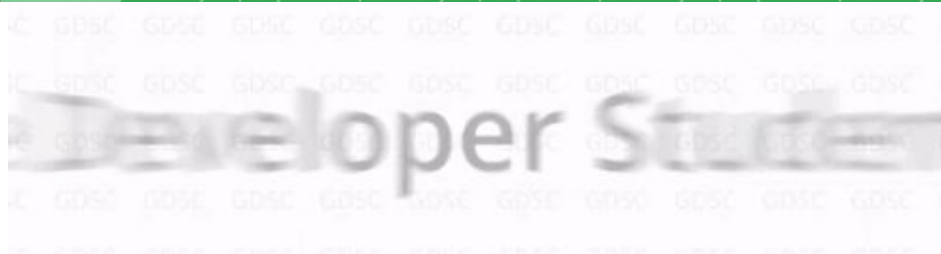


```
child: Column(  
  crossAxisAlignment: CrossAxisAlignment.  
  children:  
    /*2*/  
    Contact  
    padding:  
    children:  
      S  
    )  
  ),  
  Text(  
    'Ka  
    style:  
  ),  
  ),  
),
```



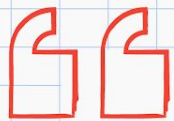


Study Jam

Week 1

Pemrograman dengan Python

Machine Learning



ML StudyJam Series

Agenda's today:

1. Hello & Introduction
2. Pretest
3. Present Material
4. QnA
5. Post Test & Feedback Form
6. Photo Session
7. Closing



```
function filterStudies([ studies, filterByOrg = false, filterByTopic = false ]){  
  return studies.filter(study => {  
    if (filterByOrg) {  
      return study.organizational_abbrev === filterByOrg;  
    }  
    if (filterByTopic) {  
      return study.topic === filterByTopic;  
    }  
    return true;  
  });  
}
```

Weekly Class Expectation Setting

Make yourself comfortable! 👍

1. Open Up Your Camera

We would love to see your beautiful and handsome faces!

2. Be Proactive

Please show your enthusiasm bcs we love to see it!

3. Take Notes

Please take notes of the important things :)

4. Don't Forget to Ask

Ask the speakers anything you would like to know

5. Don't Forget to complete pre-test, post-test and feedback form.



Pre Test

[link](#)

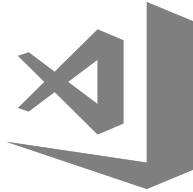
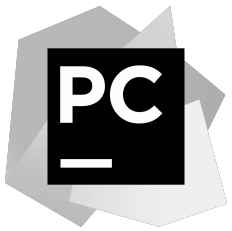
```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```



Berkenalan dengan Python

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Code Editor Untuk Python



```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Variable dan Assignment

Variabel adalah **sebuah tempat menyimpan sebuah nilai**

Assignment adalah **pemberian nilai pada suatu variabel**

```
<nama variable> = <nilai variable>
```

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```


Input, Output, dan Komentar

Input adalah **data yang dimasukkan ke dalam program**

```
input()
```

Output adalah **hasil yang diberikan oleh program setelah memproses input**

```
print()
```

Komentar adalah **keterangan pada kode program yang dikembangkan**

```
# inline commentary
```

```
"""  
block commentary  
"""
```

Berinteraksi dengan Data

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Abstraksi Data

Abstraksi Data adalah **kemampuan untuk mengerti konteks dan mempresentasikannya menjadi bentuk lain sesuai dengan konteks masalahnya**

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Data Typing: deklarasi dan inisialisasi

Deklarasi adalah proses memperkenalkan / pembuatan sebuah variabel. **Python tidak memiliki perintah untuk deklarasi.**

Inisialisasi adalah **pemberian nilai awal pada variabel yang telah dideklarasikan**

```
python = "Python"
```

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([0  
lookup.StaticV  
_buckets=5)
```

Tipe Data : Primitif

Tipe data **primitif** merupakan **jenis data yang paling dasar dalam pemrograman**.

- **Number** : integer, float, dan complex `integer = 2` `float = 2.5` `complex = 1+2j`
- **Boolean** `boolean = True`
- **String** : single quote (") atau double quote (")

```
string = 'ini'
```

```
string = "ini"
```

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Tipe Data : Collection

Tipe data **collection** merupakan **jenis data yang menyimpan satu atau lebih data primitif sebagai satu kelompok.**

- **List : ordered dan mutable** `python = [1, 0.5 , "python"]`
- **Tuple : ordered dan immutable** `python = (1, 0.5 , "python")`
- **Set : unordered, unique, dan mutable** `python = {1, 0.5 , "python"}`
- **Dictionary : (key : value) dan unordered** `python = {"nama" : "python"}`

Operasi pada List, Set, dan String

- **len** : menghitung banyak elemen
- **min & maks**: mengetahui nilai minimum dan maksimum
- **In dan Not In** : memastikan suatu nilai pada list
- **unpacking** : memberikan nilai untuk beberapa variabel
- **sort** : mengurutkan isi list

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Ekspresi

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```


Apa itu ekspresi?

Ekspresi adalah **kombinasi dari satu atau lebih variabel, konstanta, operator, dan fungsi yang bermakna untuk menghasilkan suatu nilai dalam tipe tertentu.**

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Jenis-Jenis Ekspresi

- aritmetika
- relasional
- logika

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Jenis-Jenis Operator

- **Aritmetika (+, -, *, /)**
- **Relasional (==, !=, >, <, >=, <=)**
- **Logika (and &, or |, not)**
- **Assignment (+=, -=, *=, /=, %=)**

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Control Flow

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Percabangan

- **If** : untuk satu kondisi
- **Else** : untuk 2 kondisi
- **Elif** : untuk 3 kondisi atau lebih

```
if kondisi:  
    statement
```

```
if kondisi 1:  
    statement 1  
else:  
    statement 2
```

```
if kondisi 1:  
    statement 1  
elif kondisi 2:  
    statement 2  
else:  
    statement 3
```

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Perulangan

- **For** : perulangan dengan jumlah iterasi **telah ditentukan**
- **While** : perulangan dengan jumlah iterasi **belum ditentukan**
- **Nested For** : perulangan di dalam perulangan

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Bentuk Perulangan Pada Python

```
for nilai in sequence:  
    blok kode yang diulang
```

```
while kondisi:  
    blok kode yang diulang
```

```
for nilai 1 in sequence 1:  
    for nilai 2 in sequence 2:  
        blok kode yang ingin diulang
```

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Let's take a break

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```



Array dan Pemrosesannya

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Fundamental Array

Array adalah **salah satu jenis dari struktur data linear**
dan terdiri dari kumpulan elemen bertipe data sama
dengan indeks yang berurutan atau linear

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Implementasi Array dengan Python

```
nama_variabel = [nilai_1, nilai_2, nilai_3, ..., nilai_n]
```

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Pemrosesan Sekuensial pada Array

Pemrosesan sekuensial adalah **sebuah pemrosesan setiap elemen array yang dimulai dari elemen pada indeks terkecil hingga terbesar**

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Matriks

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Fundamental Matriks

Matriks adalah **kumpulan data yang diatur dalam bentuk tabel dua dimensi dengan setiap elemennya terdefinisi berdasarkan baris dan kolom**

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Implementasi Matriks pada Python

```
nama_variabel = [  
    [nilai_1, nilai_2, nilai_3, ...],  
    [nilai_4, nilai_5, nilai_6, ...],  
    [nilai_7, nilai_8, nilai_9, ...],  
]
```

Operasi Matriks pada Python

Operasi 1 Matriks

- Menghitung total
- Mengalikan matriks
- Transpose matriks
- Inverse Matriks

Operasi 2 Matriks

- Menambahkan 2 matriks
- Mengalikan 2 matriks
- Pembagian 2 matriks

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([0  
.lookup.StaticV  
_buckets=5)
```


Subprogram

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Definisi Subprogram

Subprogram adalah **serangkaian instruksi dirancang untuk melakukan operasi yang sering digunakan dalam suatu program**

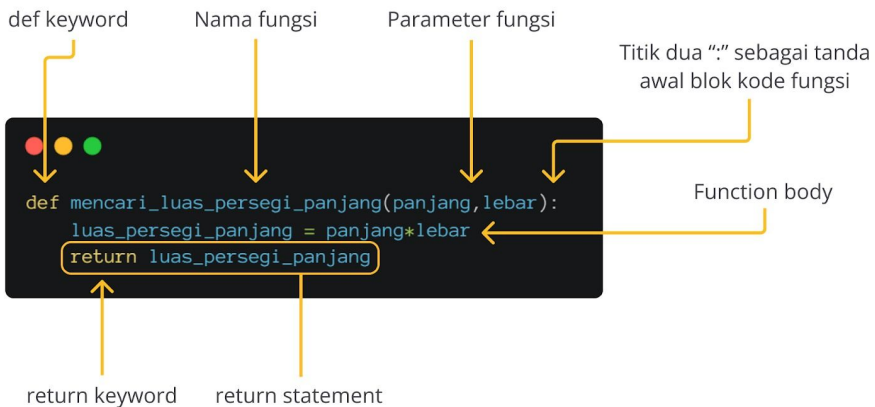
```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

Fungsi

Fungsi adalah **blok kode yang dapat menerima input, melakukan pemrosesan, dan mengembalikan output. Pada Python terdapat 2 jenis fungsi yaitu Built-in Functions & User-defined Functions**

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

Fungsi User-defined



The diagram illustrates the components of a Python function definition. It shows a code block with the following text:

```
def mencari_luas_persegi_panjang(panjang, lebar):  
    luas_persegi_panjang = panjang*lebar  
    return luas_persegi_panjang
```

Annotations with arrows point to specific parts of the code:

- def keyword**: Points to the `def` keyword.
- Nama fungsi**: Points to the function name `mencari_luas_persegi_panjang`.
- Parameter fungsi**: Points to the parameters `panjang, lebar`.
- Titik dua ":" sebagai tanda awal blok kode fungsi**: Points to the colon at the end of the function signature.
- Function body**: Points to the indented lines of code inside the function.
- return keyword**: Points to the `return` keyword.
- return statement**: Points to the entire `return luas_persegi_panjang` line.

```
children: [  
  con(icon, color: color  
  ontainer(  
    margin: const EdgeIns  
    child:  
      label  
      style
```



Google Developer Student Clubs
University Name

QNA



Machine Learning

Post Test

[link](#)

Feedback

[link](#)

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```





Google Developer Student Clubs

UPN "Veteran" Jawa Timur

SEE YOU

Next Weekly Class!

Machine Learning