

# 1. Variables, Basic Data Types, and Print

Penggunaan variabel atau suatu objek yang bisa merepresentasikan sebuah nilai atau value sangat penting dalam bahasa pemrograman. Selain itu mempermudah dalam membaca source code, pemberian variable yang efisien juga akan membuat code berjalan optimal dan dinamis. Pada sesi kali ini kita akan belajar bagaimana cara inisialisasi variable dalam beberapa data type dan menampilkannya (print).

Contoh tipe data :

1. **Boolean** : Menyatakan benar True yang bernilai 1, atau salah False yang bernilai 0
2. **String** : Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda " atau ')
3. **Integer** : Menyatakan bilangan bulat
4. **Float** : Menyatakan bilangan berkoma/desimal
5. **List** : Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
6. **Tuple** : Data untaian yang menyimpan berbagai tipe tapi isinya tidak dapat diubah
7. **Dictionary** : Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

Contoh :

In [1]:

```
var_string = "Belajar Python DQLAB"
var_int = 10
var_float = 3.14
var_list = [1, 2, 3]
var_tuple = ("satu", "dua", "tiga")
var_dict = {"nama" : "Zein", 'umur' : 20}

print (var_string)
print (var_int)
print (var_float)
print (var_list)
print (var_tuple)
print (var_dict)

print(type(var_string))
print(type(var_int))
print(type(var_float))
print(type(var_list))
print(type(var_tuple))
print(type(var_dict))
```

```
Belajar Python DQLAB
10
3.14
[1, 2, 3]
('satu', 'dua', 'tiga')
{'nama': 'Zein', 'umur': 20}
<class 'str'>
<class 'int'>
<class 'float'>
<class 'list'>
<class 'tuple'>
<class 'dict'>
```

## 2. Control Structure : Percabangan

Dalam membuat suatu keputusan biasanya pasti ada sebuah pilihan. Untuk mewujudkan suatu kerangka

Dalam membuat suatu keputusan biasanya pada ada sebuah pilihan untuk melakukan suatu kegiatan berpikir pada suatu pilihan pada bahasa pemrograman biasanya kita menggunakan konsep IF, ELSE. Pada python sebenarnya hampir sama dan tidak ada perbedaan yang terlalu signifikan. Perbedaanya mungkin ada pada

Pada praktek kali ini akan dibagi menjadi 3 bagian :

- Menggunakan IF yang jika direpresentasikan dengan kata – kata, (Jika, sesuatu memenuhi suatu kondisi, maka lakukan A, jika tidak, tidak terjadi apa-apa)
- Menggunakan IF dan ELSE yang jika direpresentasikan dengan kata – kata, (Jika, sesuatu memenuhi suatu kondisi, maka lakukan A, jika tidak, lakukan B)
- Menggunakan IF, ELIF dan ELSE. ELIF sendiri sebenarnya sama persis dengan ELSE IF. Namun, pada python disingkat menjadi ELSE IF. (Jika, sesuatu memenuhi suatu kondisi, maka lakukan A, jika tidak, lakukan pengecekan pada kondisi berikutnya, jika memenuhi lakukan B, jika tidak maka lakukan C)

### Contoh if

In [2]:

```
i = 10 #inisialisasi variable i yang memiliki nilai 10

if (i==10) : #pengecekan nilai i apakah sama dengan 10
    print("ini adalah angka 10") #jika TRUE maka akan mencetak kalimat ini
```

ini adalah angka 10

### Contoh if - elif

In [3]:

```
i = 5 #inisialisasi variable i yang memiliki nilai 10

if (i==10): #pengecekan nilai i apakah sama dengan 10
    print("ini adalah angka 10") #jika TRUE maka akan mencetak kalimat ini
else:
    print("bukan angka 10") #jika FALSE akan mencetak kalimat ini
```

bukan angka 10

### Contoh if - elif - else

In [4]:

```
i=3

if (i==5):
    print("ini adalah angka 5")
elif (i>5):
    print("lebih besar dari 5")
else:
    print("lebih kecil dari 5")
```

lebih kecil dari 5

### Contoh Nested if

Fungsi IF sendiri bisa digunakan secara bertingkat. Atau dilakukan pengecekan lebih dari 1 statement di dalamnya. Semisal seperti ini: if ( i<7 && i <3)

Contoh :

In [5]:

```
i=2
if (i<7):
```

```
print("nilai i kurang dari 7")
if (i<3):
    print("nilai i kurang dari 7 dan kurang dari 3")
else:
    print("nilai i kurang dari 7 tapi lebih dari 3")
```

nilai i kurang dari 7

nilai i kurang dari 7 dan kurang dari 3

---

## 3. Operator Matematika dan Pembandingan

Contoh operator matematika :

- **Tambah : +**
- **Kurang : -**
- **Pembagi : /**
- **Perkalian : \***
- **Pangkat : \*\***
- **Modulus : %**
- **Kurang dari : <**
- **Lebih dari : >**
- **Kurang dari sama dengan : <=**
- **Lebih dari sama dengan : >=**
- **Sama dengan : =**

### Contoh 1

In [7]:

```
a=10
b=5
selisih = a-b
jumlah = a+b
kali = a*b
bagi = a/b
print("Hasil penjumlahan dan b adalah", jumlah)
print("Selisih a dan b adalah :", selisih)
print("Hasil perkalian a dan b adalah :", kali)
print("Hasil pembagian a dan b adalah:", bagi)
```

Hasil penjumlahan dan b adalah 15

Selisih a dan b adalah : 5

Hasil perkalian a dan b adalah : 50

Hasil pembagian a dan b adalah: 2.0

### Contoh 2 : Modulus

In [8]:

```
c=10
d=3

modulus=c%d
print("Hasil modulus", modulus)
```

Hasil modulus 1

---

# Studi kasus 1

Buatlah sebuah program yang bisa menentukan suatu nilai pada variable itu genap atau ganjil! Lakukan percobaan dengan langkah berikut:

Buat variabel dengan nama "angka" isi dengan nilai 10 Ganti bagian ... dengan perhitungan untuk menentukan angka modulus 2 bernilai 0. Cek apakah benar dia bernilai genap maka keluarkan "angka termasuk bilangan genap", jika sudah ganti nilai variable angka dengan nilai 5 cek lagi apakah dia bernilai ganjil dan keluarkan "angka termasuk bilangan ganjil".

**Note:** Anda bisa menggunakan materi-materi sebelumnya dengan menggabungkannya di tugas praktek ini untuk membantu anda dalam mengerjakan.

In [10]:

```
angka=5

if(angka%2 == 0):
    print("angka termasuk bilangan genap")
else:
    print("angka termasuk bilangan ganjil")

angka termasuk bilangan ganjil
```

## 4. Looping (Perulangan)

Looping berfungsi untuk memberikan perulangan pada program. Contohnya, ketika membutuhkan proses 1000x, maka tidak mungkin jika kita harus memproses 1000x eksekusi code. Maka dari itu, dibuat perulangan untuk mengatasinya.

Selain tidak bersifat scalable, efisiensi akan mempengaruhi performa dari program yang kita buat. Seperti pada umumnya pada python juga memiliki fungsi for dan while untuk melakukan looping. Logikanya hampir sama, namun hanya berbeda dalam penulisannya jika Anda sudah terbiasa dengan C ataupun Java.

### a) Jenis Perulangan While

Struktur while pada python tidak berbeda jauh dengan bahasa pemrograman lainnya. Lebih sederhananya struktur penulisan python membuat dia mudah untuk dibaca.

In [11]:

```
j = 0 #nilai awal j =0

while j<6: #ketika j kurang dari 6 lakukan perulangan, jika tidak stop perulangan
    print("Ini adalah perulangan ke -",j) #lakukan perintah ini ketika perulangan
    j=j+1 #setiap kali diakhir perulangan update nilai dengan ditambah 1.
```

```
Ini adalah perulangan ke - 0
Ini adalah perulangan ke - 1
Ini adalah perulangan ke - 2
Ini adalah perulangan ke - 3
Ini adalah perulangan ke - 4
Ini adalah perulangan ke - 5
```

### b) Jenis Perulangan for

In [12]:

```
for i in range (1,6): #perulangan for sebagai inisialisasi dari angka 1 hingga angka yang lebih kecil daripada 6.

    print("Ini adalah perulangan ke -", i) #perintah jika looping akan tetap berjalan
```

```
Ini adalah perulangan ke - 1
Ini adalah perulangan ke - 2
Ini adalah perulangan ke - 3
Ini adalah perulangan ke - 4
Ini adalah perulangan ke - 5
```

**Penjelasan sebelum di RUN :**

**Maksud dari fungsi ini for i in range (1,6):** jika kita konversi pada JAVA atau C sama dengan for(i=1;i<6i++). Jika dikonversi menjadi kalimat adalah “perulangan dimulai dari nilai i = 1 hingga nilai i kurang dari 6 dimana setiap kali perulangan nilai i akan selalu ditambah 1”. Jika nilai i sudah mencapai 6 perulangan akan dihentikan.

### c) Jenis Perulangan for dengan Access

Keunikan lain dari looping dengan python adalah selain bahasa yang mudah dimengerti dalam looping, kita juga bisa mengakses elemen yang terdapat pada sebuah list.

In [13]:

```
count=[1,2,3,4,5] #elemen list

for number in count: #looping untuk menampilkan semua elemen pada count
    print("Ini adalah element count : ", number) #menampilkan elemen list pada count
```

```
Ini adalah element count : 1
Ini adalah element count : 2
Ini adalah element count : 3
Ini adalah element count : 4
Ini adalah element count : 5
```

## Studi kasus 2

- Buatlah sebuah program yang bisa mengeluarkan angka 1 sampai 10.
- Tampilan akan menunjukkan "Angka ganjil 1" untuk angka ganjil dan "Angka genap 2" untuk angka genap. (Menggunakan looping for)

**Note:** Kode dasar sudah disertakan, Anda cukup mengganti tanda # dengan nilai-nilai yang sesuai.

In [14]:

```
for i in range (1,11):
    if(i%2 == 0):
        print("Angka genap",i)
    else:
        print("Angka ganjil",i)
```

```
Angka ganjil 1
Angka genap 2
Angka ganjil 3
Angka genap 4
Angka ganjil 5
Angka genap 6
Angka ganjil 7
Angka genap 8
Angka ganjil 9
Angka genap 10
```

## 5. Function

Pada pembuatan program yang kompleks dan memiliki banyak fitur, kita diharuskan menggunakan fungsi. Kenapa memangnya kalau tidak menggunakan fungsi? Bisa jadi kita akan kerepotan menulis kode programnya, karena banyak yang harus ditulis dan kode akan menjadi sulit dibaca dan dirawat (maintenance). Dengan fungsi, kita dapat memecah program besar menjadi sub program yang lebih sederhana. Masing-masing fitur pada program dapat kita buat dalam satu fungsi. Pada saat kita membutuhkan fitur tersebut, kita tinggal panggil fungsinya saja.

### a) Membuat Fungsi Sendiri

Fungsi pada Python, dibuat dengan kata kunci `def` kemudian diikuti dengan nama fungsinya.

Format penulisannya adalah sebagai berikut,

```
=====
```

inisiasi nama\_fungsi() :

<--indentasi--> isi\_fungsi

```
=====
```

Contoh :

In [18]:

```
def hello():  
    print("Hello ini Fungsi") # Fungsi 1
```

In [17]:

```
def salam():  
    print("Hello, Selamat Pagi") # Fungsi 2
```

Cara memanggil fungsinya adalah sebagai berikut :

In [19]:

```
hello() # Memanggil fungsi 1  
salam() # Memanggil fungsi 2
```

```
Hello ini Fungsi  
Hello, Selamat Pagi
```

### b) Fungsi berparameter

Sekarang, bagaimana kalau kita ingin memberikan input nilai ke dalam fungsi? Kita bisa menggunakan parameter. Parameter adalah variabel yang menampung nilai untuk diproses kedalam suatu fungsi. Formatnya :

```
=====
```

inisiasi nama\_fungsi(parameter) :

<--indentasi--> isi\_fungsi

```
=====
```

Pada contoh kali ini untuk mempermudah pembelajaran telah disediakan rumusan matematika untuk menghitung luas pada segitiga. Berikut ini contoh kodenya:

In [20]:

```
def luas_segitiga(alas, tinggi): #alas dan tinggi merupakan parameter yang masuk
    luas = (alas * tinggi) / 2
    print("Luas segitiga: %f" % luas);
```

In [21]:

```
# Pemanggilan fungsi
##4 dan 6 merupakan parameter yang diinputkan kedalam fungsi luas segitiga
luas_segitiga(4, 6)
```

Luas segitiga: 12.000000

### c) Fungsi dengan Return Value

Fungsi yang tidak mengembalikan nilai biasanya disebut dengan prosedur. Namun, kadang kita butuh hasil proses dari fungsi untuk digunakan pada proses berikutnya. Maka fungsi harus mengembalikan nilai dari hasil pemrosesannya. Cara mengembalikan nilai adalah menggunakan kata kunci return lalu diikuti dengan nilai atau variabel yang akan dikembalikan.

=====

inisiasi nama\_fungsi(parameter) :

<--indentasi--> isi\_fungsi

return variabel\_return

=====

Meneruskan kembali di poin sebelumnya. Untuk fungsi luas segitiga yang sudah dibuat, mari tambahkan return value sehingga menjadi seperti ini:

In [22]:

```
def luas_segitiga(alas, tinggi): #alas dan tinggi merupakan parameter yang masuk
    luas = (alas * tinggi) / 2
    return luas
```

In [23]:

```
# Pemanggilan fungsi
##4 dan 6 merupakan parameter yang diinputkan kedalam fungsi luas segitiga
print("Luas segitiga: %d" % luas_segitiga(4, 6))
```

Luas segitiga: 12

## 6. Modul dan Package

Modul memungkinkan Anda mengatur kode Python secara logis. Mengelompokkan kode terkait ke dalam modul membuat kode lebih mudah dipahami dan digunakan. Modul adalah objek Python dengan atribut yang diberi nama yang bisa Anda tambahkan dan dijadikan referensi dalam membantu dalam proses pembuatan program. Secara sederhana, modul adalah file yang terdiri dari kode Python. Modul dapat mendefinisikan fungsi, kelas, dan variabel. Python sendiri sudah memiliki banyak library dan module yang bisa diakses oleh pengguna secara free.

## a) Import Package dan Menggunakan modul

Pada sesi kali ini, praktek akan menggunakan module math yang merupakan module standar untuk operasi matematika. Berikut ini contohnya :

```
import math
```

Cara menggunakan module cukup sederhana. Module sendiri berisi fungsi-fungsi. Cara orang melakukan akses kedalam fungsi juga berbeda. Coba lihat dan implementasikan kode di bawah ini :

In [24]:

```
import math
print("Nilai pi adalah:", math.pi) # math.pi merupakan sintak untuk memanggil fungsi
```

Nilai pi adalah: 3.141592653589793

## b) Import dengan Module Rename atau Alias

Kita bisa mengimpor modul dengan menamainya. Hal ini biasanya kita lakukan untuk menyingkat nama modul yang panjang.

In [25]:

```
import math as m #menggunakan m sebagai module rename atau alias
print("Nilai pi adalah:", m.pi) #m.pi merupakan sintak untuk memanggil fungsi
```

Nilai pi adalah: 3.141592653589793

## c) Import Sebagian Fungsi

Pada suatu module tidak bisa dipungkiri terdiri dari puluhan bahkan ribuan fungsi. Namun, yang kita butuhkan hanya 1 atau 2 fungsi saja. Untuk meminimalisir ketidakefisienan suatu program dalam load suatu module bisa dilakukan import module namun hanya beberapa fungsi saja yang kita import kedalam code. Format (from module\_name import function\_name)

In [27]:

```
from math import pi
print("Nilai pi adalah", pi)
```

Nilai pi adalah 3.141592653589793

## d) Import Semua isi Moduls

Namun, jika memang yang dibutuhkan banyak, semisal lebih dari 10 atau bahkan ratusan fungsi, bisa dilakukan import semuanya dengan menggunakan format from module\_name import \*. Tanda disini menunjukan semua fungsi diimport kedalam code.

In [28]:

```
from math import *
print("Nilai e adalah:", e)
```

Nilai e adalah: 2.718281828459045



## 7. Membaca dari File

CSV atau comma separated value adalah salah satu tipe file yang digunakan secara luas di dunia programming. Tidak hanya itu CSV pun sering digunakan dalam pengolahan informasi yang dihasilkan spreadsheet untuk diproses lebih lanjut melalui mesin analitik. CSV pun dianggap sebagai file yang agnostik karena dapat digunakan oleh berbagai database untuk proses backup data. CSV dianggap sebagai salah satu tipe data yang sering dipakai untuk mengelola data pada proses lanjutan.

### a) Membaca Teks File (CSV)

Sekarang kita akan mencoba membaca sebuah file CSV yang telah dihasilkan aplikasi atau program lain. Di Python, hasil pembacaan setiap baris pada file CSV akan dikonversi menjadi list Python.

Berikut adalah contoh kode untuk membaca file CSV dengan kasus yang sangat sederhana, coba ketik kode di bawah ini pada Code Editor:

In [39]:

```
import csv

# tentukan lokasi file, nama file, dan inisialisasi csv
f = open('penduduk_gender_head.csv', 'r')
reader = csv.reader(f)

# membaca baris per baris
for row in reader:
    print (row)

# menutup file csv
f.close()
```

```
['TAHUN', 'NAMA PROVINSI', 'NAMA KABUPATEN/KOTA', 'NAMA KECAMATAN', 'NAMA KELURAHAN', 'LA
KI-LAKI WNI', 'PEREMPUAN WNI', 'LAKI-LAKI WNA', 'PEREMPUAN WNA', '', '', '', '', '', '',
'', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'GAMBIR', 'GAMBIR', '1790', '1690', '1'
, '2', '', '', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'GAMBIR', 'CIDENG', '9159', '9206', '5'
, '6', '', '', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'GAMBIR', 'PETOJO UTARA', '10811', '104
36', '10', '8', '', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'GAMBIR', 'PETOJO SELATAN', '8455', '80
23', '5', '7', '', '', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'GAMBIR', 'KEBON KELAPA', '6300', '6078
', '6', '10', '', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'GAMBIR', 'DURI PULO', '13056', '12588'
, '2', '6', '', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SAWAH BESAR', 'PASAR BARU', '7557', '7
552', '34', '55', '', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SAWAH BESAR', 'KARANG ANYAR', '16327',
'15859', '4', '2', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SAWAH BESAR', 'KARTINI', '13610', '138
08', '3', '3', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SAWAH BESAR', 'GUNUNG SAHARI UTARA', '
9734', '9889', '15', '30', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SAWAH BESAR', 'MANGGA DUA SELATAN', '1
7269', '16458', '42', '33', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'KEMAYORAN', '12365', '119
32', '6', '8', '', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'KEBON KOSONG', '15933', '
15665', '33', '25', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'HARAPAN MULIA', '13523',
'13049', '2', '3', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'SERDANG', '17284', '17020
', '1', '1', '', '', '', '']
['2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'GUNUNG SAHARI SELATAN', '
11730', '11487', '13', '15', '', '', '', '']
```

```
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'CEMPAKA BARU', '19103', '18601', '3', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'SUMUR BATU', '13441', '13288', '20', '13', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'KEMAYORAN', 'UTAN PANJANG', '17125', '16351', '4', '3', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SENEN', 'SENEN', '4236', '3975', '10', '7', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SENEN', 'KENARI', '5445', '5252', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SENEN', 'PASEBAN', '13787', '13477', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SENEN', 'KRAMAT', '17482', '16331', '2', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SENEN', 'KWITANG', '9176', '9148', '1', '1', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'SENEN', 'BUNGUR', '11200', '10849', '4', '3', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'CEMPAKA PUTIH', 'CEMPAKA PUTIH TIMUR', '13630', '13686', '24', '24', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'CEMPAKA PUTIH', 'CEMPAKA PUTIH BARAT', '20029', '19681', '12', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'CEMPAKA PUTIH', 'RAWASARI', '12462', '12465', '1', '3', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'MENTENG', 'MENTENG', '14576', '14610', '7', '10', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'MENTENG', 'PEGANGSAAN', '13392', '13214', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'MENTENG', 'CIKINI', '4825', '4772', '5', '1', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'MENTENG', 'GONDANGDIA', '2196', '2443', '11', '13', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'MENTENG', 'KEBON SIRIH', '7976', '7439', '2', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'TANAH ABANG', 'GELORA', '1865', '1883', '6', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'TANAH ABANG', 'BENDUNGAN HILIR', '12713', '12582', '11', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'TANAH ABANG', 'KARET TENGSIN', '11002', '10271', '10', '3', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'TANAH ABANG', 'PETAMBURAN', '20112', '18794', '17', '13', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'TANAH ABANG', 'KEBON MELATI', '19826', '18672', '8', '4', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'TANAH ABANG', 'KEBON KACANG', '12873', '12357', '3', '4', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'TANAH ABANG', 'KAMPUNG BALI', '7348', '7205', '3', '1', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'JOHAR BARU', 'JOHAR BARU', '21016', '20953', '2', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'JOHAR BARU', 'KAMPUNG RAWA', '13337', '12499', '1', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'JOHAR BARU', 'GALUR', '11077', '10259', '']
[ '2013', 'PROVINSI DKI JAKARTA', 'JAKARTA PUSAT', 'JOHAR BARU', 'TANAH TINGGI', '22680', '21450', '']
```

## b) Membaca file CSV dengan menggunakan PANDAS

Bagi yang belum familiar, **PANDAS** merupakan salah satu library yang sangat sering digunakan untuk aplikasi dan implementasi data science. Baik untuk data manipulation, data pre-processing, atau data wrangling. Pada sesi kali ini, kita akan menggunakan **PANDAS** untuk membaca file dari csv.

Sebelumnya harus menginstall dulu setidaknya library pandas, numpy, dan matplotlib.

Cobalah ketik code di bawah ini:

In [38]:

```
import pandas as pd
```

```
table = pd.read_csv("penduduk_gender_head.csv")
table.head()
print(table)
```

	TAHUN	NAMA PROVINSI	NAMA KABUPATEN/KOTA	NAMA KECAMATAN	\
0	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	GAMBIR
1	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	GAMBIR
2	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	GAMBIR
3	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	GAMBIR
4	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	GAMBIR
5	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	GAMBIR
6	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SAWAH BESAR
7	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SAWAH BESAR
8	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SAWAH BESAR
9	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SAWAH BESAR
10	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SAWAH BESAR
11	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
12	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
13	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
14	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
15	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
16	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
17	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
18	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	KEMAYORAN
19	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SENEN
20	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SENEN
21	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SENEN
22	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SENEN
23	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SENEN
24	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	SENEN
25	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	CEMPAKA PUTIH
26	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	CEMPAKA PUTIH
27	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	CEMPAKA PUTIH
28	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	MENTENG
29	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	MENTENG
30	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	MENTENG
31	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	MENTENG
32	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	MENTENG
33	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	TANAH ABANG
34	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	TANAH ABANG
35	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	TANAH ABANG
36	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	TANAH ABANG
37	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	TANAH ABANG
38	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	TANAH ABANG
39	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	TANAH ABANG
40	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	JOHAR BARU
41	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	JOHAR BARU
42	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	JOHAR BARU
43	2013	PROVINSI DKI	JAKARTA	JAKARTA PUSAT	JOHAR BARU

	NAMA KELURAHAN	LAKI-LAKI WNI	PEREMPUAN WNI	LAKI-LAKI WNA	\
0	GAMBIR	1790	1690	1.0	
1	CIDENG	9159	9206	5.0	
2	PETOJO UTARA	10811	10436	10.0	
3	PETOJO SELATAN	8455	8023	5.0	
4	KEBON KELAPA	6300	6078	6.0	
5	DURI PULO	13056	12588	2.0	
6	PASAR BARU	7557	7552	34.0	
7	KARANG ANYAR	16327	15859	4.0	
8	KARTINI	13610	13808	3.0	
9	GUNUNG SAHARI UTARA	9734	9889	15.0	
10	MANGGA DUA SELATAN	17269	16458	42.0	
11	KEMAYORAN	12365	11932	6.0	
12	KEBON KOSONG	15933	15665	33.0	
13	HARAPAN MULIA	13523	13049	2.0	
14	SERDANG	17284	17020	1.0	
15	GUNUNG SAHARI SELATAN	11730	11487	13.0	
16	CEMPAKA BARU	19103	18601	NaN	
17	SUMUR BATU	13441	13288	20.0	
18	UTAN PANJANG	17125	16351	4.0	
19	SENEN	10000	9875	10.0	

19	SENEN	4236	3975	10.0
20	KENARI	5445	5252	2.0
21	PASEBAN	13787	13477	NaN
22	KRAMAT	17482	16331	2.0
23	KWITANG	9176	9148	1.0
24	BUNGUR	11200	10849	4.0
25	CEMPAKA PUTIH TIMUR	13630	13686	24.0
26	CEMPAKA PUTIH BARAT	20029	19681	12.0
27	RAWASARI	12462	12465	1.0
28	MENTENG	14576	14610	7.0
29	PEGANGSAAN	13392	13214	2.0
30	CIKINI	4825	4772	5.0
31	GONDANGDIA	2196	2443	11.0
32	KEBON SIRIH	7976	7439	2.0
33	GELORA	1865	1883	6.0
34	BENDUNGAN HILIR	12713	12582	11.0
35	KARET TENGSIN	11002	10271	10.0
36	PETAMBURAN	20112	18794	17.0
37	KEBON MELATI	19826	18672	8.0
38	KEBON KACANG	12873	12357	3.0
39	KAMPUNG BALI	7348	7205	3.0
40	JOHAR BARU	21016	20953	2.0
41	KAMPUNG RAWA	13337	12499	1.0
42	GALUR	11077	10259	NaN
43	TANAH TINGGI	22680	21450	NaN

	PEREMPUAN WNA	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	\
0	2.0	NaN	NaN	NaN	NaN	
1	6.0	NaN	NaN	NaN	NaN	
2	8.0	NaN	NaN	NaN	NaN	
3	7.0	NaN	NaN	NaN	NaN	
4	10.0	NaN	NaN	NaN	NaN	
5	6.0	NaN	NaN	NaN	NaN	
6	55.0	NaN	NaN	NaN	NaN	
7	2.0	NaN	NaN	NaN	NaN	
8	3.0	NaN	NaN	NaN	NaN	
9	30.0	NaN	NaN	NaN	NaN	
10	33.0	NaN	NaN	NaN	NaN	
11	8.0	NaN	NaN	NaN	NaN	
12	25.0	NaN	NaN	NaN	NaN	
13	3.0	NaN	NaN	NaN	NaN	
14	1.0	NaN	NaN	NaN	NaN	
15	15.0	NaN	NaN	NaN	NaN	
16	3.0	NaN	NaN	NaN	NaN	
17	13.0	NaN	NaN	NaN	NaN	
18	3.0	NaN	NaN	NaN	NaN	
19	7.0	NaN	NaN	NaN	NaN	
20	NaN	NaN	NaN	NaN	NaN	
21	NaN	NaN	NaN	NaN	NaN	
22	2.0	NaN	NaN	NaN	NaN	
23	1.0	NaN	NaN	NaN	NaN	
24	3.0	NaN	NaN	NaN	NaN	
25	24.0	NaN	NaN	NaN	NaN	
26	2.0	NaN	NaN	NaN	NaN	
27	3.0	NaN	NaN	NaN	NaN	
28	10.0	NaN	NaN	NaN	NaN	
29	NaN	NaN	NaN	NaN	NaN	
30	1.0	NaN	NaN	NaN	NaN	
31	13.0	NaN	NaN	NaN	NaN	
32	2.0	NaN	NaN	NaN	NaN	
33	2.0	NaN	NaN	NaN	NaN	
34	2.0	NaN	NaN	NaN	NaN	
35	3.0	NaN	NaN	NaN	NaN	
36	13.0	NaN	NaN	NaN	NaN	
37	4.0	NaN	NaN	NaN	NaN	
38	4.0	NaN	NaN	NaN	NaN	
39	1.0	NaN	NaN	NaN	NaN	
40	NaN	NaN	NaN	NaN	NaN	
41	NaN	NaN	NaN	NaN	NaN	
42	NaN	NaN	NaN	NaN	NaN	
43	NaN	NaN	NaN	NaN	NaN	

	Unnamed: 13	Unnamed: 14	Unnamed: 15	Unnamed: 16	Unnamed: 17
0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN
9	NaN	NaN	NaN	NaN	NaN
10	NaN	NaN	NaN	NaN	NaN
11	NaN	NaN	NaN	NaN	NaN
12	NaN	NaN	NaN	NaN	NaN
13	NaN	NaN	NaN	NaN	NaN
14	NaN	NaN	NaN	NaN	NaN
15	NaN	NaN	NaN	NaN	NaN
16	NaN	NaN	NaN	NaN	NaN
17	NaN	NaN	NaN	NaN	NaN
18	NaN	NaN	NaN	NaN	NaN
19	NaN	NaN	NaN	NaN	NaN
20	NaN	NaN	NaN	NaN	NaN
21	NaN	NaN	NaN	NaN	NaN
22	NaN	NaN	NaN	NaN	NaN
23	NaN	NaN	NaN	NaN	NaN
24	NaN	NaN	NaN	NaN	NaN
25	NaN	NaN	NaN	NaN	NaN
26	NaN	NaN	NaN	NaN	NaN
27	NaN	NaN	NaN	NaN	NaN
28	NaN	NaN	NaN	NaN	NaN
29	NaN	NaN	NaN	NaN	NaN
30	NaN	NaN	NaN	NaN	NaN
31	NaN	NaN	NaN	NaN	NaN
32	NaN	NaN	NaN	NaN	NaN
33	NaN	NaN	NaN	NaN	NaN
34	NaN	NaN	NaN	NaN	NaN
35	NaN	NaN	NaN	NaN	NaN
36	NaN	NaN	NaN	NaN	NaN
37	NaN	NaN	NaN	NaN	NaN
38	NaN	NaN	NaN	NaN	NaN
39	NaN	NaN	NaN	NaN	NaN
40	NaN	NaN	NaN	NaN	NaN
41	NaN	NaN	NaN	NaN	NaN
42	NaN	NaN	NaN	NaN	NaN
43	NaN	NaN	NaN	NaN	NaN

---

## 8. Grafik Dasar dengan Matplotlib

Grafik merupakan salah satu perangkat visualisasi. Visualisasi sendiri sudah menjadi salah satu challenge atau poin penting dalam data science. Selain bisa mempermudah seseorang untuk memahami data, bagi para data scientist visualisasi bisa meningkatkan analisa atau memunculkan knowledge yang tidak sempat terekspose ketika tidak menggunakan visualisasi. Pada sesi kali ini, kita akan menggunakan matplotlib untuk melakukan visualisasi data yang sudah ada.

Sebelumnya harus menginstall dulu setidaknya library pandas, numpy, dan matplotlib.

### a) Bar Chart

Sebelum masuk pada membuat grafik bar, kita akan membahas library yang akan gunakan. Pada Python sangat banyak library untuk visualisasi. Salah satu yang paling sering digunakan adalah matplotlib karena memang sudah lama dan relatif stabil dalam perkembangannya. Maka dari itu, matplotlib dipilih untuk belajar visualisasi pada sesi kali ini. Matplotlib sendiri menyediakan banyak jenis grafik mulai dari scatter/plot. line. bar. dan lain-

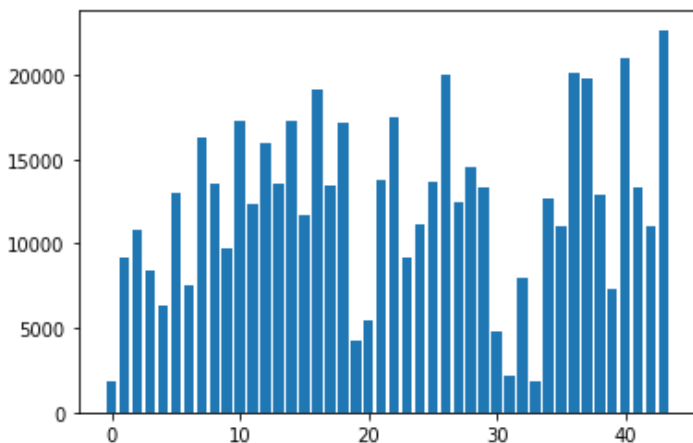
pada saat ini matplotlib belum menggunakan font yang keren maka akan diganti dengan font yang lebih keren. Selain itu, pada saat ini kita akan menambahkan library selain dengan menggunakan matplotlib dan pandas, kita akan menggunakan numpy. Numpy sendiri pada praktek kali ini digunakan untuk melakukan manipulasi data dari csv untuk memudahkan visualisasi.

Ketikan kode di bawah ini :

In [37]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

table = pd.read_csv("penduduk_gender_head.csv")
table.head()
x_label = table['NAMA KELURAHAN']
plt.bar(x=np.arange(len(x_label)), height=table['LAKI-LAKI WNI'])
plt.show()
```



## b) Parameter dalam Grafik (Memberikan Nilai Axis dari data CSV)

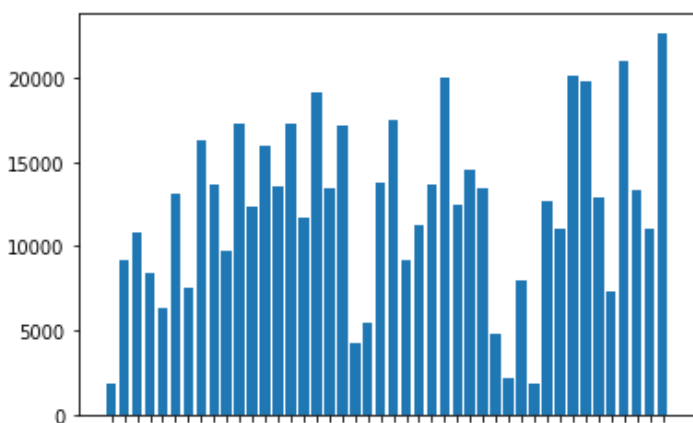
Jika ditelisik lebih dalam pada CSV, disitu ada nama kelurahan yang merupakan variabel atau seharusnya menjadi nilai AXIS pada grafik ini. Di praktek kali ini, kita akan mencoba bagaimana menempatkan nama kelurahan pada grafik.

In [40]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

table = pd.read_csv("https://academy.dqlab.id/dataset/penduduk_gender_head.csv")
table.head()

x_label = table['NAMA KELURAHAN']
plt.bar(x=np.arange(len(x_label)), height=table['LAKI-LAKI WNI'])
plt.xticks(np.arange(len(x_label)), table['NAMA KELURAHAN'], rotation=30)
plt.show()
```



GAMBER  
PETJO LUBA  
PERUMAHAN  
KARANG ANJAN  
GAMBANG  
MAWIS PAKSI  
GUNUNG SAMAN  
CEMPAKA  
BANDARA  
BEKAS  
KAMPAY TINGGI

Pada implementasi grafik, penting nyatanya untuk memberikan label pada AXIS dan Ordinat untuk menjelaskan maksud grafik. Pada Praktek kali ini, kita akan mencoba memberikan label dan title pada grafik yang telah dibuat sebelumnya.

**Cobalah ketika kode di bawah ini :**

In [41]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

table = pd.read_csv("https://academy.dqlab.id/dataset/penduduk_gender_head.csv")
table.head()

x_label = table['NAMA KELURAHAN']
plt.bar(x=np.arange(len(x_label)),height=table['LAKI-LAKI WNI'])
plt.xticks(np.arange(len(x_label)), table['NAMA KELURAHAN'], rotation=90)
plt.xlabel('Keluarahan di Jakarta pusat')
plt.ylabel('Jumlah Penduduk Laki - Laki')
plt.title('Persebaran Jumlah Penduduk Laki- Laki di Jakarta Pusat')

plt.show()
```

