanner class* ke dalam *variable* *input* jadi variabel yang kita masukkan akan berisi *scanner class*. string[] siswa = new String[3] adalah *array* kosong dan [3] merupakan batas atau ukuran *array*. *input.nextLine*() digunakan sebagai program agar pengguna bisa memasukan data dari nama murid. *system.out.print* adalah program untuk menampilkan *source code* seperti “Siswa pertama” dan *source code* yang lain.

*Array* dua *dimensi* pada bahasa Java.

*Source Code*:

|  |
| --- |
| package com.example;   public class Percobaan3java\_Kel05 {     public static void main(String[] args) {         String[][] array = {{"Nana","Mina","Sana","Rona", "Karina"},{"Bakso","Soto","Mie ayam", "Kare","Sop Ayam",}};          System.out.println( "Makanan kesukaan " + array[0][0] +  " adalah " + array[1][0] + ", dan " + array[1][2]);         System.out.println( "Makanan kesukaan " + array[0][1] +  " adalah " + array[1][1] + ", dan " + array[1][3]);         System.out.println( "Makanan kesukaan " + array[0][2] +  " adalah " + array[1][4] + ", dan " + array[1][1]);         System.out.println( "Makanan kesukaan " + array[0][3] +  " adalah " + array[1][3] + ", dan " + array[1][0]);         System.out.println( "Makanan kesukaan " + array[0][4] +  " adalah " + array[1][2] + ", dan " + array[1][4]); } } |

*Output:*



Gambar 2. 18 Output array dua dimensi pada Java

Analisis

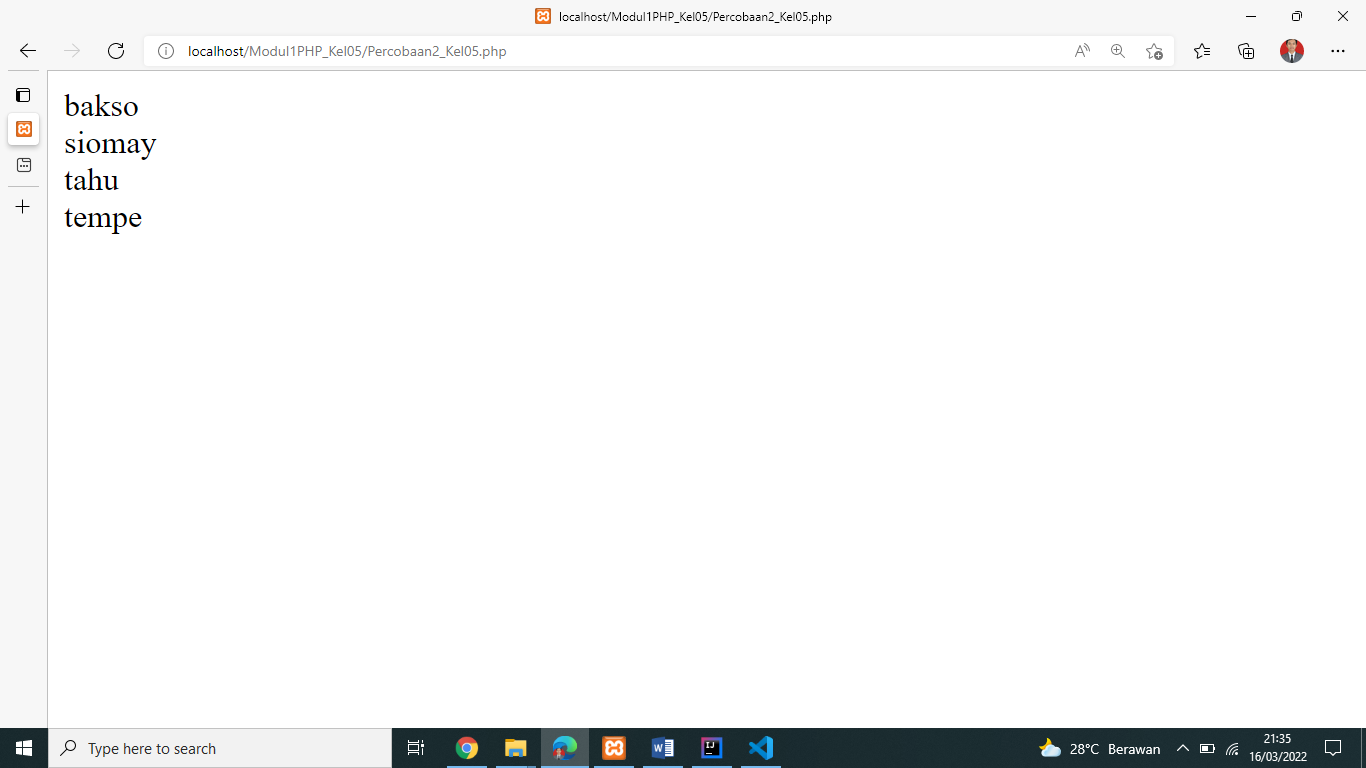
Pada program ini kita tidak menggunakan *scanner* seperti pada program sebelumnya. Sama seperti program sebelumnya *public static void main* adalah *main method* yang merupakan metode utama untuk menjalankan program. *Array* dengan tipe data *String* berfungsi untuk menyimpan data *String* . Data yang dimasukkan ke dalam *array*  dalam code tersebut adalah {{"Nana", "Mina", "Sana", "Rona", "Karina"}, {"Bakso", "Soto", "Mie ayam", "Kari", "Sop Ayam",}}. *Source code* ini termasuk 2 dimensi karena dalam satu *array* terdapat 2 baris data. *System.out.print* adalah program untuk menampilkan *output* yang berasal dari data *array*.

### 2.3.8.     Array pada PHP

*Source Code*:

|  |
| --- |
| <?php  $makanan[0] = "bakso";  $makanan[1] = "siomay";  $makanan[2] = "tahu";  $makanan[3] = "tempe";  echo $makanan[0];  echo "<br>";  echo $makanan[1];  echo "<br>";  echo $makanan[2];  echo "<br>";  echo $makanan[3];  ?> |

*Output:*



Gambar 2.19 Output array pada PHP

Analisis:

Pada *source code* ini merupakan progam bahasa PHP. Karakter dolar($) didepan kata menyatakan bahwa kata tersebut adalah variabel. Variabel dalam *code* diatas adalah $makanan. [] ini dinamakan slot *array* yang berisi angka didalam [] dan berfungsi sebagai urutan *array* sedangkan kata kata seperti "bakso"dan lain lainnya merupakan isi atau variabel yang terdapat pada *array*. Echo merupakan perintah menampilkan output pada layar.

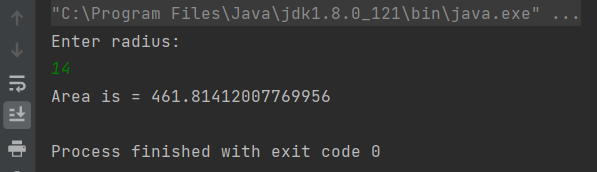
## 2.4. Tugas

### 2.4.1. Tugas 1

* *Source code* menggunakan Java:

|  |
| --- |
| /\*\*  ANGGOTA (NAMA // NIM // KELOMPOK // SHIFT)  1. DESTIANA AYU ANGGRAINI // 21120121120009 // KELOMPOK 5 // SHIFT 1  2. DONITO AKBAR LABAYA // 21120119120033 // KELOMPOK 5 // SHIFT 1  3. MUHAMMAD FAHRUR RIZA // 21120121140034 // KELOMPOK 5 // SHIFT 1  4. NURRAHMAT HANIF // 21120121130049 // KELOMPOK 5 // SHIFT 1  \*\*/  import java.util.Scanner;  public class Modul1 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("Enter radius:");  double radius = sc.nextDouble();  double area = Math.PI \* radius \* radius \* 3 / 4;  System.out.println("Area is = " + area);  }  } |

* *Output:*

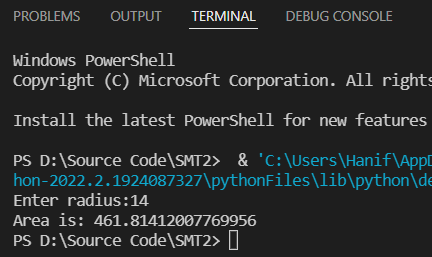


Gambar 2. 20 *Output* program menghitung luas lingkaran pada Java

* Analisis *source code*:
* Baris pertama membuat deklarasi untuk *import scanner*.
* Baris ketiga memulai program supaya program dapat dieksekusi.
* Baris keempat membuat *scanner* supaya dapat memasukkan data ke dalam program.
* Baris kelima membuat *output* berupa tulisan perintah untuk input nilai *radius* atau jari-jari.
* Baris keenam membuat *input scanner* untuk memasukkan nilai jari-jari menggunakan tipe data *double*.
* Baris ketujuh membuat variabel dan rumus untuk menghitung luas ¾ lingkaran menggunakan tipe data *double*.
* Baris kedelapan membuat *output string* untuk luas lingkaran dan ditambah variabel area untuk membuat *output* dari hasil rumus sebelumnya.
* *Source code* menggunakan Python

|  |
| --- |
| #ANGGOTA (NAMA // NIM // KELOMPOK // SHIFT)  #1. DESTIANA AYU ANGGRAINI // 21120121120009 // KELOMPOK 5 // SHIFT 1  #2. DONITO AKBAR LABAYA // 21120119120033 // KELOMPOK 5 // SHIFT 1  #3. MUHAMMAD FAHRUR RIZA // 21120121140034 // KELOMPOK 5 // SHIFT 1  #4. NURRAHMAT HANIF // 21120121130049 // KELOMPOK 5 // SHIFT 1  import math  radius = float(input('Enter radius:'))  area = math.pi \* radius \* radius \* 3 / 4  print('Area is:', + area) |

* *Output:*



Gambar 2. 21 *Output* program menghitung luas lingkaran pada Python

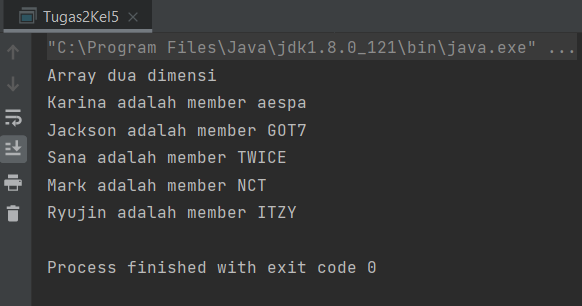
* Analisis *source code*:
* Baris pertama membuat *import math* untuk memanggil phi dalam kalkulasi program.
* Baris kedua membuat *input* data berupa *float* dengan variabel *radius* dan output *string* perintah untuk memasukkan data.
* Baris ketiga membuat variabel dan rumus untuk menghitung luas ¾ lingkaran.
* Baris keempat membuat *output string* untuk luas lingkaran dan ditambah variabel area untuk membuat *output* dari hasil rumus sebelumnya.

### 2.4.2. Tugas 2

* *Source code* menggunakan Java:

|  |
| --- |
| /\*\*  ANGGOTA (NAMA // NIM // KELOMPOK // SHIFT)  1. DESTIANA AYU ANGGRAINI // 21120121120009 // KELOMPOK 5 // SHIFT 1  2. DONITO AKBAR LABAYA // 21120119120033 // KELOMPOK 5 // SHIFT 1  3. MUHAMMAD FAHRUR RIZA // 21120121140034 // KELOMPOK 5 // SHIFT 1  4. NURRAHMAT HANIF // 21120121130049 // KELOMPOK 5 // SHIFT 1  \*\*/  import java.util.Arrays;  public class Tugas2Kel5 {  public static void main(String[] args) {  System.out.println("Array dua dimensi");  String[][]array2D= { {"Karina","Jackson","Sana","Mark", "Ryujin"},{"aespa","GOT7","TWICE", "NCT","ITZY",} };  System.out.println( array2D[0][0] + " adalah member " + array2D[1][0]);  System.out.println( array2D[0][1] + " adalah member " + array2D[1][1]);  System.out.println( array2D[0][2] + " adalah member " + array2D[1][2]);  System.out.println( array2D[0][3] + " adalah member " + array2D[1][3]);  System.out.println( array2D[0][4] + " adalah member " + array2D[1][4]);  }  } |

* *Output:*



Gambar 2. 22 *Output* program membuat program *array* dua dimensi pada Java

* Analisis *source code*:
* Baris pertama membuat *import Arrays* untuk memanggil fungsi *array*.
* Baris ketiga memulai program supaya program dapat dieksekusi.
* Baris keempat membuat output *String* untuk keterangan *array* dua dimensi.
* Baris kelima membuat variabel *String* untuk array dua dimensi.
* Baris keenam sampai kesepuluh membuat output *array* dua dimensi

## 2.5. Kesimpulan

1. Variabel dalam setiap bahasa pemrograman berbeda-beda.
2. Tipe datadalam setiap bahasa pemrograman berbeda-beda.
3. Logika dalam setiap bahasa pemrograman memiliki sistem yang mirip, namun dengan *syntax* yang berbeda.
4. Array dapat menampung data dalam jumlah besar dengan tipe data yang sama sehingga efektif penggunaannya.
5. Kesulitan dalam membuat program bervariasi dan relatif pada setiap bahasa pemrograman.
6. Setiap baris dalam program memiliki satu fungsi yang berkaitan pada baris selanjutnya.
7. Baris dalam program harus dibuat berurutan secara logika supaya program dapat berjalan lancar.